Management of Temporally and Spatially Correlated Failures in Federated Message Oriented Middleware for Resilient and QoS-Aware Messaging Services

PhD Thesis

Jinfu Wang

Supervisors

John Bigham, Chris Phillips

Department of Electronic Engineering and Computer Science

Queen Mary University of London

April 2015

Acknowledgements

My first acknowledgement goes to my supervisors Dr. John Brigham and Dr. Chris Philips for their support and friendship. Also I would like to thank Dr. Wenye Li and my friends across seas for their encouragements and inspiration in both work and life.

My special and earnest thanks go to my family for their gracious support and their caring especially during the time I was unwell. They are the indispensable driving force for my recovery and finishing this work.

Abstract

Message Oriented Middleware (MOM) is widely recognized as a promising solution for the communications between heterogeneous distributed systems. Because the resilience and quality-of-service of the messaging substrate plays a critical role in the overall system performance, the evolution of these distributed systems has introduced new requirements for MOM, such as inter domain federation, resilience and QoS support.

This thesis focuses on a management frame work that enhances the Resilience and QoS-awareness of MOM, called RQMOM, for federated enterprise systems. A common hierarchical MOM architecture for the federated messaging service is assumed. Each bottom level local domain comprises a cluster of neighbouring brokers that carry a local messaging service, and inter domain messaging are routed through the gateway brokers of the different local domains over the top level federated overlay. Some challenges and solutions for the intra and inter domain messaging are researched.

In local domain messaging the common cause of performance degradation is often the fluctuation of workloads which might result in surge of total workload on a broker and overload its processing capacity, since a local domain is often within a well connected network. Against performance degradation, a combination of novel proactive risk-aware workload allocation, which exploits the co-variation between workloads, in addition to existing reactive load balancing is designed and evaluated.

In federated inter domain messaging an overlay network of federated gateway brokers distributed in separated geographical locations, on top of the heterogeneous physical network is considered. Geographical correlated failures are threats to cause major interruptions and damages to such systems. To mitigate this rarely addressed challenge, a novel geographical location aware route selection algorithm to support uninterrupted messaging is introduced. It is used with existing overlay routing mechanisms, to maintain routes and hence provide more resilient messaging against geographical correlated failures.

Table of Contents

Index of Tables

Index of Figures

Abbreviations

AS	Autonomous System
BGP	Border Gateway Protocol
BRITE	Boston university Representative Internet Topology Generator
CORBA	The Common Object Request Broker Architecture
CPS	Cyber Physical Systems
DCOM	Distributed Component Object Model
DDS	Data Distribution Service
DSS	Decision Support System
EA	Evolutionary Algorithm
EKSP	Enhanced K-shortest Path
GA	Genetic Algorithm
GAP	Geo-aware Alternate Path Algorithm
GEMOM	Genetic Message Oriented Secure Middleware
GPS	Global Positioning System
HA	High Availability
IoT	Internet of Things
ISP	Internet Service Provider
JGAP	Java Genetic Algorithms Package
JMS	Java Messaging Service
JVM	Java Virtual Machine
KSP	K-shortest Path
LAN	Local Area Network
LD	Load Detector
LSA	Link State Advertisement
MA	Managing Agent
MDP	Markov Decision Processes
MOM	Message Oriented Middleware
MPLS	Multiprotocol Label Switching
NGN	Next Generation Network
NNLS	Non-negative Least-square
NP	Non-deterministic Polynomial-time
OM	Overlay Manager
OSPF	Open Shortest Path First
P2P	Peer to Peer
PF	Proximity Factor
Pub/Sub or P/S	Publish and Subscribe
QoS	Quality of Service
RLB	Reactive Load Balancing
RM	Resilience Manager
RMI	Remote Method Invocation

RON	Resilient Overlay Network
RPC	Remote Procedure Call
RQMOM	Resilient and QoS-Aware MOM Management
RSS	Rich Site Summary
SOA	Service Oriented Architecture
SON	Service Overlay Networks
SoS	Systems of Systems
SQS	Simple Queuing Service
TRIDEC	Collaborative, Complex and Critical Decision-Support in Evolving Crises
VM	Virtual Machine
WAN	Wide Area Network
WDM	Wavelength-division multiplexing
XSLT	EXtensible Stylesheet Language Transformations

Chapter 1 Introduction

1.1 Motivation

Message Oriented middleware (MOM) is widely recognized as a promising approach that is becoming increasingly widespread for the communications between heterogeneous distributed systems. It is applied when designing and building applications across different domains, including application integration [1], cyber physical systems (CPS) [2, 3], Internet of Things (IoT)[4-6], the federation of systems of systems (SoS) [7], data dissemination [8], Rich Site Summary (RSS) feed distribution and filtering [9, 10], SOA environments [11] and business process management [12]. This is because messaging is a simple and natural communication paradigm for connecting the loosely-coupled and distributed components in those systems. According to the application scenario the scale of MOM systems varies considerably, from intra-domain messaging where there can be only one or a few brokers to inter-domain federation of several geographical or organizationally separated systems; and to very large scale messaging over the internet. MOMs often support both message queuing and publish subscribe (Pub/Sub) paradigms to provide support for different application semantics. Many MOM standards and products have been adopted. Examples include IBM Websphere MQ, Java Messaging Service (JMS), and the Advanced Message Queuing Protocol (AMQP). Pub/sub and message queuing are fundamental mechanisms for interconnecting disparate services and systems in the service-oriented computing architecture.

Enterprise computing system, which is the focus of this research, is undergoing major transformation. These transformations have introduced new non-functional requirements for MOM, such as quality-of-service (QoS) and self-* capabilities. Self-* refers to the automatic actions of information processing systems to adapt dynamically to changes. It includes self-configuration, self-optimization, self-healing, self-protection, etc. To keep

up with these new requirements, MOM must be improved with new designs and supporting mechanisms to be aware of and satisfy the particular needs of these new systems. Additionally MOM is required to enable resilient messaging, i.e. provide and maintain an acceptable level of service in the face of faults and challenges to normal operation, such as internal broker faults or burstiness of workload, and external substrate network (a.k.a. underlay network) congestion or failures.

An example of such major transformation is the growing interconnection and interoperation of enterprise systems over a geographically widely distributed area, as triggered by either infrastructure or architecture requirements such as connecting several data centres across countries. Other drivers are business practices like partnerships, off-shoring, outsourcing, and the formation of virtual enterprises [13]. An emerging engineering discipline is the study of SoS [7]. In the realm of SoS, the constituent systems may be distributed over a large geographic area, e.g., across a nation or even spanning multiple continents. Messages between the systems often have to travel a long communication path, incurring much larger delay than local-area messaging. It is also harder to maintain stable QoS and high availability because a long-haul communication path increases the number of nodes and links along the path. Further, the systems are likely to be deployed and operated by separate organizations, which result in different security properties and degrees of trustworthiness to be associated with these systems. Many SoS applications require messaging capabilities with certain assurance on a range of QoS metrics including latency, throughput, availability and security. An example of an SoS assimilated from the interaction of multiple systems used by government agencies to facilitate the distribution of real-time national air surveillance data among these agencies [14].

Another example of such transformations in enterprise computing originates from massive integration of sensing, analytics and control capabilities of electronic devices and information systems. This integration of sensing data from real world events into information system leads to an evolution of cyber

physical systems (CPS) [2] and the Internet of Things (IoT) [15-17]. CPS and IoT have been developed for a wide variety of application domains ranging from the smart electricity grid to environmental monitoring and to intelligent transportation. Sensor event data of high volume and control commands of critical importance need to be transported between distributed sensors and actuators and backend enterprise servers for complex event processing and integration with the business processes. Sensor data and control commands are often time-sensitive in that the correct data and command may become the wrong ones, if they cannot be delivered in time. For mission critical applications such as air traffic navigation and monitoring, failing to meet such stringent time constraints may lead to massive error in control decisions or even catastrophic consequences. On the other hand, sensors and actuators are often distributed across geographic locations and utilize heterogeneous communication networks. Middleware such as MOMs are a natural technology for integrating CPS or IoT. Such characteristics require MOM to effectively address the QoS requirements of CPS and IoT.

The resilience and quality of service (QoS) of the messaging substrate plays a critical role in the overall system performance as perceived by the end users. While these current systems and standards provide essential features of reliability, security, transaction and persistence, there is little consideration for real-time QoS such as end-to-end latency and resilience. Also, they are typically deployed within one or a few well-connected data centres. Motivated by the gaps between current MOM system and the application requirements, this thesis presents the design and validation work on facets of a hierarchical overlay-based messaging system that can enhance the resilience of messaging service and manage the end-to-end QoS in wide-area communications based on the application requirements. For convenience the targeted MOM management framework is given the label RQMOM (Resilient and QoS-aware Message-Oriented Middleware Management). To address the needs for federated messaging over wide-area networks, such as to integrate

separated enterprise applications, and distributed data centres or sensors and actuators with back-end processing capabilities, RQMOM is designed as a hierarchical federated overlay of a few separated local domains. Each bottom level local domain comprises one or a cluster of neighbouring broker(s) that perform a local messaging service, while inter domain messages are routed through the gateway brokers in the local domains over the top level federated overlay. The resilience and real time QoS is enhanced by a set of functions that provide local domain workload allocation, load balancing and overlay level routing. These are discussed in the following chapters.

1.2 Research Contributions

The research focuses on resilience and real time QoS in MOM messaging services. Here, resilience is defined as the provision and maintenance of an acceptable level of service in the face of faults and challenges to normal operation, such as broker faults, burstiness of workload, and substrate network congestion or failures. The major contributions of this research are summarized as follows:

At the local MOM domain level:
A novel risk-aware workload allocation mechanism is implemented to allocate workload among local domain brokers. The mechanism quantifies the risk of overloading processing capacity of MOM as probability values in a novel way. It quantifies saturation of bursty high volume throughput, which causes messaging delay degradation, into probability values. This notion of quantified risk exploits the variation and co-variation between the peak-period messaging volumes of different workloads. The approach avoids allocating highly correlated workloads. These would bring bursty high volume messaging traffic that degrades a broker's processing capacity and increases risk of bottlenecks. Both a branch-and-cut algorithm and a genetic algorithm (GA) approximation are designed to solve this problem. This risk-

aware allocation approach is shown to consistently make improvements over Weighted Round Robin allocation and reduce the risk of overload. This workload allocation mechanism is also novel in its pro-active nature within the local domain system. It is designed to work together with existing workload balancing functions in brokers, as an enhancement of resilience performance. The workload allocation mechanism aims to reduce the probability of overloading a MOM system. Overloading reduces broker performance and introduces load balancing overhead. However it is not possible to always eliminate it. Hence, if the messaging system is nearly overloaded a traditional load balancing approach in place should be activated to offload loads.

At the federated MOM domain level:
Coping with the dynamics of the underlying substrate is vital to provide a resilient overlay networking for federated MOM systems. In the past, different overlay routing methods have been researched to support QoS requirements of applications and bypass congestion and failed components.
The focus of this work is to provide resilient overlay networking in the face of underlay substrate failures – in particularly, geographically correlated failures that can vastly impact the substrate. This thesis presents the design and evaluation of a novel overlay routing method that selects end-to-end multi-path set considering the geographical proximity between paths. The novel multi-path routing method developed here calculates multiple paths that satisfy different constraints while ensuring the prescribed geographical distance metric between selected paths. As a result, it provides multi-path selection with better performance to survive in geographical correlated failures. Hence it allows the overlay routing to provide a resilient networking for federated MOM systems over WAN.

A list author's selected publications are as below, which cover the above contributions and other related research. A fuller list of the author's

publications can be found in the appendix.

1. Jinfu Wang, John Bigham, Bob Chew, Jiayi Wu "**Enhance Resilience and QoS Awareness in Message Oriented Middleware for Mission Critical Applications**", 2nd Symposium on Middleware and Network Applications, April 11-13, 2011, Las Vegas, Nevada, USA
2. Jinfu Wang, John Bigham, Beatriz Murciano "**Towards a Resilient Message Oriented Middleware for Mission Critical Applications**", The Second International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE 2010), November 21-26, 2010 - Lisbon, Portugal
3. Habtamu Abie, Reijo Savola, Jinfu Wang, and Domenico Rotondi (2010): "**Advances in Adaptive Secure Message Oriented Middleware for Distributed Business Critical Systems"**, 8th International Conference of Numerical Analysis and Applied Mathematics**,** (ICNAAM 2010), 19-25 September 2010, Greece
4. Jinfu Wang, Peng Jiang, John Bigham, Bob Chew, Beatriz Murciano, Milan Novkovic, Ilesh Dattani, **"Adding Resilience to Message Oriented Middleware"**, in Proc. Serene 2010 ACM, 13-16 April, 2010 London, UK
5. Jinfu Wang, John Bigham, "**Anomaly detection in the case of message oriented middlewar**e". 1st International Workshop on Middleware Security (MidSec 2008), December 2, 2008 Leuven, Belgium.

1.3 Organisation of the Thesis

The thesis is organized as follows. Chapter 2 introduces the background of MOM systems, empirical deployment scenarios and reviews related work; Chapter 3 reviews the system architecture and the building blocks to support resilience in both local domains and a federated overlay domain. Chapter 4

presents the local domain resilience mechanism that combines proactive risk-aware work allocation and mirroring of workload and reactive dynamic load balancing. Chapter 5 first explains the challenges and routing techniques employed on the federated overlay domain. Then it introduces the design of a novel proximity-aware overlay routing mechanism and evaluates its performance in a set of geographical correlated failure scenarios. Chapter 6 concludes the thesis and discusses possible extensions of this work.

Chapter 2 Background and Related Work

2.1 Message-Oriented Middleware and Alternative Communication Paradigms

2.1.1 Basic Communication Patterns in Distributed Systems

Distributed systems often involve many distributed entities whose behaviour and location may greatly vary throughout the lifetime of the system. Individual *point-to-point* and *synchronous* communications lead to rigid and static applications, and introduces repeated efforts and overhead to such distributed systems in both run time and development. Hence a dedicated middleware infrastructure with adequate and standard communication schemes is often preferably employed as the glue in distributed systems. Depending on different applications, the message producer and consumer in distributed systems may require decoupling in three dimensions, viz space, time and synchronization.

In geographical space decoupling, the producers do not need to know the consumers participating in the interaction. In time decoupling, the producers and consumers do not need to be actively participating in the interaction at the same time. In synchronization decoupling, the entities do not block themselves from their main flow of control while producing and consuming messages.

The communication schemes employed in distributed systems are as follows, message passing, remote procedure call (RPC) and derivatives, message queuing and Pub/Sub. Message passing represents a low-level form of distributed communication, in which participants communicate by simply sending and receiving messages. The producer sends messages asynchronously through a communication channel (previously set up for that purpose). The consumer receives messages by listening synchronously on that channel. The producer and consumer are coupled in both time and space.

Message passing is nowadays viewed as a primitive to build complex interaction schemes and rarely used directly. RPC is a widely used form of distributed interaction proposed in [18, 19] for procedural languages, and in object-oriented contexts for remote method invocations, for example, in Java Remote Method Invocation RMI [20], The Common Object Request Broker Architecture (CORBA) [21, 22], Microsoft Distributed Component Object Model (DCOM). RPC makes remote interactions appear almost the same way as local interactions. Although it is be made transparent to the application, because further types of failures e.g., communication failures, exceptions raised and the different semantics associated with parameter passing need to be handled by applications, the transparency is in fact partial. Originally in RPC the producer performs a synchronous call that is processed asynchronously by the consumer. Some derivatives of RPC such as CORBA make changes to producer behaviour such as not to expect a reply or only to retrieve a reply when needed.

The conceptual design of Message Oriented Middleware, is first introduced by IBM researchers in [23] to glue loosely integrated applications. In contrast to method invocation centric RPC, message queuing [24] and pub/sub communication patterns are message centric communication schemes. Message queuing and pub/sub are tightly intertwined paradigms. In these message centric communication schemes, middleware mediate the asynchronous communication between end applications. The end applications are known as *producer* and *consumer* in point-to-point message queuing pattern as in figure 2-1, or *publisher* and *subscribers* in Pub/Sub pattern as in figure 2-2. In both patterns middleware manages one or multiple queues to process and disseminate information. Message queuing middleware buffers the produced messages in queues while supporting, when required, reliable delivery, transactional and ordering guarantees. In the point-to-point message queuing pattern that is shown in figure 2-1, a message producer sends messages to a dedicated queue provided by middleware and one message can only be fetched by a single consumer. To

support multiple applications, middleware must set up an individual queue for each pair of communicating applications.

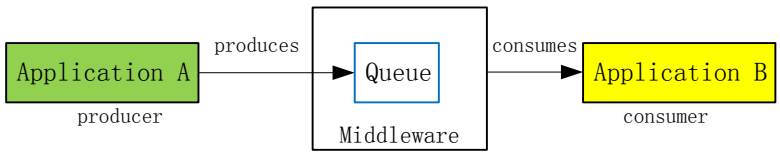

Figure 2-1Point-to-point queuing messaging pattern

Pub/sub supports many-to-many communication, and the communicating applications are usually referred as publishers and subscribers. In pub/sub as illustrated in figure 2-2, middleware keeps the registered interests (a.k.a. subscriptions) of subscribers according to the topic (for topic-based pub/sub, such as [25, 26]) or matching conditions defined by subscribers (for content-based pub/sub, such as [27, 28]); and dispatches a message to all subscribers that match with their registered interests.

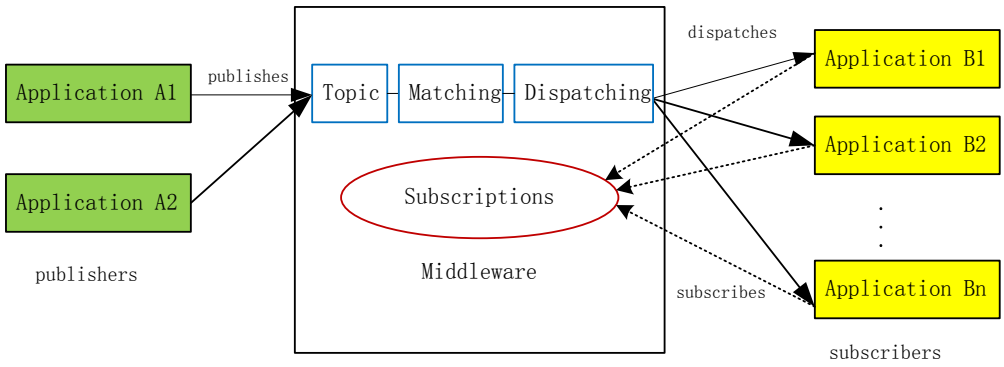

Figure 2-2 Publish and subscribe messaging pattern

Many current MOM systems employ a fusion of topic and content based pub/sub, in which messages are first sent to topics and then can be further filtered or matched according to requirements of subscriptions as shown in figure 2-2. In figure 2-2, two publishers *A1* and *A2* are publishing to a topic on the middleware, which is subscribed by multiple subscribers *B1*, *B2*, …, *Bn*. MOM adopts the message centric approaches and usually employ both message queuing and pub/sub communication schemes. In section 4.4, the basic internal components of MOM brokers and performance models are

further described.

2.1.2 Variations in MOM Standards

Existing well-known MOM standards fall into several categories. Despite there being different MOM standards they have evolved to support some common functions such as transactional guarantees, and Java Messaging Service (JMS) support. Two main representative classes are enterprise messaging systems and real-time data distribution systems. Two examples of the most widely used MOM standards are Advanced Message Queuing Protocol (AMQP) [29] and Data Distribution Service (DDS) standard [30].

Intended to address traditional business needs, enterprise messaging systems, such as AMQP, provide message delivery assurance, transactional guarantees, high-availability clustering, etc. It also supports federation of multiple domains over WAN or internet to transport inter-domain messages. However, they do not have much support of providing temporal QoS. The unsupported features such as temporal QoS will require clients to customise solutions at the application level. Besides the business sector such as financial markets, AMQP is also widely employed in other scenarios, such as internet applications and cloud computing.

Real-time data distribution systems, on the other hand, were originally designed to target real-time traffic publish and subscribe, e.g. sensor and actuator, military and defence system. They often conform to the DDS standard which originally has an emphasis on User Datagram Protocol (UDP) transport layer, and offer temporal QoS assurance by allocating resources and scheduling messages based on application-specific QoS objectives. However these systems are limited to QoS management within a local area or a single domain. QoS management over wide-area messaging involving multiple separate domains is still a topic under research [31].

On the other hand, JMS is also a widely used different standard [32]. It defines generic behaviours of components in a messaging system with an

application programming interface (API). These behaviours include the basic structure of a JMS message, how publishers and subscribers generate or receive messages, filtering options to describe subscriptions, etc. The message mediation server and network and transport layer is not specifically defined by JMS. JMS is an application-specific communication layer standard. It needs to be supported by other MOM standards or implementations, e.g. AMQP.

In contrast, RQMOM considers a hierarchical federated MOM architecture which is by extending this intra domain messaging scenario to a federated overlay on WAN, internet scale or cloud scale applications. This architecture is introduced in the next chapter. Hence RQMOM is more similar to federated enterprise messaging scenario as described above, which is designed for inter domain messaging to connect, for example a few federated cross-country data centres of different organisations and integrate remote resources and sinks to enterprise application domains over WAN or internet. The RQMOM messaging system is designed to address the resilience challenges of in the hierarchical federated MOM architecture. The following Chapter 4 and 5 focus on a risk and QoS aware workload allocation problem in local domains to pre-emptively prevent correlated bursty workload overloading the broker processing capacity, and geographical separation aware multi-path routing in the federated overlay domain to provide resilience against geographical correlated failures.

2.1.3 Related Works on MOM Systems

In MOM systems both the fundamental messaging services and more fine-grained QoS mechanisms like those provided by DDS standard is built upon the healthy processing capacities of MOM brokers. However in the practical world, performance of MOM is vulnerable to processing power degradation caused by excessive workload or different unexpected faults and failures. Load balancing and availability consolidating mechanisms are important to

cope with these challenging situations.

Load balancing has been widely researched in distributed computing systems. The goal of all load balancing solutions is to distribute workload to all available resources in a way that maintains the system health and avoids or mitigates overloading a single processing component. Load balancing often refers to either relocating current workloads to different servers in a system, or mapping newly arriving workload or newly available server into existing system. Load balancing are applied in four layers: network [33], operating system [34], middleware [35-39], and application [40, 41]. This thesis investigates how to balance workload of MOM at the middleware layer. Cheung et al. [42] developed a fine-grained load balancing in content-based pub/sub systems. In their work, load balancing is triggered by surges in one of these measurements: input utilization, output utilization, CPU usage, memory usage or matching delay. The effects of different pieces of workload on system performance are modelled, so as to estimate the performance changes among brokers if a piece of workload is relocated. Their work is targeted on high performance data centres and their own content-based MOM implementation. In other settings load balancing may be applied with a different granularity. For example, in a peer-to-peer (P2P) pub/sub system Meghdoot et al. [28] balance load by simply splitting the heaviest loaded peer in half and allocating to its new joining neighbour peer; *Opportunistic Overlay* [43] proposed a dynamic overlay reconstruction algorithm that performs load distribution on the CPU utilization when a client finds another broker that is closer than its home broker. Subscription clustering [44-46] divides the set of subscriptions into a predefined number of clusters or groups to reduce overall traffic on the network. It is a popular approach to balance content-based pub/sub systems. In recent studies, [47, 48] divide the multi-dimensional information space into partitions, redundancy and reactive load balancing is provided on the level of partitions of workload. In a similar spirit to these approaches, in RQMOM, workload is partitioned into items to be allocated.

Items can represent a portion of workload that is partitioned with any arbitrary method chosen by administrators. Different from RQMOM, none of the prior solutions have considered the statistical correlation between workloads. In order to maintain high performance and to maintain a stable and healthy system, RQMOM pre-emptively allocates workloads to minimize the risk of overloading the system, and reducing the frequency and overheads of load balancing. In a local domain, this is achieved by allocating the total workload to edge brokers in a way to minimizing the probability of overloading brokers while exploiting the variation and co-variation between workloads. However, when unexpected heavy loads do occur due to the dynamics of the system, a subset of suitable subscriptions are offloaded (the common approach of broker mirroring) so that both the offloading broker and the load-accepting broker are more healthily balanced.

Availability of MOM often refers to the portion of time over system's life span during which the messaging service is provided to functioning clients. High-availability (HA) often refers to mechanism to provide minimum system down time, and in some cases also considers QoS agreements such as exact once delivery of messages. Traditional HA clustering is often supported by MOM implementations such as AMQP MOMs. It means system state and data store are replicated among multiple instances of the same broker, and once a failure is detected, the faulty broker failover to redundant replicas. Meanwhile clients are masked from the internal faults of HA cluster. Besides traditional HA clustering, availability through overlay of brokers has been researched in [49-56]. In these approaches system-wide subscription state are replicated among overlay, and upon failures through reconfiguration of brokers and overlay paths, the clients remain connected to messaging service. Reliable message delivery using pub/sub overlays has been considered in [51, 57]. These papers also focus on an exactly-once, in-order, guaranteed delivery service. The approaches in [51, 57] are taken as baseline services of enterprise MOMs. In RQMOM's hierarchical architecture, the important gateway brokers of local domains are often consolidated with traditional HA

clustering. Within a well connected network of a local domain, subscription information is shared among edge brokers, so that upon failures, clients can failover to remaining live brokers. In addition RQMOM use a risk-aware workload allocation algorithm to calculate solutions that mirrors redundant workloads pre-emptively, so that upon failovers the load accepting broker has a low probability to be overloaded by incoming clients.

2.2 *Overlay Network*

2.2.1 Overview of Overlay Network

Overlay network paradigms give federated MOM systems leverage to improve the messaging performance. The common characteristics of overlay network are the presence of two or more layers of networks, and that upper-layer overlay nodes can select routes and forward packets independently from the underlay layer nodes. Hence overlay networks have the appealing feature that additional features such as routing can be independently designed and applied in an application-specific manner without modifying the underlying IP routing scheme. Some key issues from different aspects of overlay network are shown in figure 2-3 below. These issues of overlay network, and their relation to this thesis are reviewed in the following paragraphs. The blue text highlights the issues related to the research in this thesis, which will be further discussed in Chapter 5.

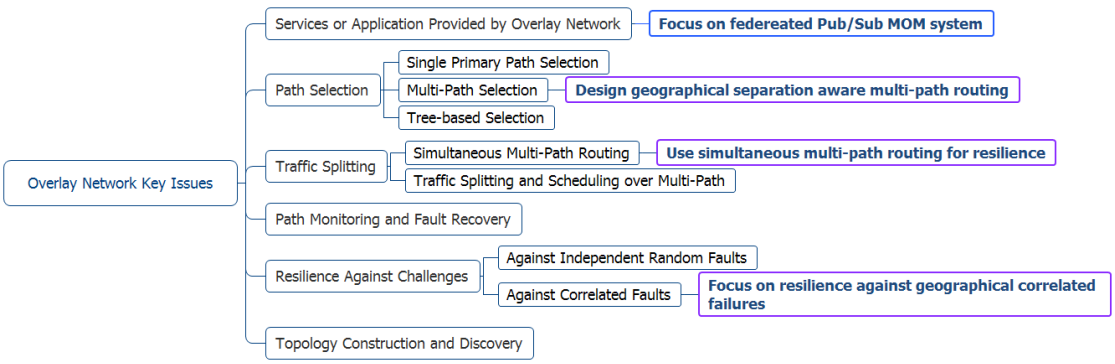

Figure 2-3 Overlay network key issues and the focuses of this work

The benefits of overlay routing began to draw wide interests from experiment based studies such as Resilient Overlay Network (RON) [58] and the analysis of the benefits of overlay routing against direct routing in terms of latency performance, bandwidth and resilience in studies such as [59]. Although RON uses a simple link state path selection in the overlay network, benefits from these studies inspired more overlay routing mechanisms to customize overlay network researches for different purposes.

Overlay routing has been widely used for both performance and resilience purposes. Overlay routing is customized in various settings to address different scenarios. RON [58] uses latency, packet loss rate, throughput as metric to monitor and select overlay paths, so as to recover faster from internet path outages and performance degradations. To enhance performance other works may use other metrics, such as available bandwidth between end-hosts [60]. To provide more resilient networking service, many different models of challenges, besides random single failures and random multiple failures, have been considered. These include [61], which selects overlay back up paths basing on the statistical correlation between links; and [62] designed a simultaneous multipath routing according to a specific risk model for earthquakes. Inspired by these works a part of this thesis focuses on multipath routing against generic geographical correlated failures. This will be described in Chapter 5.

From the aspect of applications or services provided, overlay networks are researched to solve existing problems such as fault localization [63] and mitigating DDoS (Distributed Denial-of-Service attacks) [64, 65], and providing new applications such as Peer-to-Peer (p2p) [66] network and pub/sub messaging service [26, 67]. RQMOM presented in this thesis focus on publish and subscribe messaging service provided by an overlay of federated MOM brokers.

2.2.2 Overlay Routing

In contrast to direct routing on the substrate network level, overlay level networking can apply application specific techniques and strategies without changing the standard underlay infrastructure. Overlay level routing paradigms can be dimensioned roughly into different categories: best path routing, multipath routing and tree-based overlay routing. Multipath routing is used in this thesis.

End-to-end best path routing as employed in overlay routing are of a selfish nature, as they greedily choose paths that offer the highest performance, regardless of the implications on the performance and stability of the whole network. Many researchers have addressed overlay routing from the aspect of system wide performance, e.g. [68, 69]. However its optimization solution is computed by centralized algorithms using a global view of the network. For local decisions, one practical approach is to only consider the limited scope of information that a node can obtain from measurements of its links. In such a case, Seshadri and Katz [70] make suggestions to improve the overall stability of the system by imposing some restraints on the degree of selfishness of each node, such as applying constraints and randomization to path selection. These mechanisms were shown to improve the overall system stability in their research. From a theoretical aspect, user-optimal or selfish routing achieves a Wardrop equilibrium [71, 72], which states that users do not have the incentives to unilaterally change their routes. Xie et al. [73] present a routing scheme that takes into account user-optimal routing and network-optimal routing, where the former converges to the Wardrop equilibrium and the latter to the minimum latency.

The term multi-path routing has been used with different connotations. In all of them multiple paths are used from the source to destination over which traffic is transported. The first interpretation of multi-path routing is to

increase the resilience of the network, by simultaneously transmitting duplicates of the same packet over each path. This technique is also often used in wireless ad-hoc networks [74] and it is sometimes referred to as redundant multi-path routing or simultaneous multi-path routing. A performance comparison between simultaneous multi-path routing via one random intermediate overload node and probe-based reactive best path routing is studied in [75]. In their study, redundant multi-path provides comparable but marginally better performance.

Another way of using multiple paths is by distributing the traffic volume over a set of paths. Although by introducing this path diversity the routing is made inherently more robust to failures of individual links, its purpose is rather on improving performance by using multipath data distribution techniques. An issue in multi-path routing is the issue of packet reordering, as some packets may overtake each other on different paths. The destination node must buffer the received packets and place them in the right order before delivering them to higher layers. In the work presented in Chapter 5 of this thesis, simultaneous multipath routing is used, which that replicates messaging along a multiple paths.

To distribute traffic among multi-paths, traffic splitting and scheduling along multi-paths are commonly employed. Optimal multipath data transfers on overlay networks have been studied in [76-79]. In [78] Tao et al. demonstrated the potential of multiple overlay paths for improving Internet performance. They developed a path switching mechanism that subsequently selects a potentially less congested path among several available paths to transmit data. A drawback of this approach is the overhead caused by active measurements and the accuracy of the path model. Moreover, in many cases, sending data always to the less congested path is not the optimal policy. In [79], a multipath controller is proposed that randomly chooses a transmission path following a throughput-proportional selection scheme. This path selection scheme is a randomized version of the well-known Weighted Round

Robin (WRR) scheme [80] where the weights, in this case, are the path throughputs. A key issue of this scheme is how to accurately obtain the path throughput, by means of online measurements. However, it is shown that, even with the correctly estimated path throughput, WRR is unlikely to be an optimal scheme for QoS metrics, such as loss, delay and jitter [81]. In wireless scenarios, the work [82] presents an algorithm, borrowed from wireless scheduling problems [83], to distribute packets on different available paths while minimizing delay, maximizing throughput, and respecting the split proportions assigned by the routing algorithm. In the context of data replication and collection, the authors of [76] and [77] studied the problem of multipath data transfer, where the objective was to minimize the makespan, i.e., the duration of the longest transfer. Both studies formulated the problem by using graph theory, where the graph is the overlay network topology, and the constraints are the path capacities. However, this approach has a number of potential issues, as admitted by the same authors [76]. First, the graph theory formulation assumed that the time required to transfer a unit of data through a link is fixed, which is unrealistic because in real networks, in the presence of background traffic, this transfer time varies. Secondly, since end-to-end delay of a path is not always proportional to its capacity, using path capacities as constraints to minimize the makespan may not lead to an optimal solution. In such a dynamic environment, a promising approach is based on a dynamic optimization framework like Markov Decision Processes (MDPs). Markov Decision Processes [84] are a powerful mathematical framework for making decisions in a dynamic environment that exhibits Markov property. [85, 86] propose MDP based approaches to make data forwarding decision over multiple paths, to minimize the average networking delay. These multipath routing mechanisms aim to optimize the average latency experienced by a single application.

As alternatives to the above work on traffic scheduling and splitting, some interesting works apply biological self-adaptive mechanisms to multipath routing. A well-known, biologically inspired technique that is very efficient

for routing is AntNet [87], which uses mobile agents that mimic the behaviour of ant colonies. It operates by sending forward ants to probe routes and backward ants to update the routing tables at each intermediate node. Traffic is then routed along the paths with certain probabilities. Another biologically inspired approach is [88], which considers a problem different from AntNet, in that it focuses on the adaptive selection of already determined paths. [88] proposes to solve the problem of multi-path routing in overlay networks by adapting the transmission of data packets to changes in the metrics of each path with adaptive response by attractor selection, which is analogous to the cell in a gene network switching from one state to another depending on the availability of a nutrient.

2.2.3 Other Issues

Some aspects of overlay networking are outside the main focus of this thesis, however they still need to be considered for a compete overlay networking solution.

Network monitoring is one common aspect in overlay networking. Most overlay network applications use active periodical probing and passive measurement to monitor the performance. RON uses a fully connected mesh topology with active probing to monitor and react to network dynamics. This approach is suitable to smaller size network scenarios, such as the overlay federation of a moderate number of local enterprise messaging domains, which is the focus of this thesis. However it will incur significant monitoring overhead in much larger scale networks. In [89], the authors discuss the relationship between the effectiveness of an overlay and its overhead consumption. They conclude that a comparable QoS performance to that of full connectivity can be achieved with a topology of less node degree. This also leads to the problem of how to construct overlay topology.

Topology construction researches are can be categorized basing on different

problem settings. In the case of constructing topology over a given static network a few settings are: 1. fixed-node topology construction[90, 91], which means given predetermined overlay nodes, the links between nodes need to be chosen to construct the topology; 2. Selecting or placement of overlay nodes from a set of candidates [92, 93], considering utility of nodes according to their characteristics such as underlay level diversity provided by an candidate node; 3. Constructing the overlay topology given the requirements described as a traffic matrix. Another case is adapting a dynamic overlay topology [94, 95], against changes in either underlay network performance or the overlay elements such as leaving and entering of nodes. The overlay network considered in this thesis is a hierarchical federation of enterprise MOM systems, which is usually of a smaller scale overlay formed only by gateway brokers of local domains in relatively stable environments. Hence we can consider a full mesh or other pre-determined known topology. For completeness some of these topology construction approaches are still further reviewed in Chapter 5.

2.3 Summary

In this chapter the background of MOM has been introduced. The paradigm shift of communication between applications from simple message passing and RPC to pub/sub middleware is reviewed. The federation of these pub/sub middlewares usually constitutes a federated overlay providing messaging service. In the overlay of brokers inter-connecting multiple local MOM domains, overlay networking mechanisms can be often applied to enhance networking resilience and performance. Related research literatures on overlay network have been reviewed.

Comparing the related works in 2.1 and 2.2, for inter-domain federation, RQMOM constructs a hierarchical overlay network of federated gateway brokers on top of the physical topology, which can be viewed as a message-centric Service Oriented Network. On this federated overlay RQMOM can on

one hand, in the local domain preemptively allocate workload with risk-awareness to ensure healthy processing capacity of brokers in order to satisfy the real time QoS requirements of different applications; and on the other hand, in federated overlay of brokers the messaging can be made resilient via geographical distance aware path selection and adaptation when either major network outages or excessive delays occur along delivery paths. In particular this routing mechanism is tested and compared to conventional path selection mechanism against geographical correlated failures. Based on the literature mentioned above, the next chapter starts out to further introduce the hierarchical architecture of the federated MOM system that is considered in this thesis. Then the research goals and novel work under this architecture is illustrated.

Chapter 3 System Architecture and Resilience Building Blocks

This chapter outlines the proposed MOM architecture and also the solution adopted to enhance resilience messaging service in both the local domain and federated overlay architecture.

3.1 System Architecture

The system structure of RQMOM is designed as a hierarchical overlay of a few separated local domains. Each bottom level local domain comprises a cluster of interconnected neighbouring brokers. Local messaging services are carried among these brokers. Meanwhile inter domain messaging are routed through the gateway brokers of local domains over the top level federated overlay as illustrated in the figure below.

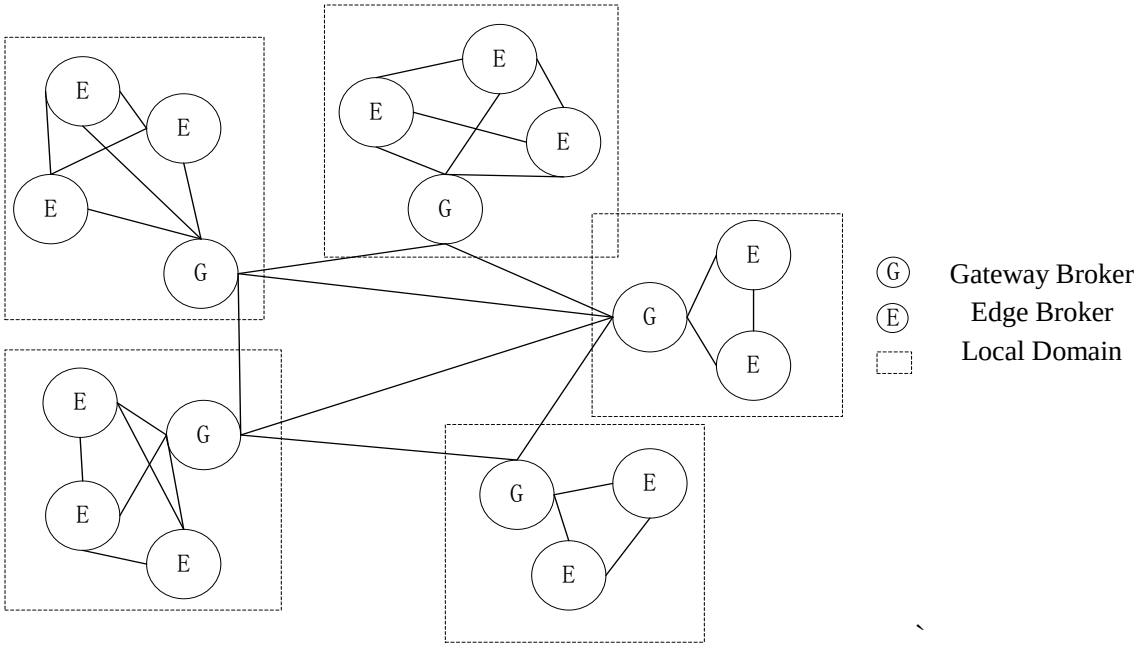

Figure 3-1 RQMOM's hierarchical structure of federated brokers

End nodes are not shown in the above figure. End nodes are messaging clients that are connected via the above messaging infrastructure such as a message processing application, a message producing application, a sensor, or

an actuator. Each end node is attached to at least one broker, which normally are edge brokers in its local domain, unless it is specifically configured to attach to a gateway broker. Intra domain messaging workload is normally served among local brokers. On the other hand, the inter domain messaging service is provided on the overlay network level. Each gateway broker works as an overlay node and routes the messages along the overlay network according to routing protocols employed. There can be an arbitrary number of topics in the system, which can be defined either through administrative tools or dynamically using programming APIs. Each endpoint can publish and subscribe to one or multiple topics, while each broker can perform publish/subscribe matching, can transport messages to local endpoints or neighbouring brokers, and optionally perform message mediation, e.g., format transformation using Extensible Stylesheet Language Transformations (XSLT). Depending on different security policies, topics can be accessed only by authorised and authenticated users. For example some topics in the system could be authorized for access only inside a local domain, while other topics are accessible on the federated domain. However, the issues of authentication servers and other aspects such as confidentiality and integrity are out the scope of this work. Each endpoint application can subscribe to any authorized topic at any time. (This could be generalized to include more sophisticated authorization schemes without affecting the approaches described here.) Such subscriptions are sent to a local domain broker that this endpoint is attached to. The subscription and publication coming from and destined to the local domain go through the local broker. Each broker maintains a *local subscription table* to record which topics each local endpoint subscribes. The topics that are limited within a local domain are always kept only in local tables. If the topics are accessible in the federated overlay domain, then the gateway brokers also propagate these topics to other brokers. As a result, each local domain knows which topics any other federated domain needs; it maintains such information in a *remote subscription table*. When an endpoint publishes a message on a topic, say *t*, the message is sent to the local domain

broker. This broker first checks the local subscription table and transmits to all local subscribers of *t*. If *t* is also subscribed to in the remote subscription table, the message will be forwarded to routes through the local gateway broker to find all remote domain gateway brokers that subscribe to *t*, and sends the message to these brokers using the federated overlay. Local gateway brokers are often designed with redundancy to provide extra resilience, with some replication or mirroring strategies. Upon receiving this message, these gateway brokers further forward it to their respective local domain subscribers. As such, the message will eventually arrive at all subscribers of topic *t* in the federated system.

The hierarchical structure approach is an adaption to practical MOM deployments in real world application scenarios. The separated local MOM domains are deployed in geographical separated data centres or organizational separated enterprise infrastructures. Hence each local MOM domain is often within a well-connected underlay network, and different local MOM domains are inter-connected into a federated overlay domain, likely with dedicated network links. Such hierarchical approach has some architectural merits as follows:

– *Scalability*: In each local domain subscribers can be assigned to different local brokers, so that the message processing and forwarding work in local domain can be shared among a number of brokers. The gateway brokers can be configured to be dedicated to process the messaging in the federated overlay domain, and each gateway broker only communicates with a small number of neighbouring brokers and hence avoids maintaining too many pair-wise connections, which would be prohibitively expensive as the system scales up in a federated overlay.

–*Federation*: The system is likely to be deployed and operated jointly by multiple organizations. In such a federated scenario, it is critical that each administrative domain can independently manage the access from and to its own nodes, which can be easily facilitated by the local brokers.

– *Heterogeneity*: The endpoint applications and sensors are inevitably heterogeneous in a large-scale system. The MOM brokers of the federated messaging system use unified protocol, while the local brokers use a few adapters to communicate with the local domain application or sensor endpoints, instead of understanding all the protocols used in different domains.

As previously mentioned, the focus is on providing real time QoS assurance within the whole broker overlay network. In the next subsection, the QoS model employed in this work is elaborated. Despite the architectural merits, there are challenges to build a resilient MOM system providing QoS assurance in both local and overlay domains. The challenges to be addressed, and the approaches to meet each challenge are introduced in the following section.

3.2 Research Goals

Providing predictable QoS is an essential requirement for mission-critical applications. In particular, the messaging middleware should ensure timely and reliable delivery of critical messages, such as emergency alerts or real-time control commands. Formally stated, the goal is to provide QoS-aware publish/subscribe service in terms of *message processing latency and uninterrupted delivery between all matching pairs of publishers and subscribers*. Note that the end-to-end delay for a given message consists of the cumulative processing delay at each intermediate broker and the communication delay between adjacent brokers. The former is affected by the load (i.e., message arrival process) of a broker, while the latter is affected by the characteristics of the network links. The broker processing delay also varies over time as each broker dispatches messages on multiple topics and the messages may arrive in bursts. Specifically, since local networks are characterized with very low delay and high reliability, in the local domain messaging latency variation is dominated by the processing latency of brokers. In the face of faults and

challenges to normal operation, such as internal broker faults or burstiness of workload, our system seeks to maximize the probability that brokers work within their processing capacities and hence avoid possible performance degradation. Furthermore, on the federated overlay domain since the different domains are deployed over a large geographic area, they will inevitably operate over wide-area networks, where the network connections are subjected to external substrate network congestion or failures. The MOM should also enable resilient messaging across local domains, which means to strive to provide uninterrupted delivery of messages and maintain an acceptable level of service. In particular this work focuses on mitigating geographical correlated failures. Geographical correlated failures may cause large scale interruption to underlay substrate networks, however are not well studied. Some other works rely on underlying network to provide dedicated resilient assurance and redundancy mechanisms. In contrast this work assumes a more a general underlay model, none of the above underlay network supports is required. Also this work does not require the specific knowledge of historical failure statistics information, such as the failure probabilities on underlay network nodes or edges. Such relaxed model allows our system to be applicable in different deployment scenarios.

3.3 Building Blocks for Resilience and QoS

3.3.1 Building Blocks in the Local Domain

In a local domain scenario, the cluster of MOM brokers is usually deployed in a well-connected network, e.g., a LAN within a data centre, where network latency is normally small and very predictable. On the other hand, the dynamics of the delay for a given message is commonly in the processing delay at the intermediate local broker that the subscriber is connected to, and this is affected by the load (i.e., message arrival process) of a broker. The broker processing delay varies over time as each broker

dispatches messages on multiple topics and the messages may arrive in bursts. If large messaging volumes simultaneously arise in several topics then the arrivals are likely to exceed the threshold volume that a broker can handle with a normal processing latency. This means the broker is overloaded and the messaging service will experience larger than average delay or even losses if the input message queue is full.

Although some research has provided mechanisms for load balancing between brokers, such post hoc load balancing mechanisms provide mitigation after an overload occurrence is measured and detected. However to our knowledge few previous works have addressed how to allocate messaging workload within local domain, such that the system wide message processing latency is smaller and more stable, and hence less post hoc load balancing effort is needed. Another building block for a resilient MOM system is that some critical topics may need extra redundancy, (a.k.a. mirror of a topic) – in case of a broker fault or failure – either on-demand or automatically. However when an unexpected heavy load occurs due to the dynamics of the system, our load balancing building block chooses to offload a subset of suitable subscriptions so that both the offloading broker and the load-accepting broker are healthily balanced.

Hence to address the above problems on local MOM domain level, a novel risk aware workload allocation and mirroring algorithm for local domains has been designed and implemented. In a local domain, the message volume of different topics, often have some correlated predictable patterns. Our algorithm quantifies the risk of overloading a broker, processing speed degradation in brokers caused by saturation of high volume throughput. The notion of quantified risk exploits the variation and co-variation between the critical peak-period messaging volumes of different workloads, and avoids allocating highly correlated workloads that would bring bursty high volume messaging traffic that degrades a broker's processing capacity. Additionally

the allocation algorithm is used to find the solution of where to place mirrors of critical topics - prior to or on the occurrences of faults in an original broker carrying the critical topics, without introducing too much risk of overloading the system.

Note that the focus of this risk-aware workload allocation mechanism is maintaining a stable and near optimal message processing capability at every MOM broker. In other words, the mechanism aims to avoid the performance degradation or faults caused by overloading the maximum processing capacity of a MOM broker. The cause of degradation or faults can also be described as when total incoming workload exceeds the broker maximum allowed throughput for a stable and fast message service.

However, while limited processing capacity for MOM brokers is a key issue, it is not the only cause of the performance bottlenecks. Some of other possible causes include problems specific to MOM, such as the message store overload, and also generic problems such as software and hardware faults. These are different problems and have been addressed by previous research. Message store overload is caused by the difference between a message producer's publishing rates and message consumers' receiving rates. A fast producer or slow consumer will result in building up a large message store in a MOM broker memory. This will further cause problems such as frequent garbage collection of the Java Virtual Machine (JVM) or other heap memory mechanism. This problem can be addressed by either a traffic control approach, such as throttling the incoming message rates [96]; or by adaptable provision of extra memory resources such as dynamic queuing [97] supported within a broker cluster. Hence, these problems are not the focus of this thesis.

The work on risk-aware workload allocation mechanism in the local domain is discussed in Chapter 4.

3.3.2 Building Blocks in the Federated Overlay Domain

In the local domain the main issue was the processing capability of the broker

and ensuring that the input to the broker did not exceed its optimal capacity. For inter domain messaging the communication delay between publishers and subscribers in the local domain is not a main focus. However, inter domain messaging services need to run over a WAN or internet. Mission-critical and time-sensitive applications often define resilience and QoS requirements, despite the different failure and delay dynamics of the underlying substrate. This motivation is as described in section 1.1. To address this requirement, RQMOM employs overlay routing mechanisms to provide resilience and QoS-aware messaging, on top of the overlay network of federated gateway brokers distributed in separated geographical local areas, on top of the heterogeneous physical network.

In addition to the merits of previous overlay routing paradigms, namely the exploitation of diversity in the network paths and direction of the traffic over link(s) with good quality, this component also focuses on providing resilient overlay networking in the face of different degrees of underlay substrate failures – in particular, geographically correlated failures, which can cause vast impact to the substrate. Driven by this, a novel overlay routing method is designed and evaluated, that selects an end-to-end multi-path set considering the geographical proximity between paths. It will be demonstrated that under certain conditions it copes better with geographical correlated failures that often cause major damage interrupting messaging along many overlay paths. The proximity-aware multi-path set selection is an important component in a range of route maintenance mechanisms, including simultaneous or split-path multipath routing and primary and back-up routing. The novel multi-path routing method developed here calculates multiple paths that satisfy different constraints while ensuring good geographical separation between selected paths according to a prescribed geographical distance metric. As a result, the novel geo-aware alternative path selection algorithm (which is referred to as GAP) is able to find alternative paths with better spatial separation then other common algorithms. Additionally, together with the fusion of route calculation, establishment, and route maintenance strategies, it provides

multi-path selection with better performance to survive in geographical correlated failures. Hence it allows the overlay routing to provide a resilient networking for federated MOM systems over WAN. This work is described in chapter 5.

3.3.3 Integration into a Federated MOM Cloud

There are several cloud based systems that perform a MOM service, e.g. Amazon Simple Queue Service (SQS). A basic cloud platform provides resilience by setting up multiple instances of same applications on demand. However, when a MOM system is deployed on cloud, a basic cloud implementation does not support geographical resilience for uninterrupted messaging service, nor does it mitigate bursty messaging workload instantly. A resilient MOM system needs to have multiple possible entry points to the MOM system so as to be able to route around geographically centred faults or manage surges in demand or denial of service attacks that impinge on the MOM. In the proposed architecture the leverage provided by cloud and Virtual Machine (VM) technologies can be used to support the creation and removal of brokers at the local domain level and the creation of new domains at the federated level. At the local domain level this allows elasticity in that the number of brokers can be increased and decreased readily. The following figure illustrates the architecture.

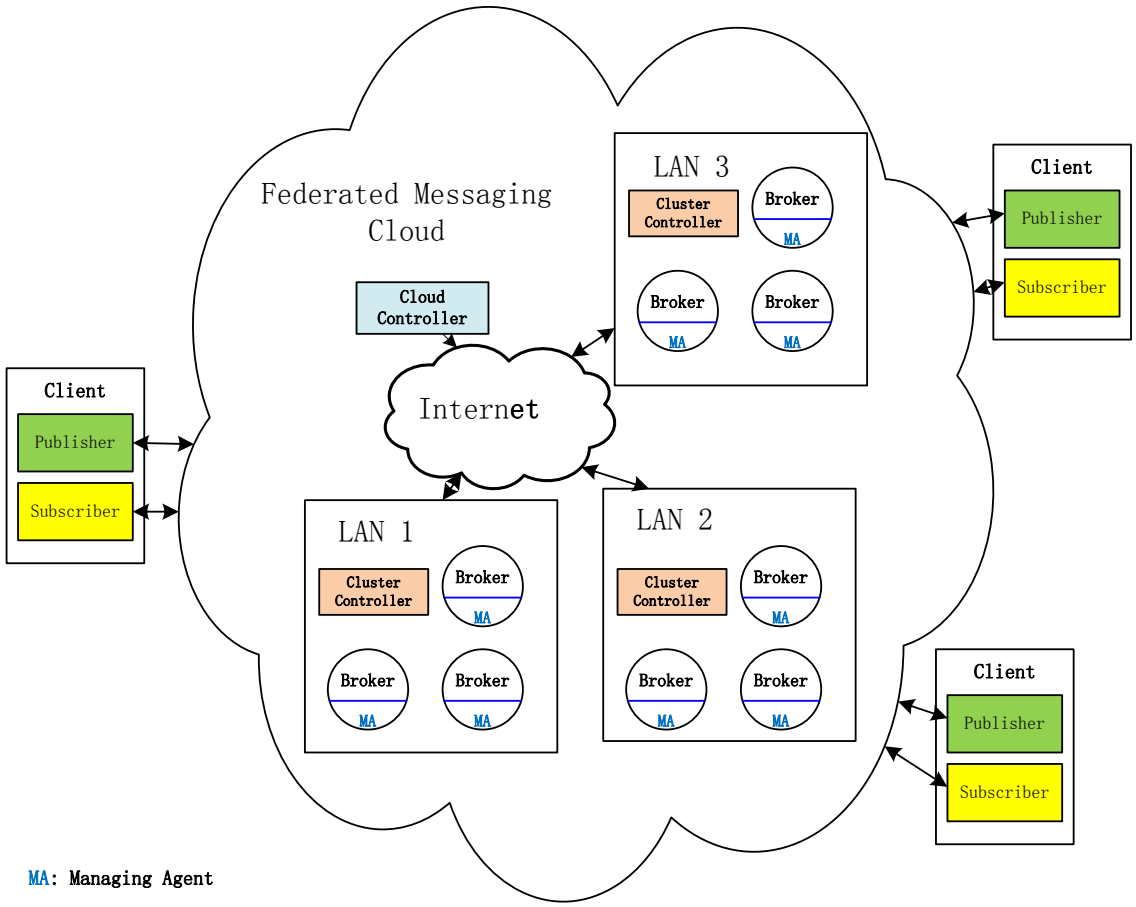

Figure 3-2 MOM architecture in the cloud environment

Each local domain comprised of a LAN or other type of well-connected network. In above figure, the 3 LAN-based local domain MOM system forms a federation on the internet. In the local domains, the each broker is managed by a Managing Agent (MA). In each LAN, the elements such as cluster controller, broker and MA are hosted and managed by cloud in VMs or other containers. Sometimes in this architecture each local domain is part of an individual private cloud system and the federation of these individual local domain clouds is known as cloud-of-clouds [98-100]. Federated pub-sub MOM system is viewed by some research, e.g. [101, 102], as a suitable universal communication backbone in such scenario.

MA's functions will be described in section 4.3.1. In the local domain a cluster of brokers are managed locally by cluster controller to provide VMs that host brokers and managing agents associated with the brokers. Often there would

be only one VM per host unless for specific privacy reasons different customers required different VMs running on the host. (Some brokerage providing companies such as StormMQ charge extra for providing a non-shared broker.) In this case the brokers would not be able to move topic responsibilities between each other, as such a broker comprise indivisible "units" of topics. As will be seen later, the resource allocation mechanism is formulated in terms of units and can accommodate the mirroring and switching to "private" broker VMs. If the local domain was a set of interlinked LANs then the clusters would be the set of LANs. Again a node could correspond to a single broker or multiple brokers. A managing agent is associated with each broker and the constituent overlay manager (all of which are described later) ensures:

- that publishers know where to publish to and the subscribers know where subscribe to
- the current policy for topic mirroring is applied in the local domain when triggered

Mirroring workload and switching end users between primary and mirror brokers does not necessarily require extra idle back up brokers in the local domain. Of course, in principal, it could be effective to also transfer some of the publishing out of the local domain and directly to another domain, bypassing the gateway HA (high-availability) cluster. However, this would require special security arrangements between members of a federation as the domains could be in different ownership and the number of brokers and identities of the brokers in a domain may change dynamically. Hence in this architecture we are not developing this option.

In this architecture it is assumed that if a cluster is geographically separated then there is high bandwidth connectivity. Otherwise they are treated as separate local domains. Gateway brokers are often expected to be multi-homed.

3.4 Summary

This chapter first describes the hierarchical architecture of the federated MOM system that is considered in this thesis. The architecture has at least two levels. Each bottom level local domain is deployed in a well-connected network. Interconnected local domains form a top level federated overlay network of brokers. Then the research goals and novel works under this architecture are illustrated. The focus of this thesis at the local domain level is to quantify the risk of overloading processing capacity of MOM, and then pre-emptively allocating workloads or mirrors of the workloads among local domain brokers to minimize this risk of overloading MOM.

At the federated MOM, the focus of this work is to provide resilient overlay networking in the face of underlay substrate failures – in particular, geographically correlated failures, which can vastly impact the substrate. This thesis presents the design and evaluation of a novel overlay routing method between nodes. It provides multi-path selection with better chance to survive in geographical correlated failures, and hence allows the overlay of brokers to provide a resilient messaging over WAN.

Chapter 4 Local Domain Resilience

4.1 Introduction

Time-sensitive or mission critical applications, e.g., financial data delivery banking transactions, and remote control commands all place different requirements on publish subscribe message oriented middleware for reliability and performance. They are very sensitive to performance degradation and network and broker failures. This chapter looks particularly at the provision of resilience in local domains to adapt proactively and reactively against performance degradation. Specifically, within a well connected network of a local domain, the cause of performance degradation is found mainly to be the fluctuation of workloads that might overload the processing capacity of brokers. When the processing capacity of a broker is overloaded, the message matching delay and output queuing delay will rise above its normal standard. Motivated by these specific problems the design of the resilience mechanisms are as follows. Firstly the workloads are allocated to brokers to minimize the risk of overloading in the overall system according to the statistical patterns of local messaging loads. Secondly, because in a dynamically changing system, the messaging loads can exhibit unpredictable peaks and brokers may also underperform because of internal faults then an effective off-loading policy needs to be determined. At instances of excessive load, the load balancing mechanisms described are designed to detect system underperformance or unbalanced loads among brokers, and then offload load subscriptions to maintain a healthy messaging performance.

This work therefore aims to maintain a stable and responsive messaging system by combining risk-aware workload allocation and mirroring and load balancing. The risk-aware workload allocation and mirroring aims to maintain stable system wide health, using load statistics over a long time scale e.g. over one or a few days, which purely reactive load balancing cannot

optimize. The load allocation proposed here proactively minimizes the probability of overloading brokers in peak periods, and operates in the off-peak periods. On the other hand, if workload characteristics change unexpectedly and overload a broker or a faulty broker is short of resource to process messages, load balancing reactively mitigates the degrading broker in tens of seconds.

This work was partly done as part of the FP7 GEMOM project (Genetic Message Oriented Secure Middleware) [103, 104]. GEMOM was driven by the requirements to produce resilient and secure MOM solutions appropriate for future real-time and business critical systems. Other partners on GEMOM also conducted researches on adaptive security and trust mechanism [105-107], and implemented basic broker features. These will not be discussed in this thesis. The work in Chapter 4 and 5 are based on the author's contribution.

The rest of the chapter is structured as following. Section 2 presents related state of art MOM approaches. Then, in Section 3 the architecture and resilience mechanisms of our MOM system is explained. Section 4 describes the system model and performance metrics that are used in this work. Section 5 presents the risk-aware workload allocation and mirroring mechanisms. Section 6 presents the load balancing mechanisms. Section 7 explains the mediation of operations during load allocation and balancing sessions. Section 8 concludes the chapter.

4.2 State of the Art

This section describes the some functional features from most state-of-the-art MOM systems. The local domain system of RQMOM is composed of these basic features with additional extensions. Most of state-of-the-art MOM implementations, e.g. Apache's AMQP Qpid and ActiveMQ, offer high-availability clustering (HA clustering) and federation functionalities to

enhance reliability, interoperability and scalability of the messaging service. The two mechanisms can be combined and adapted to build MOM deployments with different topologies and to suit scenarios where applications have different performance and reliability requirements. The current HA clustering techniques and federation techniques are orthogonal concepts. Broker clustering techniques consist in creating groups of brokers that work closely together. HA clusters improve the reliability by replicating entire states and messages of the working broker to another broker. They support the clients to failover to another broker in the cluster if the working broker fails. Federation enables the communication between brokers. Federation thus both supports connecting brokers in different domains as a MOM overlay, and improves system scalability by distributing computation and bandwidth contention of the message brokerage to multiple brokers such as [108].

Some problems associated with the current HA clustering and federation motivates our research. Firstly, the bursty surge in demand of workloads will cause significant performance degradation [109] and the surge of correlated workloads will have a super additive influence on such degradation. A workload allocation mechanism that minimizes such problem is missing. In our approach instead of only considering the mean value of workloads, our system employs a novel workload allocation and mirroring algorithm which quantifies the probability of workloads exceeding the capacity of brokers. The allocation algorithm accommodates the correlation between messaging workloads as expressed by the variances and covariances of the topic message rates. Secondly, introducing redundancy with HA clustering requires replicating the entire workload of a primary broker to another broker, which requires much computational resource that is not needed most of the time and the consequent replication intra LAN does not fit well with MOMs deployed for internet messaging. With our algorithm, the partitions of workloads (workload is partitioned into sets of topics called items) on a primary broker are mirrored to different neighbour brokers in the local domain, instead of

replicating the entire workload to another broker. This allocation mechanism is illustrated in section 4.5. Thirdly, the federation of MOMs across internet requires low convergence time from random underlay IP network failures and faults. Our system employs overlay level multi-path routing for resiliency in networking between local domains. This is described in Chapter 5.

Resiliency in Pub/Sub based MOM is a popular research field, with some related works focusing on specific contexts. Yoon [110] designed a set of protocols to replace a faulty broker with an extra spare broker. The replacement is through recovering the connections to neighbour brokers by contacting an external directory service and recovering subscription tables from the reconnected neighbour brokers. Their protocol is for on-demand replication which means there is no live redundancy for continued messaging. The entire workload of the faulty broker is replicated together similar as the failover in HA clustering. Our approach provides live mirroring for continued messaging in the face of single broker failure with workload mirroring and reconfiguration solution for multiple broker failures. Workload mirroring is explained in section 4.6.

4.3 System Definition

4.3.1 System Components

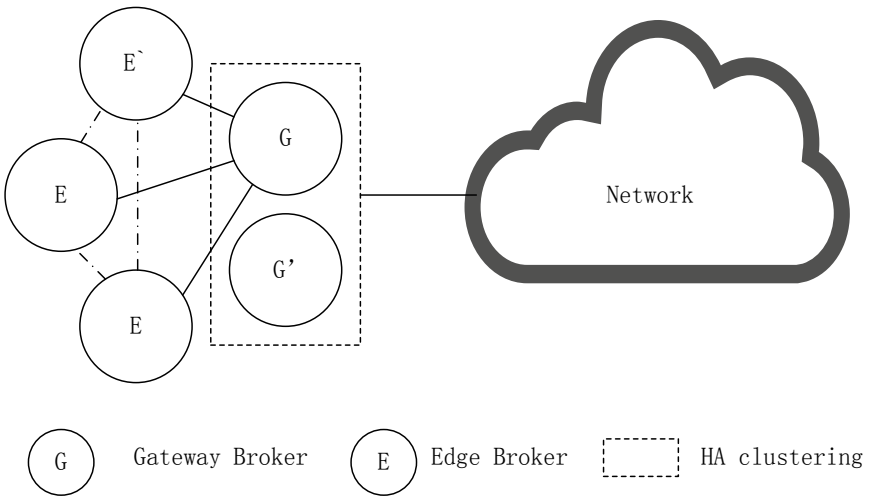

Figure 4-1 A simple example of local domain MOM system architecture

The local domain architecture is shown in the above figure**Error! Reference source not found.**. A local domain is made of a gateway broker and several edge brokers. The gateway broker is configured to dedicatedly process the inter domain messaging in the federated overlay. Because the importance of the gateway broker, it is often enhanced by two mechanisms: 1st HA clustering, which means a cluster of dedicated brokers is exposed as one virtual gateway broker and 2nd passive replication i.e. the failed gateway broker is replaced by one of the edge brokers.

At the local domain level a gateway broker is only connected with local edge brokers, and local publishers. Usually no local subscribers directly connect to a gateway broker unless it is as such by administrators and the subscriptions do not require any processing of message content. The publishers of topics that are subscribed to only by local subscribers are directly connected to edge brokers. Publishers of topics that are subscribed by only remote domain subscribers are directly connected to the gateway broker. On the other hand, the publishers of topics both locally and remotely subscribed to are able to connect to either gateway or edge brokers; however, if minimizing inter-domain latency is demanded by the system, they are preferably connected to gateway brokers. At the federated overlay level each gateway brokers only communicates with a small number of neighboring gateway brokers, hence avoiding the maintenance of too many pair-wise connections. The latter would be prohibitively expensive as the system scales up in a federated overlay. All the local subscribers are assigned to different edge brokers, so that the message processing and forwarding work in local domain can be shared. The load information and subscription information are exchanged among edge brokers and gateway broker, and this information may be exchanged using the inherent Pub/Sub infrastructure.

In the RQMOM framework there is a Management Agent (MA) collocated with each broker (local and gateway). MA conceptually consists in components including the Resilience Manager (RM), Overlay Manager (OM), Mediator, Load Detector (LD), and offload algorithms, as shown in Figure 4-2

Components of a managing agent. These components are necessary for a fully functional system, but only the aspects needed to validate the mechanisms proposed are in fact developed. Some of the components are based on those developed in GEMOM. Among these components, workload allocation functionality are realized by RM, OM and mediator; and load balancing is realized by LD, mediator and offload algorithms. These components are as shown in the following figure.

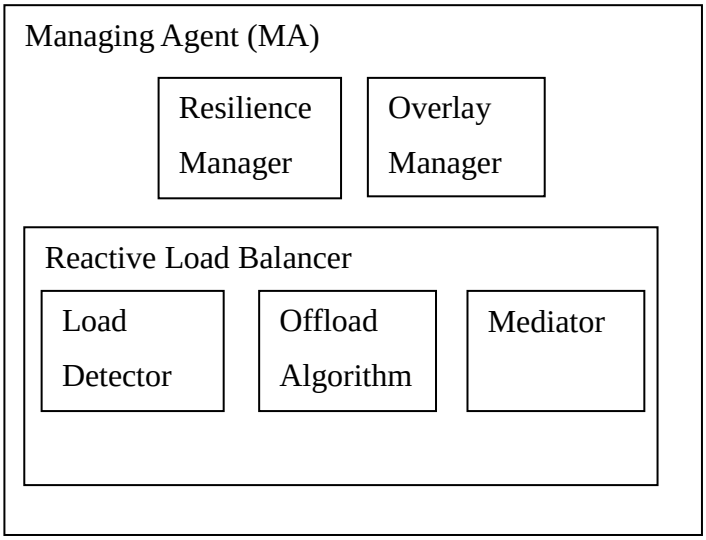

Figure 4-2 Components of a managing agent

One of the MAs in the local domain is elected as Master MA, which is used to compute workload allocation policies and distribute the policies and updates to other slave MAs. A workload allocation policy defines the set of subscriptions each broker carries, i.e. the topics that are published to it. When a Master MA fails, one of the other MAs is elected as the new Master MA. The up-to-date policies in all MAs and the master election mechanisms provide resilience to master MA faults. Currently the host with most computational power is the natural choice to host the Master MA, although different election algorithms can be applied. Inside the Master MA there is a RM and the RM and operates as follows. The RM computes the workload allocation policies and redundant workload replication policies which are called ***mirroring*** policies. The mirror policy preemptively informs which alternative broker a

subscriber should connect to, in case the currently attached broker has failed. These policies are computed for different possible states of the MOM system, e.g., initial allocation, single broker failure, double failure and hence over time the RM also can build a case base (as part of the master MA) with these policies so that faster response is possible, by state matching. Note that in low load situations the policies associated with switching off certain brokers are essentially the same as the failure case. If the load is low, for lower energy and computing resource consumption certain brokers may be disabled and their topics allocated to the other brokers in the local area. These calculations are based on the provided load (message volumes, topics and subscriptions) and are not triggered necessarily by failure.

New policies are disseminated to the other RMs in the local domain so that each stores an up to date case based. The RM communicates with the local OM and sends updates of the case base of the new policy and its context. The case base in each broker is only committed when all OMs acknowledge receipt of the update. The OM is the management interface of to the broker and clients i.e. publishers and subscribers. The OM listens to the system state from monitoring tools, using subscriptions to system management defined topics. The master OM decides when to apply the policies. When a critical event, e.g., a broker failure, is under the radar, an optimal policy under the new current state of system is retrieved from the case base and is used by OM. The Master MA has distributed changes in policy case base to all slave MAs to keep them up to date, each time it computes a policy for a new situation.

A risk-aware workload allocation algorithm and hard constraints are used to compute the policies. This is explained in detail in 4.5.2. Compared with high-availability clustering, the policies are computed at a granularity of predefined sets of topics called items and applied on “item” level. The use of item rather than topic is simply to reduce computation and the size of case base and to allow extension e.g. to cloud management. An item is a system administrator defined partition of total workload carried by the MOM. An

item can be considered as set of topics, treated by the RM as a single entity with respect to re-allocation of resources. A topic in a Pub/Sub MOM is simply a label that a publisher (or publishers) can use to identify a message stream of particular type and subscribers can subscribe to the topics they want. Another difference in this finer grained approach is that a redundant replication (i.e. a mirrored item) can be mirrored to live brokers instead of just to an idle slave broker that replicates the entire workload and states of a live broker. The workload allocation policy will allow a smaller set of brokers to carry the traffic while still ensuring overload probability is lower than a prescribed risk bound. This algorithm is described in section 4.5.
In the load allocation algorithm it does not matter whether redundant replication of messages is being used or topic mirroring (where the publishing and subscription configuration is set up to anticipate switch over) as in each capacity needs to be allocated. The choice will depend on bandwidth capability, charges, and switch over without message loss functionality available in the brokerage system used.

Besides the RM and the OM, each MA associated with a broker in the system runs an instance of the reactive load balancer (RLB). In general, the RLB detects overload and load imbalance at the specific broker, and it will also have its own heuristics implemented in its offload algorithm, upon which to offload topics if there is no policy in the case based that is applicable. (For example, the policies in the case base may be all generated under the hypothesis of single broker failure.) The heuristic will transfer specific subscriptions to a broker with more available resources. The RLB consists of the Load Detector, Offload Algorithm and Mediator.

To explain the control flow of a typical load balancing session a load balancing process is described. Note that RLB is not the focus of the thesis; however the basic steps of the load balancing process, which has cost overheads, is described for completeness. First the Load Detector detects if its

own broker is overloaded by monitoring a number of performance metrics. If overload or load imbalance is detected, the detector tells the Mediator to establish a load balancing session between its own broker, called here the *offloading broker* (the broker with the higher load doing the offloading) and a *load-accepting broker* (the broker accepting load from the offloading broker). Secondly once the load balancing session is established, the offloading broker collects publication delivery statistics for subscriptions that it is carrying and passes this information to an offload algorithm. Finally the offload algorithm strategically selects the set of subscriptions to offload and passes this set to the Mediator to coordinate the subscriber migration process. The load balancing session is over once the subscriber migration is complete. Such dynamic load balancing is not the focus of this work. The purpose of this description is to put them in the context of the architecture proposed in this thesis. Details of different approaches can be found in [42]. As described above a load balancing session processes imposes a considerable overhead in computing power as well as signaling and information exchange. The following section will introduce a novel workload allocation mechanism to reduce the probability of overloading local domain brokers, and thus reduce the possible overhead of load balancing.

4.4 *Broker Performance Metrics and Models*

The following sections first introduce some findings from previous studies on the performance characteristics of MOM. Building on these, this section then describes the metrics that have been used here to monitor and measure the MOM performance and then explains the prediction performance model of MOM that is used and the workload allocation mechanism that has been developed.

4.4.1 Performance of MOM

To define our performance metrics, we assume a general broker architecture consisting of three processing components: an input queue to buffer incoming messages to the broker matching engine; a message matching engine (a.k.a. message processing engine) that takes a message from the input queue, performs the matching (i.e., publications against subscriptions, apply content or header filters), generates and puts zero or more messages to route into the output queue; and at least one output queue to buffer messages from the matching engine to get transmitted to the subscriber or another relay broker. Please note here a topic based Pub/Sub is assumed, in which a subscription is identified by the associated topic and it is possible for a subscriber also to define filters (aka selectors) in a subscription. The figure below shows an example of such broker architecture.

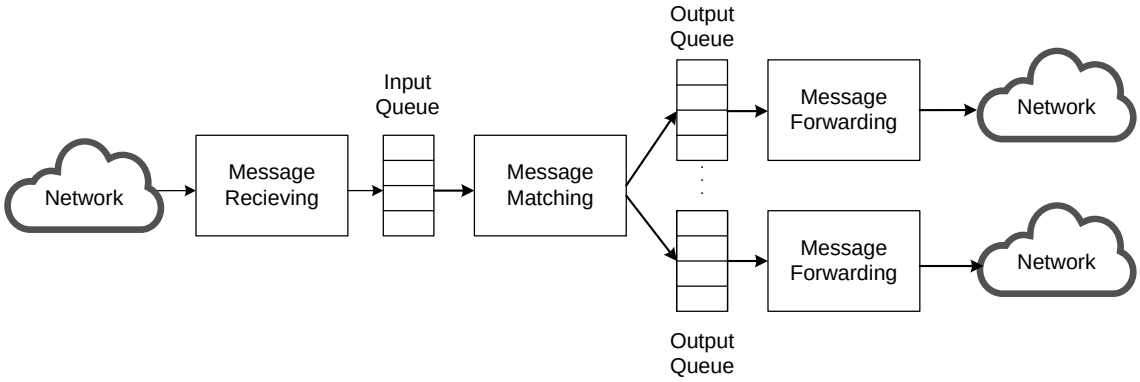

Figure 4-3 Broker model assumed

Each of the three processing steps will consume certain computing resources, and the processing capacity of a broker is determined by these resource consumptions.

Previous works have studied the impact that different system and workload parameters have on the performance and scalability of MOM [111, 112]. As previously described, any change of parameters that increase overheads in any of the three processing steps above will increase processing delay and

decrease maximum throughput. As the test in [112] shows, the maximum throughput decreases as the number of subscribers and filters increases on a broker. Additionally, [113] shows that performance and scalability can be increased by using a fan-out architecture to distribute the workload to additional brokers, and limit the number of subscribers within a prescribed threshold number. Note their test also demonstrates that when the workload overwhelms the capacity of a broker, the broker performance will drastically deteriorate, become unstable and possibly crash in the worst scenario.

Besides the computing capacity of broker host, the performance of broker is affected by the amount of workload, system parameters and other external factors affects host processing capacities such as software or hardware faults. To effectively maintain the optimal operation of the brokers, the state of brokers and hosts are closely monitored. The monitor process requires proper measurable metrics to be defined, so that the state of brokers can be represented and likely performance degradations or system faults are detected. When such events are detected, proper actions are taken to mitigate performance degradations or recover from system faults effectively and in time. Such responsive actions can include common load balancing and other self-* responses. Here reactive load balancing and proactive workload allocation are the focus of this work.

4.4.1.1 Performance metrics

Load balancing is commonly used action to mitigate overload of a broker or other events such as faults. This section describes some proper performance metrics that are used for monitor the overall MOM system state and help to perform load balancing. Since the focus of this local domain work is to combine proactive workload allocation and reactive load balancing to provide better resilience, even though the former one is the emphasized feature, load balancing still plays an important part. To monitor the performance of a

broker and its hosting operating system, two sets of metrics need to be chosen. The broker performance metrics include average message processing delay, input utilization ratio, message store size. In the hosting operating system, over consumption of resources by either broker processes or other applications will reduce the performance of a broker. Operating system resource utilization metrics include CPU, memory, network bandwidth utilisation ratios.

More detail regarding the broker performance metrics is now given:

The average processing delay d is defined as the average time spent by the matching engine between taking a message as input and producing zero or more messages as output. The average matching delay metric is important because it captures the average amount of processing time each message undergoes when processed by a broker. Time-critical applications often require having maximum allowable the processing delay as one of the QoS parameters. High processing delays indict that processing capability of a broker is overloaded by incoming workloads or is hindered by other software or hardware anomalies. During the operation of MOM, a threshold value of delay is chosen by the system operator or user. During the observation window, if the actual processing delay is above the configured threshold for a certain length of time, then load balancing will be triggered.

Maximum throughput m_r and average processing delay are mutually convertible. The maximum throughput is the maximum incoming message processing rate m_r, under which the system can operate without instability or performance degradation. m_r is calculated by taking the inverse of the average matching delay per message $m_r = 1/d$. When incoming message rate is reaching m_r, the computing resource utilization is approaching 100 percent. Furthermore as demonstrated in broker performance calibration work [111], if the incoming message rate continuously exceeds m_r, then system computing resources will be much over used. It subsequently leads to instability and

maybe a crash. Keeping this in account, MOM systems such as [114] prefer to keep the incoming message rate below $m_{r,}$ or by a margin below m_r. Hence, the **throughput utilization ratio** (I_r) is calculated from i_r representing the incoming publication rate in msg/s, and m_r representing the maximum matching rate, which is the inverse of the average matching delay per message, also in msg/s :

$$I_r = \frac{i_r}{m_r}$$

I_r can have any value greater than or equal to 0. A value of 1.0 or greater for I_r signifies that the broker processing resource is totally consumed or over consumed. A threshold value is defined for I_r, and when I_r is above the configured threshold for a certain length of time during an observation window, then load balancing will be triggered.

The length of message store in a broker can be defined by its configuration. It determines the volume of messages that can be hold in a broker's memory. Fast message production or slow message consumption can lead to message volume increase in the message store. When the message store over flows, the broker may take reactions such as dropping messages, write messages to persistent storage. This subsequently leads to performance degradation or loss of messages. To prevent this, before the messages hold up the total storage of message store, some workload should be offloaded to other available brokers, or more storage should be provided to current broker. Otherwise, traffic control policy needs to be implemented to throttle or drop incoming messages.

Besides the broker performance metrics, its hosting operating system also needs to be monitored with defined performance metrics. These metrics are more common among information systems. Operating system level metrics monitor the current computing and networking resources, such as CPU utilization, memory utilization, input and output bandwidth utilization, etc.

Alternatively, in place of the utilization ratio counterparts, it is also possible for the load balancer to use input queuing delay and output queuing delay as performance metrics. However, the queuing delay metrics has limitations as follows: firstly, queuing delays cannot show the utilization level of a resource but only indicate that the resource is overloaded when queuing delays rise rapidly. (The length of the queue is however a good indicator of danger of bottleneck.) Secondly, queuing delay measurements do not accurately indicate the load of a broker at the instant the metric is measured because it is obtained after the message gets dequeued. Therefore, the measurement is lagging by the delay measured.

4.4.2 Broker Performance Model

4.4.2.1 Linear Multiple-regression model

To understand the performance characteristics for generic JMS supported MOMs, Henjes, etc. [111] carried out a series of experiments which calibrate the performance of a broker under different workload and configurations. From the experiment results and analysis on MOM broker operation, he derives a few key factors that influence the performance of MOM broker, and builds an abstract analytical performance model basing on these key factors. This multiple linear regression performance model, to our knowledge, is one of the most well tested and comprehensive performance models available to date. Hence the workload allocation mechanism uses this performance model. This model describes the maximal throughput of a broker by combination of the configuration parameters of system and the timing parameters of different processing tasks. It is assumed that within an application scenario the configuration parameters of workloads can be easily obtained, such as average number of filters (selectors in JMS terminology) and subscribers per topic. Also some basic performance tests carried out to before the MOM brokers are deployed, to determine the other performance parameters. Then

this model is used to establish the maximal system throughput, which its allocated workload should not exceed, for a stable and near optimal system performance. This section describes key properties of this model.

As illustrated in Figure 4-4, the MOM broker's messaging process is abstracted into a several tasks, each of which will consume computing resources and contribute to the overall messaging delay. To process and deliver each message, a MOM broker will carry out at least three different tasks. First, the after receiving a message from a publisher, the broker puts this message onto an internal queue so that it is stored in memory and waits for further processing steps. In the second task, the broker matches the message with filters associated with its subscriptions. After a message is matched, the third task is to dispatch the message to all matching subscribers. The total time to process a message, is then the aggregation of processing time in these three tasks. Henjes [111] , after experiments, chooses a linear multi-regression model to approximate the broker performance dynamics and relate the performance to several different factors. This model is also used to predict MOM system performance under different workloads and configurations. From the analysis above, the a**verage message processing time** T is predicated as:

$$T = t_{rcv} + n_{fltr} \cdot t_{fltr} + r \cdot t_{tx}$$

t_{rcv} is the overhead to receive a message and put in the internal queue (The first task above.)

$n_{fltr} \cdot t_{fltr}$ is the total overhead for matching all the filters and corresponds to the second task above. n_{fltr} is the number filters applied, and t_{fltr} is the average processing time of a single filter.

$r \cdot t_{tx}$ is the total time to dispatch a filtered message to all its r subscribers. Here r is the average number of subscribers that a message is matched to, and it is also known as message replication grade. t_{tx} is the average time to dispatch a message to a single subscriber. The time to forward a message to all recipients is $r \cdot t_{tx}$, which corresponds to the time to forward r copies of the message.

Here, n_{fltr} and r can be viewed as the regression variables and T as the dependent variable in the regression model. In this model the processing times t_{rcv}, t_{fltr} and t_{tx} are the regression parameters that need to be calculated through model fitting.

Among these five parameters, n_{fltr} , and r are parameters of the workloads; t_{rcv} , t_{fltr} , and t_{tx} are parameters that are dependent on the MOM implementation and computing capacity on the hosting operating system and hardware. They can be estimated from the workload characteristics in an application scenario.

In the broker calibration test before it is assigned real workload, the parameters are estimated via least squares regression. Also in a certain application scenario, the message size, replication grade and number of filters are known. Then average message processing time T is predicted under a given workload and previously calculated parameters. We can further predict a MOM's capacity or throughput from T**.** The received throughput B can be calculated by inverse of average processing time.

$$B = \frac{1}{T}.$$

If overall throughput B_a is considered instead, which is the sum of input and output messages, then it can be calculated as $B_a = \frac{r+1}{T}$. Here the total number of output messages are r times the incoming messages, where r is the average replication grade.

The experiments in [111] showed B is primarily determined by the computing resource of the host, not the network resource. The experiments also showed that in all tests carried out, the broker reaches its maximal throughput before network bandwidth is fully consumed. It means that in a well connected LAN environment where MOMs are usually deployed, the broker performance becomes the system bottleneck, instead of the network bandwidth. Hence to

plan the capacity of a MOM system, the broker performance has to be considered. Furthermore, the maximal throughput of a broker under a given configuration is naturally the performance metric that needs to be considered. When a broker is overloaded beyond its maximal throughput, its performance will degrades and potentially lead to an unstable or faulty system state.

A workload allocation mechanism is built to address the capacity planning problem. Before the workload allocation mechanism is introduced, the method to estimate the parameters by model fitting is first described. Then the derived linear multi-regression model can be used to predict broker performance by the workload allocation mechanism.

4.4.2.2 Estimating the Parameters

As previously described, basing on experiments and analysis, several parameters that have significant impact on MOM performance been chosen. A linear multi-regression model [111] built basing on chosen parameters will be used. Now the model fitting is needed to find the value of parameters. In the following, least-square approximation is used to calculate the model specific parameters.

First, it is needed to briefly introduce the basis of linear multi-regression. In a multi-regression model, the value of dependent variable Y is represented as multiple functions of a set of n independent variables $\{x_i\}$. In the most simple case, when $n=1$ and the function of x is linear, dependent variable Y is represented as one linear regression model.

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \epsilon_i$$

However in the more complex systems, such as describing MOM performance, multiple factors often impact the output dependent variable. Hence multiple variables x_1 , x_2 , … x_n are in need. In a linear multi-regression model, the impacts from independent variables are modeled as linear influences.

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_n x_{i,n} + \epsilon_i$$

In the equation, $\{\beta_i\}$ are $n+1$ regression parameters and $\{x_{i,k}\}$ are n independent variables. Y_i is the i^{th} out of m observation value. The error ϵ_i is the error between prediction value $\widehat{y_i} = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_n x_{i,n}$ and observation value Y_i.

Given the a set of observation values Y and corresponding variable values $\{x_{i,k}\}$, the model is fitted by minimizing the least-square function, with respect to $\{\beta_i\}$. Here the least-square function is:

$$\mathrm{L} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \epsilon_i^2 = \sum_{\mathrm{i}=1}^{\mathrm{m}} \left(Y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{i,j} \right)^2$$

The above multiple can be easily transformed in matrix notion:

$$Y = X\beta + \epsilon$$

The least-square function that is minimized can be also transformed:

$$\mathrm{L} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \epsilon_i^2 = \epsilon^{'} \epsilon = (Y - X\beta)^{'} (Y - X\beta)$$

For a vector x with size n, its Euclidean norm is

$$\|\mathrm{x}\|_2 = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x}_\mathrm{i}^2}$$

Note in the performance model:

$$T = t_{rcv} + n_{fltr} \cdot t_{fltr} + r \cdot t_{tx}$$

$Y = (T_1, T_2, \ldots, T_m)'$ is observed total processing time. $X = (X_1, X_2, \ldots, X_m)'$ where $X_i = (1, n_{i,fltr}, r_\mathrm{i})'$ are the system parameters in the i^{th} observation. Given X, and Y, the regression parameters to be calculated $\beta = (t_{rcv}, t_{fltr}, t_{tx})'$ represent the processing delays, which are greater or equal to zero. Hence the minimization problem for fitting performance model is = can be transformed into the following optimization, with the added constraint to avoid negative results that represent a best fit.

$$\min_{\beta} \|X\beta - y\|_2 \qquad (\beta \geq 0)$$

The above minimization belongs to a subset of least-square problems, which have additional inequality constraints. Here the single constraint $\beta \geq 0$ is a non-negative constraint. This specific constraint problem is known-as non-negative least-square (NNLS) problem. It is a standard problem in linear algebra, with commercial and open source approximation implementations of approximation algorithms, which is out of the scope of this work. For example [111] used LSQNONNEG function of MATLAB suite.

After the model is fitted, and β is calculated by the approximation algorithm, given a new vector of variables X, a prediction of the Y value can be made by

$$\widehat{y_i} = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ik}$$

Specifically in the broker performance model, after the approximate values of processing delays in reviving, matching and dispatching messages are found, given a system parameters a broker, its total processing delay can be predicted. Hence the maximal throughput can be further derived for capacity planning.

4.5 Messaging workload allocation

4.5.1 Problem and Algorithm Description

Mission critical applications such as remote control commands and financial data are very sensitive to the service degradation and message loss due to link or broker failures. Service degradation is often caused by the workloads on a broker rising and exceeding the processing capacity of the broker [109, 115]. Reactive load balancing mechanisms provide mitigation only after an overload occurrence is measured and detected. Such post hoc load balancing between brokers is necessary, however it will not minimize the probability that overloading happens. To my knowledge few previous works have addressed how to allocate messaging workload within local domain, such that the system wide message processing latency is smaller and more stable,

and hence less post hoc load balancing effort is in need.

Correlated workloads often exist in real world systems. Such correlated workloads will likely rise together and thus will have a super additive effect on the total workload. For example financial trading increases at certain times of day when exchanges open and quiet after they close. The shares of companies in similar sectors are likely to be active simultaneously during trading. Uncorrelated or negatively correlated workloads, on the other hand, will make the total workload more stable. Hence to address the above problems on local MOM domain level, RQMOM proposes a novel risk aware workload allocation and mirroring algorithm in local domains.

The purpose of the algorithms can be further explained in concrete terms in two contexts. Firstly, correlated bursty increases in workloads may exceed broker capacity in a rush, and cause system performance insatiability and faults. Reactive load balancing can mitigate the overloaded brokers with some overheads, but it is only after an overload is detected. Secondly, capacity planning of MOM system is bounded by limited computing resources. IT systems need to make as efficient use of resources as necessary for profit. On the other hand reducing energy consumption for costs and environmental protection is another reason capacity planning is needed to reduce the operating computing resources. Hence when a MOM operator needs to limit or reduce the number of brokers in quiet periods, it also needs to keep the risk in known bounds.

The algorithms are able to compute workload allocation policies and redundant workload replication policies, i.e. *mirroring* policies, under different possible states of the MOM system, e.g., initial allocation, single broker failure, double failures, using this algorithm. In a local domain, the message volume of different topics, often have some correlated predictable patterns. Our algorithm quantifies the risk of overloading a broker,

processing speed degradation in brokers caused by saturation of high volume throughput. The notion of quantified risk exploits the variation and co-variation between the critical peak-period messaging volumes of different workloads, and avoids allocating highly correlated workloads together, which will bring bursty high volume messaging traffic that degrades a broker's processing capacity. Additionally RQMOM uses the allocation algorithm to find the solution of where to place mirrors of critical topics - prior to or on the occurrences of faults in original broker carrying the critical topics - without introducing too much risk of overloading the system.

In comparison to HA clustering, where the entire workloads of a primary broker is replicated to another live broker, the mirroring strategy divides and assigns the mirror workloads of a primary broker to several live neighbour brokers in local domains, and does not necessarily introduce many extra idle brokers. The atomic unit for the workload allocation and mirroring is an *item*. An item, according the specific scenario where the MOM system is deployed, is a pre-defined partition from the total traffic carried in MOM system. For example, in the case of topic-based Pub/Sub MOM, an item is defined as one or more topics that are bundled together in workload allocation. Every item is mapped to a primary and a mirror broker. A physical broker logically can be the primary broker for some and the mirror broker for some other items at the same time.

4.5.2 Model and Solve the Workload Allocation Problem

Instead of relying only on dynamic load balancing, a pro-active workload assignment mechanism is employed improve resilience. So that, the probability of messaging workload exceeding processing capacity is reduced and less dynamic load balancing is needed. The workload assignment mechanism optimises the workload assigned among brokers, according to a performance model [111] previously researched.

The goal of the algorithm is to minimize the quantified risk of overloading the system by computing workload allocation and mirroring policies.

Additionally the allocation may be subject to important hard constraints prescribed by administrators. Such hard constraints include, for example, a few pre-allocated items to specified brokers, the probability threshold that the workload exceeds each broker's capacity should be lower than a prescribed upper bound, or that the maximum number of topics (or topic volume, defined as the message rate *X* message size summed over all topics) that a broker is allowed to carry is not exceeded.

This problem is formulated as allocating a set of items (an item is a set of topics) *J* to each broker B_i, as shown in Figure 4-4.

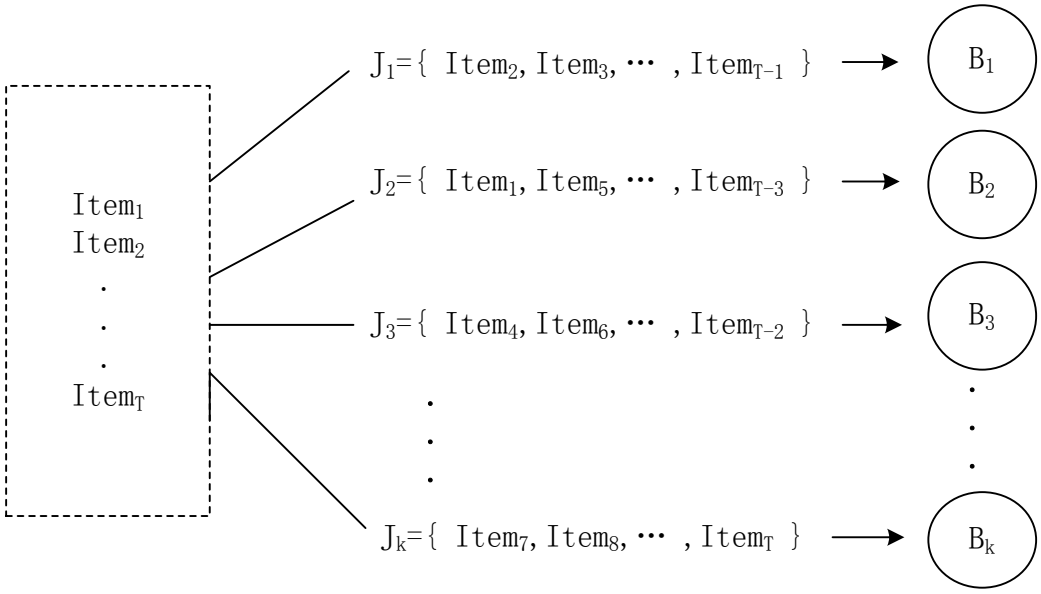

Figure 4-4 The allocation problem illustrated

Consider the optimal primary allocation case, i.e., the case where we simply want to maximize a utility function that will be defined in terms of risk, while allocating items to the brokers that are candidates that satisfy the pre-defined constraints. If J_i represents the set of items allocated to broker B_i, then the goal is to find the optimal solution $s = \{J_1, \ldots J_k\}$, which is *k* sets of items allocated to the corresponding *k* brokers, maximizing overall system utility (i.e., minimizing the overall risk) in the system. The peak period message rate of item set *j* is a random variable denoted by $\boldsymbol{V^j}$ measured in messages per unit time.

This problem resembles the classical "Generic Assignment Problem" for which various algorithms have been proposed [116, 117] . However a more complex structure of utility function makes this combinatorial problem more difficult. The "Generic Assignment Problem" is different, because it assumes that the costs or utilities associated with items are simply constants for each possible assignment. On the other hand here, the utility has a more complex structure. The utility is a probability that depends on both the assignment of workloads and the processing capacity of a broker. The processing capacity of a broker is further modelled as a linear function of regression parameters and variables that are determined by the broker and workload characteristics.

Modelling system utility requires identifying the key requirement metric emphasized by a MOM system such as the rewards over carrying message service on the risk of overloading MOM system, etc. Two possible utility models are described here. Both models include the risk estimation. The first one assumes an information brokerage model, and service reward is defined by the timely delivery of each subscription. The utility balances the system wide workload for rewards under the risk of possible overloads. The second is a generalized model, which maximize the probability that every broker works under a prescribed resource utilisation ratio, instead of being confined to the specific information brokerage use case. After the utility functions are described, a branch and bound search algorithm is introduced to illustrate the complete workload allocation mechanism. Then after this section, a more fine-grained GA approximation algorithm is introduced and evaluated. The GA is based on the second utility function and the linear multiple-regression performance model.

The allocation mechanism is first introduced using a simple case assuming the average message processing capacity m_r is a broker dependant constant value instead of a comprehensive performance model that depends on load and configuration. In this simple illustration the capacity m_r^i of broker i is a constant C_i (measured in messages per unit time). This is a reasonable assumption for a broker with abundant bandwidth in local domain, and each

topic is subscribed by a moderate number of subscribers with a small number of filters. This mechanism is further refined with an approximation algorithm for the finer grained linear multi-regression model. The refined model will be described in section 4.5.5. Given items to be allocated and the broker capacity values, the utility of current local domain U_{domain} can be modelled to represent both the reward for the carried message service for these messages within the broker capacity and the penalty for not covering carried message service exceeding broker capacity. The problem to maximize system utility, by finding the best combination of allocations, is formulated:

$$\arg\max_{\{j_1 \cdots j_k\}} (U_{domain} = \sum_{i=1}^{k} U_i)$$

$$where\ U_i = R_i^j - \int_{C_i}^{\infty} P_i^j (x - C_i) P_r(V^j = x) dx$$

The utility U_i of broker i carrying item set j is calculated from R_i^j which is the reward for carrying item set j and the penalty $P_i^j(x - C_i)$ for the arriving messages x exceeding the capacity of broker i. R_i^j is a value associated with item set j. P_i^j ideally could have different forms, e.g. either a constant or proportional to $(x - C_i)$. V^j is the volume (bytes per second) associated with item set j.

By assuming V^j follows a normal distribution and P_i^j is a constant value associated with item set j, we simplify (1) as:

$$\arg\max_{\{j_1 \cdots j_k\}} (U_{domain} = \sum_{i=1}^{k} U_i)$$

$$where\ U_i = R_i^j - P_i^j P_r(V^j > C_i)$$

We can approximate by $P_r(V^j > C_i) = P_r(\frac{V^j - \mu_j}{\sigma_j} > \frac{C_i - \mu_j}{\sigma_j})$. $\frac{V^j - \mu_j}{\sigma_j}$ follows the standard normal distribution, and the parameters are estimated by analyzing peak period message rates samples of all topics.

In a general workload allocation problem an *item* is a set of workload *units*. In the specific problem of MOM system workloads can be as divided into specific *units* such as topics. Hence in this work, an item is defined as a set of topics. Then the correlation and covariance among items are determined by the topics contained within these items. The definition of an item provides users with the flexibility to define the item according to specific environment. Since the workload units such as topics are nested into items, the correlation among items can be easily calculated with the known correlation matrix of the workload units.

The message rate sample is a series of messages arriving in a unit of time for each topic in the system, for example over peak periods. (However, the approach could equally be used to reduce energy costs, by reducing the number of brokers in quiet periods, keeping the risk in known bounds) Suppose the mean message rate is μ_t for each topic t. Let $V^j = \sum_{t \in j} V_t$, $\mu_j = \sum_{t \in j} \mu_t$. V_t is the volume in bytes per second for topic t.

From the sampled time series of message data overall T topics in a system, we can calculate a variance covariance matrix of $T \times T$. The variance of V^j is ${\sigma_j}^2 = \sum_{a \in j} \sum_{a' \in j} cov(V^a, V^{a'})$ where a and a′ are any item in j; and $cov\left(V^a, V^{a'}\right) = cov(\sum_{t \in a} V^t , \sum_{t' \in a'} V^{t'})$, where t and t′ are any topic in item a and a′ respectively. Intuitively covariance measures the degree to which two variables change or vary together (i.e. co-vary). The positively correlated items′ message rates vary together in the same direction relative to their expected values, hence they result in a relative larger σ, which leads to a larger $P_r\left(V^j > C_i\right)$ and smaller U_i. Hence by maximizing $\sum_{i=1}^{k} U_i$ the allocation solution will result in a small system risk of being overloaded and positively correlated items are less likely to be allocated to the same broker.

For a generalized system, the key requirement is to provide a resilient messaging service, which should minimize the risk of overloading brokers or

in other words maximizing the probability that all brokers maintain a healthy operation. As analysed before among different performance metrics the bottleneck in local domain MOM broker is mostly its processing capability, which can be captured by maximum throughput *I*. Other than a simple constant value in the demonstration of the previous utility function, the performance metric of maximum throughput can be predicted by the linear multi-regression model in section 4.3. Now the allocation algorithm is extended to a finer grained version.

$$\arg \max_{\{j_1 \dots j_i\}} \prod_{\mathrm{i}=1 \dots \mathrm{k}} p_r\left(I^i < I^{th}\right)$$

$$s.t.\left(d^i < d^{th}\right)$$

$$s.t.\ p_r\left(I^i > I^{th}\right) < p_r^{th}$$

$$s.t.\ r^i < r^{th}$$

Here we compute workload allocation or mirroring policies to minimize the total quantified risk of overloading the system. Some of above performance metrics have been introduced before in section 4.4.1. The optimisation also requires the allocation subjects so that they satisfying prescribed constraints such as: expected average matching delay per message (d^i) is smaller than a threshold value (d^{th}); the probability that the broker i's current input utilization ratio (I^i) under current workload exceeds input utilization ratio threshold (I^{th}) is lower than a prescribed threshold (p_r^{th}). The average replication grade r^i on a broker *i*, i.e. the average number of subscribers that need to be posted once a message is published, is lower than a threshold r^{th}. Detailed definitions of the above metrics can be found in section 4.4.1.

4.5.3 Search Algorithm

A branch-and-cut algorithm revised from combinatorial auction is implemented to compute the allocation solution. The search algorithm first generates the possible combinations of items. Note if there are items that are

pre-allocated by administrators, they can be excluded from the allocation, but they are used to calculate the utility and risk upper bound of allocation. The allocation algorithm use a depth first search to traverse and search for the best exhaustive partition of items allocated to brokers.

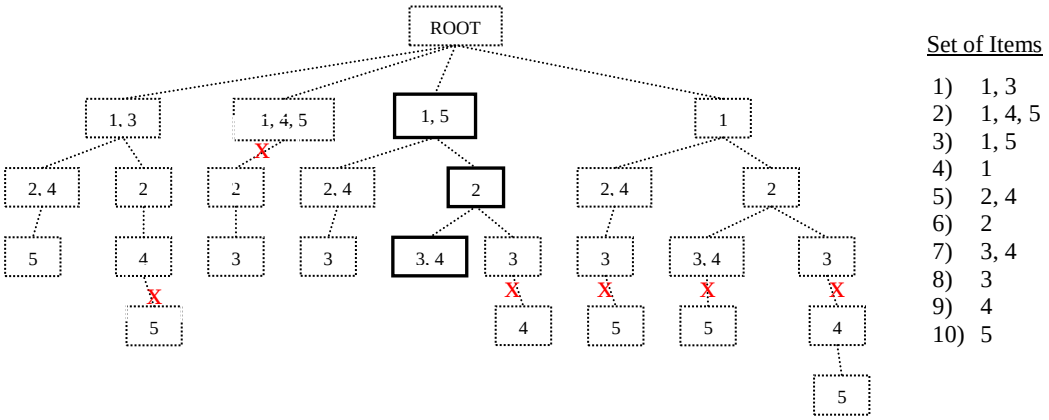

Figure 4-5 Part of a depth first search tree allocating 10 possible combination of 5 items to 3 brokers

A branch and cut approach is used to prune the search, i.e., the children of nodes that do not satisfy prescribed hard constraints such as risk upper bound of input message rates or processing delay are pruned. In the above figure, the red crosses represent the branches that have been pruned. And the highlighted branch is the solution found. The solution of search should be an exhaustive partition of items, which means each item is included exactly once in the allocation to brokers. Hence the items in a parent node are excluded from generating its child nodes, and the paths that have incomplete allocation of items are discarded. The figure above shows part of a depth first tree constructed to search for the allocation of 5 items to 3 brokers. Under the imaginary ROOT node, each level of tree represents the allocation to a corresponding broker. Due to the page constraint, only a part of tree is shown and 10 sets of combination of items are generated. The crossed paths are pruned because of hard constraints on risk upper bound and incomplete allocation below 3 levels of search.

4.5.4 Illustration

The message rate sample is a series of messages arriving in a unit of time for each topic in the system, for example over peak periods. The message rate for a topic t can be represented by a random variable V^t. The purpose of workload allocation is to effectively and efficiently manage the risk of overloading a broker given limited computing resource in a MOM cluster.

Suppose the mean message rate is μ_t for each topic t. Let $V^j = \sum_{t \in j} V_t$, $\mu_j = \sum_{t \in j} \mu_t$. From the sampled time series of message data over all the T topics, we calculate a variance covariance matrix of $T \times T$. The variance of V^j is $\sigma_j{}^2 = \sum_{a \in j} \sum_{a' \in j} cov(V^a, V^{a'})$ where a and a′ are any item in j; and $cov\left(V^a, V^{a'}\right) = cov(\sum_{t \in a} V^t, \sum_{t' \in a'} V^{t'})$, where t and t′ are any topic in a and a′. Intuitively covariance measures the degree to which two variables change or vary together (i.e. co-vary). The positively correlated items' message rates vary together in the same direction relative to their expected values, hence they result in a relative larger σ, which leads to a larger $P_r(V^j > C_i)$ and $smaller$ U_i. Hence by maximizing $\sum_{i=1}^{k} U_i$ the allocation solution will result in a small system risk reward loss because being overloaded. And positively correlated items are less likely to be allocated to the same broker.

To illustrate the allocation algorithm, suppose there are 32 topics overall in a local domain, which are divided into 6 items, to be allocated to 3 brokers with different capacities. From sample series data the statistical parameter such as mean, covariance matrix and correlation coefficient matrix calculated are shown in the tables below. The mean of message rates of six items are shown in table 4-1.

Item1	Item2	Item3	Item4	Item5	Item6
150	300	350	200	100	130

Table 4-1 Mean message rates (msg/sec) over sampled periods

The variance covariance matrix of items is shown in table 4-2 below. Both positively and negatively correlated items are present in the mix. Comparing with correlation coefficient, variance covariance matrix demonstrates both the degree and magnitude of correlations between items.

	Item1	Item2	Item3	Item4	Item5	Item6
Item1	5310	2456	962	1071	840	409
Item2	2456	7544	611	418	1237	542
Item3	962	611	6972	-1622	3401	2387
Item4	1071	418	-1622	7538	-102	4266
Item5	840	1237	3401	-102	4992	3626
Item6	409	542	2387	4266	3626	4202

Table 4-2 Variance covariance matrix of the 6 items

The corresponding correlation coefficient matrix between items is calculated as shown in table 4-3. Note that correlation coefficient is a normalised value of covariance. The relationship between correlation coefficient and covariance is: $\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$, where ρ_{XY} is the correlation coefficient between item X, Y; it is the covariance between X and Y, σ_{XY}, normalised by the standard deviations of each item σ_X and σ_Y.

	Item1	Item2	Item3	Item4	Item5	Item6
Item1	1.0000	0.3880	0.1581	0.1693	0.1632	0.0866
Item2	0.3880	1.0000	0.0842	0.0554	0.2016	0.0963
Item3	0.1581	0.0842	1.0000	-0.2237	0.5765	0.4410
Item4	0.1693	0.0554	-0.2237	1.0000	-0.0166	0.7580
Item5	0.1632	0.2016	0.5765	-0.0166	1.0000	0.7917
Item6	0.0866	0.0963	0.4410	0.7580	0.7917	1.0000

Table 4-3 Correlation coefficient matrix of the 6 items

We find the solution for the following utility with a Depth First Search using $R_i^j = P^j = \mu_j$as the reward and penalty associated with j. We apply a

prescribed risk *threshold*=0.007 as a hard constraint to the search. The risk *threshold* is a hard constraint can be optionally chosen by system administrators to represent a system requirement, in addition to the objection function.

$$\arg\max_{\{j_1 \ldots j_i\}} \prod_{i=1 \ldots k} p_r\left(I^i < I^{th}\right)$$

$$s.t. \quad P_r\left(V^j > C_i\right) < threshold$$

We compare the allocation result with a Round Robin allocation which allocates each item in turn to a broker with maximum resource reserve ratio.

$$\boldsymbol{reserve\ ratio} = \frac{\boldsymbol{C_i - V^{j\prime}}}{\boldsymbol{C_i}}$$

The two solutions are shown in the following **Error! Reference source not found.**. Solution 1 is returned by Round Robin, and solution 2 is by risk-aware allocation algorithm.

	Broker 1	Broker 2	Broker 3
C_i	700	750	600
Solution 1	Item 2,6	Item 1,3	Item 4,5
Solution 2	Item 1,2	Item 3,5	Item 4,6
$P_{r_1}(V^{j\prime} > C_i)$	0.0068	0.0054	0.0170
$P_{r_2}(V^j > C_i)$	0.0018	0.0016	0.0010

Table 4-4 The broker processing capacity (msg/sec) and allocated solution of each broker

The risk values of the two solutions are evaluated by computing a normalized gain. Risk in round robin solution and our solution is $\mathbf{Pr_1}$ ***and*** $\mathbf{Pr_2}$. A postive gain indicates $\mathbf{Pr_1} > \mathbf{Pr_2}$. The gain is shown in figure 4-6. Our algorithm shows capability to reduce the risk of overloading system. It is achieved by exploring the correlation between items, while avoids allocating highly correlated workloads to the same broker.

$$Gain = \frac{(\Pr_1(V_i^{j\prime} > C_i) - \Pr_2(V_i^{j} > C_i))}{\Pr_1(V_i^{j\prime} > C_i)}$$

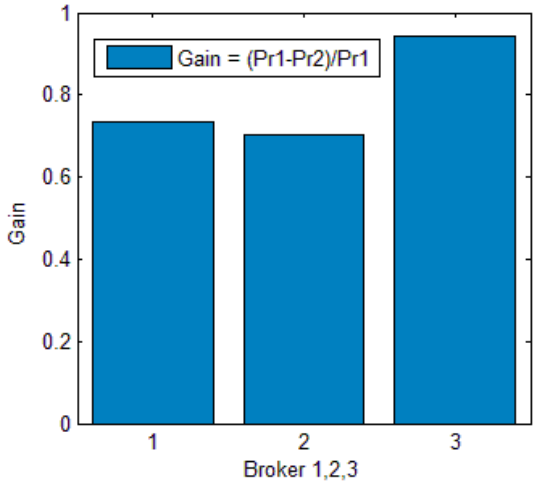

Figure 4-6 Normalized gain of our solution over round robin at each broker

In this section the workload allocation problem is illustrated, while the mirror problem will be discussed in the following section 4.6. In this illustration, in term of the estimated risk our solution shows obvious advantage over round robin solution. It is because our allocation algorithm avoids allocating some highly correlated workloads to the same broker by exploring the correlation via the variance computed from the variance covariance matrix.

4.5.5 Approximation with Genetic Algorithm

4.5.5.1 Introduction

The exact best solution(s) for reliability and performance optimization is not always desirable. It is because exact solutions are often difficult to obtain and their utility as an abstraction of resiliency are often only marginally better than a good approximation. Also, note that the solution's optimality is in terms of the model built representing underlying system, and given a good model its best solution is still not likely perfect in the underlying system. The workload allocation problem is a modified generalised assignment problem, which is a NP (Non-deterministic Polynomial) complete problem. Since it is

not easy to efficiently find the optimal solution, a heuristic or meta-heuristic algorithm is needed to approximate and find a viable near-optimal solution. The purpose of this is to find an allocation solution efficiently that is better than the allocation methods currently widely employed such as weighted round robin.

The Genetic Algorithm has proven to be effective on difficult optimization problems for which it is difficult to efficiently and systematically improve solutions using heuristic or deterministic iterative methods. In this section the evaluation of the approximation algorithm is conducted in comparison with round robin which is usually used to allocate workload in practice. The multiple linear regression performance model is used in the work.

4.5.5.2 Background to Genetic Algorithms

Genetic algorithms are a kind of search algorithm modelled on the natural selection process. In a GA, each solution is encoded as a vector of parameters and corresponds to a point in a search space. A vector of parameters is called a chromosome. Depending on the encoding of decision variables the parameters may use different representations, e.g. binary alphabet, string, integer, or real valued. Here in the workload allocation problem the vector of parameters is decision variables encoded as integers and the vector is a representation of the configuration of the MOM system which includes the publishers, subscribers and brokers. The solution vector in a GA is similar to the biological concept of a chromosome. Biologically a chromosome is the template that organisms are developed from. The fitness of an organism to the environment determines its abilities to reproduce and hence the fitness impacts the probability that its chromosome evolves by genetic operations such as crossover and mutation. A chromosome is composed of genes. Each gene is located in a particular locus and encodes a trait of the organism. The possible values of genes are called alleles. The chromosome values (genotypes)

are decoded into phenotypic values that represent an organism's characteristics, such as eye colour, brain size, and skin colour. Phenotypes correspond to decision variables that are represented and evolved as chromosomes in GA. In both the biological world and in GAs the genetic operators operate on chromosomes, which are the encoding of decision variables, rather than the decision variables themselves.

To evaluate the performance of each individual solution, fitness values are calculated by applying a fitness function to chromosomes – the fitness function plays the role of the natural environment in biological systems. The phenotype of a GA is the real system variable evaluated, and here the fitness value represents the system performance under the configuration applied to the set of brokers, publishers and subscribers that operate the messaging system. The phenotypes with higher fitness have better changes to reproduce their offspring. Three types of operator are used to create offspring, viz. selection, crossover and mutation.

Selection: During each generation, the fitness function is used to select a portion of the population for reproduction.

Crossover: Parts of the parent chromosomes are used to generate a new individual for the next generation.

Mutation: A small fraction of a chromosome is modified randomly. This helps the search cover new areas of the search space.

These genetic operators emulate the natural evolution processes of organisms, and breed chromosomes with good fitness values and boost a population's diversity in the search space. The fitness function applied before each selection step, is derived from the objective function, which characterizes an individual chromosome's performance in real domain.

The basic GA is shown in the following figure. There are many variants in each step in the process.

1: Randomly generate an initial population of solutions
2: Evaluate each individual solution using the fitness function
3: **repeat**
    4: Select parents to create offspring using crossover
    5: Mutate the offspring.
    6: replace the current population with the new population.
    7: calculate the fitness of each solution in the population
8: **until** a stop condition is satisfied.

Figure 4-7 The top level structure of a basic GA

In GAs, a set of generations is called a run and at the end of a run the more highly fit chromosomes in the population are treated as an optimal solution. Because of the numerous random selections in the process, two runs with different random-number seeds can produce different behaviour. Hence statistics averaged over different runs of the GA are needed. GA is particularly useful on problems with little domain knowledge because they can succeed even when the solution space is hard to characterise.

A chromosome is defined as a vector of encoded decision variables. The encoding can be in different forms such as string, integer, double values, binary values, or in other customized forms. However in any form of encoding, the encoding should enable genetic operators are to reproduce a new population, and provide a fitness function to evaluate the quality of a solution encoded in a chromosome. For example in integer encoding, an integer vector $x = (x_1, x_2, \dots, x_n)$ represents a possible allocation of n items to k brokers. Here $x_i \in \{1, .., k\}\ \forall i \in \{1, .., n\}$ and is the *id* of the allocated broker for item i For example if there are 5 items and 3 brokers the chromosome (3,1,1,2,2) means that item 1 is allocated to broker 3, item 2 to broker 1 and so on. A single point crossover is used to generate the children. So for example if two parents are (3,1,1,2,2) and (2,3,1,2,3) and the randomly selected crossover point is at (3,1,1↓,2,2) then the two children are (3,1,1,2,3) and

(3,1,1,2,3) .

A Definition of the Constrained Optimisation Problem

Many search and optimisation problems in science and engineering involve a number of constraints that the optimal solution must satisfy. In general, a constrained numerical optimisation problem is defined as below [118, 119]:

Find $\overline{x}$ which optimise

$$z = f(x_1, x_2, \dots, x_N), \quad x \in \Re^N$$

Subject to:

$$g_j(\overline{x}) \leq 0, \quad j = 1, \dots, J$$

$$h_k(\overline{x}) = 0, \quad k = 1, \dots, K$$

where $\overline{x}$ is the vector of solutions, $\overline{x} = [x_1, x_2, \dots, x_N]^T$, J is the number of inequality constraints and K is the number of equality constraints. Normally, equality constraints can be transformed into inequality constraints using the form:

$$|h_k(\overline{x})| - \varepsilon \leq 0$$

where ε is a very small value and is the allowed tolerance.

A feasible policy is a policy that satisfies all the predefined constraints, whereas an infeasible policy cannot. The set $S \in \Re^n$ defines the search space, and $F \subseteq S$ defines the feasible part of the search space. The unfeasible part U is the remaining set. If no constraints are given then F equals S. The following figure shows one example of a search space S and its feasible area F.

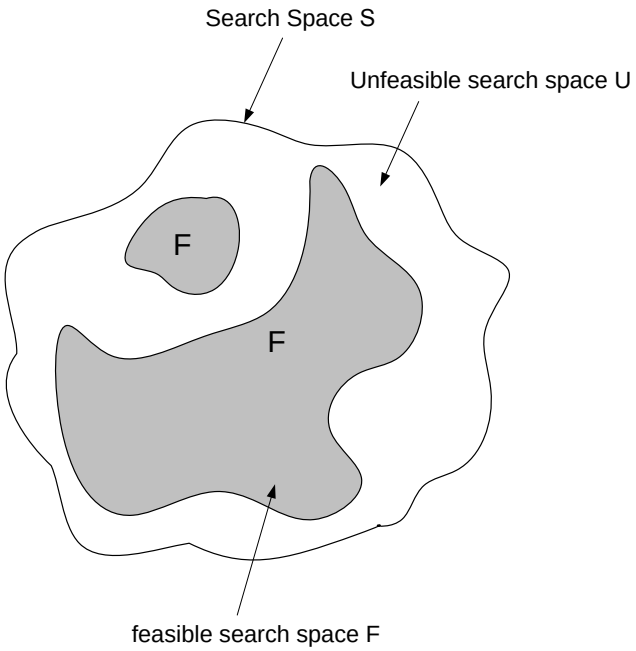

Figure 4-8 A search space [118]

The main features that make a global constraint optimisation problem difficult to solve are described by Michalewicz & Schoenauer [118], and are:

1. The type of objective function (such as linear or non linear function).
2. The types of constraints (such as linear or non linear constraints, equality or inequality).
3. The number of constraints: the difficulty in satisfying constraints will increase (generally more than linearly) with the number of constraints.
4. Connectivity of the feasible region (disjoint or connected).
5. Size of the feasible region with respect to the whole search space. Many constraint handling methods fail when the ratio of the feasible to infeasible area is too small.

In general, problems with non differentiable objective functions or even non differentiable constraints, and problem with disjoint feasible regions will be difficult for any mathematical programming technique [120]. To resolve these

kinds of constraint optimisation problems, EA (Evolutionary Algorithm) can be a very competitive method as EAs do not require that the objective function or the constraints of a problem to be continuous. If they are differentiable they must be continuous. Additionally, being population-based techniques, EAs are less prone to becoming trapped into local optima, even when dealing with fairly large and complex search spaces [120].

However EAs are unconstrained search techniques. Thus, incorporating constraints into the fitness function of an EA is an open research area. Constraint handling approaches tend to incorporate information about infeasibility (or distance to the feasible region) into the fitness function in order to guide the search into feasible region. The major difficulty to be solved is how to balance the pressure of feasibility of a solution with the pressure to optimise the objective function [118].

4.5.5.3 GA Configuration

The GA to solve the workload allocation problem is formulated as described in the following. Each gene on a chromosome represents a subset of workload to be allocated. A chromosome then is a solution of allocating all the workload to available brokers. The fitness function evaluates the quantified risk of overloading a broker. The detailed configuration of chromosome representation, fitness function, penalty, and genetic operator of GA is as below:

Chromosome Representation of a Solution

In the following illustrations, integer valued chromosome is used. And one gene represents a subset of workload - in the case illustrated a basic unit of workload subset allocated is the subscriptions of one topic. Then in a chromosome$x = (x_1, x_2, \dots, x_n)$, the value of the i^{th} gene x_i represents the *ID* of the broker that is assigned to carry the corresponding i^{th} subset of workload i.e. subscriptions of a topic. A complete chromosome $x = (x_1, x_2, \dots, x_n), where\ x_i \in \{1, .., k\}\ \forall i \in \{1, .., n\}$ is the complete solution of

allocating all workloads to the given *k* brokers.

Fitness function

Fitness function is obtained from generalized utility function presented in section 4.5.2 which represents the overall system resiliency. The utility or objective function is

$f_o(x) = f_o(x_1, x_2, \dots, x_N) = \prod_{i=1}^{K} p_r\left(V^{j_i} < C_i^{th}\right)$

The V^{j_i} is the incoming message rate arriving at broker i and j_i is all the workload allocated to broker i. $f_o(x)$ represents the probability that all the brokers are working and not overloaded. It is transformed to sum of log values as an easy to use fitness function. Then the GA uses the following fitness function $F(x)$:

$$F(x) = F(j_1, j_2, \dots, j_k) = \sum_{i=1}^{K} log\left(p_r\left(V^{j_i} < C_i^{th}\right)\right) + f$$

The fitness value is $F(x)$ is log $(f_0(x))$ added with a positive offset value $f = 10$. It is because the GA framework chosen, JGAP (Java Genetic Algorithms Package), requires positive fitness value only.

Constraints as penalties

The solution to the workload allocation problem can be subjected to *m* constraints. Here *m*=2, constraints are put on the probability of a broker is not overloaded and the processing delay is within prescribed bounds, i.e.

$$s.t.\ p_r\left(V^{j_i} < C_i^{th}\right) > c^{th}$$

$$s.t.\ d < d^{th}$$

th_c and th_d are thresholds that are set to check the basic quality of a solution. Here every constraint is encoded as a penalty added in the fitness value to prune the search space.

$$F'(x) = F(x) - Penalty$$

The penalty is a non-positive value defined from every individual constraint as:

$$Penalty = \sum_{l=1}^{m} \max(0, penalty_l)$$

$$penalty_1 = 10 \times (c^{th} - p_r(V^{j_i} < C_i^{th}))$$

$$penalty_2 = 10 \times (d - d^{th})$$

Initial Population

The first step in running GA is to create an initial population. This is the population to start to the evolution and create future generations of child populations. Since the solutions of the problem are encoded as a set of integer valued genes, the straight forward way to create initial population is randomly generated integer numbers as broker ID within the range [1, K] to each gene. Here an enhancement is made to the straight forward approach: a reasonable solution calculated from Round Robin allocation is made an individual chromosome in the initial population, and the other initial chromosomes are randomly generated.

4.5.6 Evaluation of the GA and RR Workload Allocation

A GA has been presented to extend workload allocation with an approximation algorithm basing on linear multiple regression performance models. To evaluation of this workload allocation mechanism, a Round Robin allocation is used as a comparison. As Round Robin is straight forward allocation mechanism and as a de facto method it is widely used in enterprise data centres when workload allocation is required. However the random nature of Round Robin allocation means that it does not provide allocation solutions with a consistent quality. Hence better allocation methods should be designed and here Round Robin is a practical comparison with the novel workload allocation mechanism.

The approximation algorithm is a big part of the workload allocation mechanism. The GA uses Round Robin as a starting point to search for better quality solutions.

4.5.6.1 Configuration of the test

This section describes the experiment to validate the performance of the GA described to solve the workload allocation problem. In the test, the workload that consists of 100 topics is to be allocated among 3 brokers. In GA, a solution is represented as a chromosome. So of allocating 100 topics is encoded as a vector of 100 integers $x = (x_1, x_2, \ldots, x_{100}), where\ x_i \in \{1, .., k\}\ \forall i \in \{1, .., 100\}$, and here k=3 representing 3 candidate brokers. Each gene represents an item from the overall workload, and its value is the broker ID that it is assigned to.

In order to test the algorithm under different settings, variance and covariance matrices of message rates with different characteristics are generated as test input. It is used to capture different load settings for the workload allocation problem and to allow the influence on the on the accuracy and efficiency of GA to be examined. Even if the items carry the same mean value of workload, the covariation among items still affects the probability of overloading the brokers' capacities. Therefore this workload allocation algorithm is tested for its capacity of the solving the modelled problem and improving overall system stability.

The broker processing delay is modelled from the previous described linear multiple regression model, with the parameters from the test work by [111]. The processing delay is determined from the configuration and regression parameters calculated from performance experiments under the corresponding configurations. In that work [111],the authors calibrated the performance parameters of a number of MOM systems, that includes WebsphereMQ, ActiveMQ, and etc. The processing delay is formulated as:

$$d = t_{rcv} + n_{fltr} \cdot t_{fltr} + r \cdot t_{tx}$$

The maximal incoming throughput is inferred from the processing delay: $c = \frac{1}{d}$. Here n_{fltr} and r are the number of filters and replication grade associated with messaging topics, they are part of the configuration for the

workload and messaging system. t_{rcv}, t_{fltr} and t_{tx} are processing delays that are modelled and calculated from regression tests.

Note that some constraints are applied these parameters are determined when the workload does not overload and decrease the system performance. From the experiments in [111] if the incoming message rate is above c, or message replicate grade r grows large, the processing delay d will significantly increase and hence reduce performance.

In the prediction model the replication grade r of a set of workload is calculated through a weighted average with the mean message rates. This set of workload may consist of one or more items.

$$r = \frac{\sum_{i=1}^{I} V_i r_i}{\sum_{i=1}^{I} V_i}, \quad in\ which\ r_i = \frac{\sum_{t=1}^{T_i} V_t r_t}{\sum_{t=1}^{T_i} V_t}$$

Here I is the number of items in the set of workload, V_i is the incoming volume of item i and r_i is the replication grade for item i.

The number of filters n_{fltr} in the prediction model is approximated in a similarly way:

$$n_{fltr} = \frac{\sum_{i=1}^{I} V_i n_i}{\sum_{i=1}^{I} V_i}$$

$$n_i = \frac{\sum_{t=1}^{T_i} V_t n_t}{\sum_{t=1}^{T_i} V_t}$$

The values of other performance parameters of the prediction model are obtained from multiple linear regression and least-square approximation. In this experiment here to evaluate the workload allocation mechanisms, the parameter values calculated and validated by experiments in [111] are used. The MOM experimented with is one of the most widely used open source implementation ActiveMQ. The performance parameters calibrated in their work, as in the following table.

Parameter	t_{rcv}	t_{fltr}	t_{tx}
second	4.8810^{-5}	$1.62*10^{-7}$	$1.64*10^{-5}$

Table 4-5 Broker performance model parameter

An average replication grade and number of filters installed for each message are assumed to be valued: r = 10, n = 3. With all the broker performance parameters set as above, the multiple linear regression model estimates the average of normal message processing delay as,

$$d = 0.000213296 \text{ sec}$$

The maximum processing rate for incoming message is then

$$\text{th} = \frac{1}{\text{d}} = 4688 \text{ msg/sec}$$

The experiments assess the effect of temporal correlation in the publishing rates. In the tests the messaging statistic parameters such as means, coefficient of variations, etc. remain the same. So does the MOM performance parameters. With these parameters two different correlation matrix are used to show the effects of different co-variation levels among messaging workloads on overall system performance. One correlation matrix consists of both positively and negatively correlated workloads, and the other one with only positively correlated workloads. A valid variance and covariance matrix must be a semi-definite matrix. To assure the validity of testing data, the matrices are checked and if it is not a valid correlation matrix, it is adjusted to its nearest correlation matrix using the tool provided by [121]. The tool is developed to solve the nearest correlation matrix problem, which preserve as much as possible the defined characteristics.

Note that both the mean values and variation level of messaging workload contribute to the overall system load. So given a predefined correlation matrix, the overall system load can be adjusted by choosing the mean values and corresponding coefficients of variation of workloads. For a chosen coefficient of variation C_v and mean μ, the standard deviation of message rate of an item σ is

$$\sigma = \mu C_v$$

In the evaluation the overall load is adjusted to a level that is high enough to stress the system performance.

In the evaluation both Round Robin allocation and GA approximation are tested. The GA uses a given Round Robin as a starting gene in its initial population and search for better solutions in its process of evolution. So to test the effectiveness of GA as enhancement for Round Robin, three Round Robin solutions with different level of quality are used to evaluate the improvements of GA in each case.

4.5.6.2 Results of Workload Allocation Evaluation

As described above, the evaluation is to allocate 100 items to 3 brokers. The tests are done here with two types of correlation matrices. *Test A* uses a correlation matrix consists of both positively and negatively correlated workloads, and *Test B* uses a correlation matrix with only positively correlated workloads. The other statistical parameters remain the same for the tests. This is configured to show the effects of different co-variation level among messaging workloads towards overall system performance. It is expected that among these two, the correlation characteristics dictate that the first correlation matrix can result in the better allocation solutions.

Furthermore, in these tests, Round Robin solutions with three different levels of quality are used to evaluate the improvements of GA, including one good solution, one weak solution and another medium one with quality in between. The objective is to examine the effectiveness and improvements of GA solution given possible different Round Robin solutions. Especially when Round Robin does not produce a strong solution, will the GA solution make an improvement and make the allocation more resilient against bursty workload.

The resulted GA and Round Robin solutions in both tests are as following:

Test A	Round Robin			GA		
	P_{B1}	P_{B2}	P_{B3}	P_{B1}	P_{B2}	P_{B3}
Case A1	0.856702	0.856702	0.856702	0.895866	0.895866	0.895866
Case A2	0.829735	0.829735	0.856702	0.895865	0.895866	0.895866
Case A3	0.752948	0.752948	0.856702	0.895865	0.895866	0.895865

Table 4-6 Test A, for workloads with both positive and negative correlation (P_{BX} is the probability the broker Bx is working normally, i.e. under workload threshold)

Test B	Round Robin			GA		
	P_{B1}	P_{B2}	P_{B3}	P_{B1}	P_{B2}	P_{B3}
Case B1	0.802357	0.802357	0.802357	0.826069	0.826069	0.826069
Case B2	0.794602	0.794602	0.802357	0.826069	0.826069	0.826068
Case B3	0.752948	0.752948	0.802357	0.826069	0.826068	0.826068

Table 4-7 Test B, workloads with positive correlation only (P_{BX} is the probability the broker Bx is working normally, i.e. under workload threshold)

In the above results, the two test data sets are processed by both Round Robin and GA algorithms. In Test A and Test B, case 1 starts with the good quality Round Robin solution, case 2 the medium quality solution, and case 3 the weak solution. These three solutions are used by GA algorithm as one of initial population to produce respected allocation solutions for comparisons. P_{Bi} is the probability that the i^{th} broker works under prescribed workload threshold.

The results show that, firstly GA produces relatively consistent solutions throughout the two tests, despite of different quality of Round Robin solutions. GA makes improvements to Round Robin allocations in all cases. Secondly, the GA produces better quality allocation results in Test A where correlation matrix used has both positively and negatively correlated workloads. Thirdly, for the Round Robin solution of weaker quality, more significant improvements are made against them.

The overall system wide improvements on all 3 brokers can also be illustrated

clearly with the joint probability $P_{sys} = P_{B1} \cdot P_{B2} \cdot P_{B3}$, which is also used in the GA to calculate utility values. It represents the probability that all brokers in the system are working under a prescribed risk threshold. The detailed values are shown in the following table, where $P_{sys.RR}$ is the value of Round Robin solutions, and $P_{sys.GA}$ is the value of GA solutions. Figure 4-9 shows the percentage of improvements that $P_{sys.GA}$ made over $P_{sys.RR}$, through all cases of both the tests. The blue line represents the proportionate improvements of Test A, and the red line represents Test B.

	Test A		Test B	
	$P_{sys.RR}$	$P_{sys.GA}$	$P_{sys.RR}$	$P_{sys.GA}$
Case 1	0.628766	0.719001	0.516538	0.563701
Case 2	0.589805	0.718999	0.506601	0.563701
Case 3	0.48569	0.718999	0.454881	0.5637

Table 4-8 Joint probability that the system is working normally ($P_{sys} = P_{B1} \cdot P_{B2} \cdot P_{B3}$)

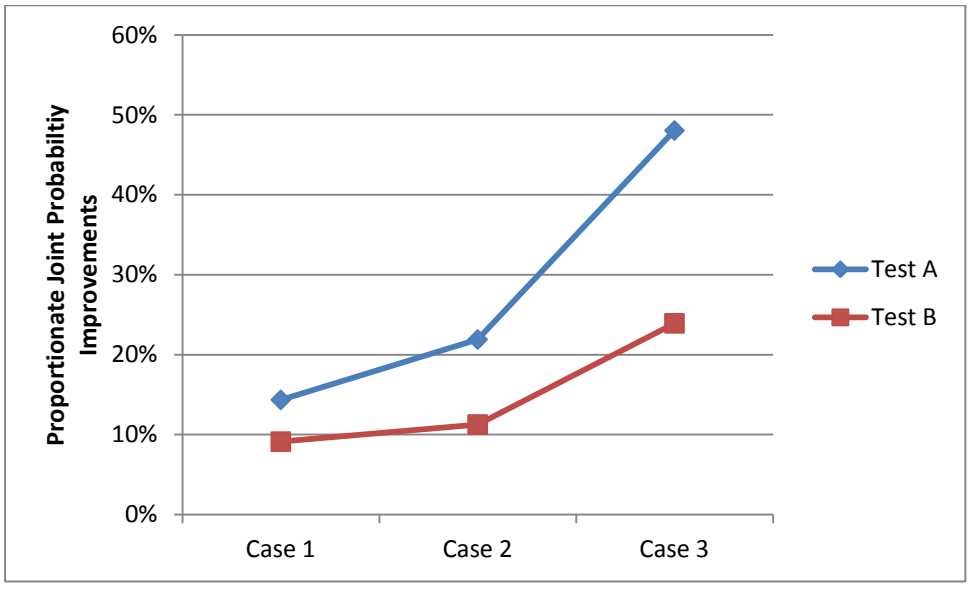

Figure 4-9 Proportionate improvements of GA against RR for joint probability the system is working normally

In both tests, the system wide joint probability improvements in the above figure are defined as:

$$(P_{sys.GA} - P_{sys.RR})/P_{sys.RR}$$

It is shown that bigger improvements are made by GA when Round Robin produces bad solutions as in Case 3. It means GA can find a more reliable solution when a risky allocation is produced by Round Robin.

These improvements are made because GA is capable of exploiting the smaller positively correlated or negative correlated workloads to find more stable allocation. The result of Test A is superior to Test B. Because with the temporal correlation of A, workloads with negative correlations can potentially be assigned to the same broker, so that the system risk is lower in busy time. Overall these tests demonstrate that GA approach is effective in consistently producing more resilient allocation to safe guard system performance, while weak quality solutions may be produced by Round Robin.

4.6 Mirroring and Load Balancing

4.6.1 Workload Mirroring

Introducing Redundancy

In the above section the workload allocation mechanism used to allocate each unit to one broker without redundant backups. However in practical cases, MOM systems need to deal with the situation that the primary broker carrying the workloads stops working or work with degraded performance due to transient or long term fault. Alternatively it may be intentionally shut down for system administration. Moreover some critical workloads call for resilient service with minimal interruption even in the face of system faults.

Thus the common solution is found for this situation, which to replicate the subscription information and sometimes the actual message or even entire primary broker state to another broker within a cluster. With such redundancy set up, the messaging can be in more resilient in at least two ways: either the messaging is served simultaneously by more than one broker, or it can be quickly switched from a primary broker to another backup broker that has replicated subscription information and broker state.

Workload Mirroring as Redundancy Mechanism

The workload allocation mechanism in a MOM system needs to be able to accommodate such redundancy for backup workload when it is required. The risk-aware allocation algorithm introduced in previous sections can be applied with a different setting to solve the *redundancy* or *mirroring* allocation problem. In RQMOM, the replication of redundant workload is referred as the *workload mirroring*.

In the MOM system, some subscriptions of more importance can express the need for extra resilience provided through redundancy. For these subscriptions in a MOM system that require the extra redundancy, mirroring solution is calculated by allocation algorithm, and then in the mirroring process, these subscriptions information on a original primary broker is replicated to a secondary *mirror broker*. The workloads being replicated to the mirror broker is known as *mirrored workloads*. Conversely, different from workload mirroring, alternatively current High-Availability clustering implementations, such as Qpid, RabbitMQ and etc., replicate the entire primary broker state to another back-up host. With workload mirroring it is not necessary to have spare back-up hosts for each broker as High-Availability clustering does, since the subscriptions can be easily propagated to another broker online. The backup is spread across the other brokers, which are actively serving publishers and subscribers. In the situations of failure or overload of primary broker, the mirrored subscriptions allow publisher and subscriber clients to switch quickly to back-up brokers as a fail-over or load balance mechanism. RQMOM use the same principle as the previous presented workload allocation mechanism to allocate the mirror workloads within a local domain of brokers.

Specifically, the solution for workload mirroring is calculated as follows:

$$\arg \max_{\{l_1 \ldots l_k\}} \prod_{\mathrm{i}=1 \ldots \mathrm{k}} p_r\left(I^i < I^{th'}\right)$$

$$\text{s.t.} \quad j_i \cap l_i = \phi$$

$$\text{s.t.} \quad d^i < d^{th'}$$

The p_r is calculated in the same way as described workload allocation. Also the same broker performance model by Henjes [111] in the above section is used to predict broker performance. This model is used to predict MOM system performance under different workloads and configurations. The a**verage message processing time *T*** is predicated as:

$$T = t_{rcv} + n_{fltr} \cdot t_{fltr} + r \cdot t_{tx}$$

A detailed description can be found in section 4.4.2.

Now given the original workload allocation of k brokers $\{j_1 \dots j_k\}$, it is the problem to find out the best solution $\{l_1 \dots l_k\}$ to allocate additional workload from a working broker that requires to be mirrored, assuming that this broker would fail. In the total k brokers, the original workload plus mirroring workload will be $\{j'_1 \dots j'_k\}, where\ j_i^{'} = j_i \cup l_i$, and it subject to $j_i \cap l_i = \phi$ such that the mirror will not be on the same broker as the original.

With the original workload allocation mechanism in place, the workload mirroring problem is solved in this following process:

There are k brokers, suppose the failing broker is B_k. (where k is chosen without loss of generality) of the remaining working brokers are $B_{1,} \dots , B_{k-1}$ and these have their previously computed load allocation. We solve the problem of allocating the load on B_k into the remaining capacity on $B_{1,} \dots , B_{k-1.}$ The same algorithm as used in the previous section is used. The difference being the available capacity is no longer the full capacity, but only that which is left.

For each broker *B* in the local domain

Given the solution from original workload allocation problem

The solution is the mirror allocation of the workloads from *B*

Obtain the complete mirror solution from all brokers

Apply the workload mirroring, by replicate workload

Figure 4-10 The workload mirroring mechanism

The workload mirroring solution follows the same principle as the risk-aware mechanism evaluated above. In the case when the workload from primary broker needs to switch to its mirror, the risk of overloading a mirror broker will be contained. It should be noted that switching can be implemented very effectively without message loss if the messages are replicated to the mirroring brokers. This is not an issue as the brokers have this capacity. However, this is not a necessary assumption.

4.6.2 Integration with Reactive Load Balancing

The risk-aware work allocation contains the system overloading probability pre-emptively. However, when the unexpected overloading or faults do occur, additional reactive mechanisms such as load balancing can mitigate the events reactively. Widely used reactive load balancing is a complement to proactive allocation mechanism. Most popular MOM implementations include some mechanisms for load balancing. This thesis will not go to the describe load balancing implementations. The fundamental elements are metrics to detect possible overload, selection of the load-accepting broker, and parameters in state transitions of the load balancing process.

4.7 Summary

This chapter presented the architecture and operation of the local domain system. A proactive risk-aware workload allocation and mirroring mechanism is developed. This mechanism formulates a risk-aware workload allocation problem, and uses an allocation algorithm that minimizes the risk in a quantified way and mitigates correlated bursty workload to different brokers. The mirroring solution adapts the system to tolerate broker and faults and failures in the application end points in local domains.

In the allocation phase, in contrast to the load balancing phase, the time taken to create a solution is less critical, but it is important that a solution near optimal can be found. In this chapter an approximation approach using

genetic algorithms is designed and validated. The fitness function, the chromosome structure, how the crossovers are accomplished, and how the constraints are handles are described. The GA approximation algorithm is evaluated against widely used Round Robin workload allocation. It is shown that the GA produces solutions with consistent quality and it is effective to safe guard the system against possible weak Round Robin solutions.

Reactive load balancing mechanisms mitigate overloaded or faulty brokers when any performance metrics degradation is detected in a broker. The load balancing mechanism finds a healthy load accepting broker from system performance information exchanged among local domain brokers. Then it finds appropriate subscriptions to offload by estimating the performance metrics change on both offloading and load-accepting brokers. These have not been discussed but they are not the focus of this thesis.

Chapter 5 Federated Overlay Domain Resilience and QoS Support

5.1 Introduction

In the federated overlay domain, since the different local domains are deployed over a large geographic area, they will inevitably operate over wide-area networks, where the link quality fluctuates due to the dynamic traffic load in the substrate underlay network. The design goals of inter-domain messaging on the federated MOM overlay is to provide the performance that satisfies the clients QoS requirements and reliable messaging despite the dynamic delay and connectivity over WAN or internet, for mission critical applications. In contrast to local domain messaging where networking is very reliable and latency is small or even negligible, the networking performance and connectivity over WAN is dynamic and it is the key factor in inter-domain messaging.

The focus in this work is to cope with geographical correlated failures. Mitigation mechanisms against geographical correlated failures have not been studied extensively, despite many responsive overlay routing methods have been designed to random network degradations or faults such as congestion, inter-AS link failure, random physical network fault, etc.

RQMOM constructs an overlay network of federated gateway brokers on top of the physical topology. On this federated overlay the routing building blocks described in this chapter employs overlay level multi-path routing, in which besides a standard primary path between each source node and destination node, best alternative overlay path(s) are also chosen through a separated decision process. Then the source node enforces the path selection by commanding the downstream overlay nodes to route messages to the next hop that it specifies.

The design goal of inter-domain messaging on the federated overlay is: firstly the end-to-end networking of the federated MOM overlay should fulfil the real time QoS requirements of different applications; and secondly the federated MOM system should be resilient, which means the MOM system should maintain an acceptable level of service in critical situations when the system is stressed by faults or external challenges, and also to effectively recover from faults, e.g., broker or link degradations. As stated before, this work has a focus to address resilience mechanism against geographical correlated failures. To achieve this design goal, a set of routing mechanisms are employed, including primary path and backup path selection and path maintenance. These will be presented in this chapter. Specifically after a primary path that satisfies QoS requirements is first selected, a corresponding backup path is calculated. Different from traditional backup paths selection approaches, a geographical proximity-aware backup path selection algorithm is designed. The proximity-aware algorithm will be shown to make the overlay routing more resilient to large scale geographically correlated failures, in certain conditions that is important to many mission critical applications.
This chapter is organised as follows. First the scope of the research is outlined. Section 2 introduces important related work. Then section 3 presents the routing algorithm employed and section 4 the novel routing algorithm is evaluated against a commonly used benchmark algorithm. Section 5 concludes the chapter.

5.1.1 Research Motivation and Scope

First it is necessary to introduce some fundamental ideas behind overlay routing and some motivations for the research. Then this section explains the rationale of the overlay routing methods employed for more resilient networking against geographical correlated failures.

Why Overlay Routing?

As an enhancement to routing in IP networks, routing by overlay networking has been a very active topic. Its benefits have been revealed by the previous work. IP-level underlay routes are selected and maintained by intra and inter domain routing protocols, which are managed by independent operators. Beyond the fixed IP routes in substrate networks, routing with an overlay network provide overlay nodes with responsive alternative overlay routes. The alternative overlay routes allow nodes to make path selection according to customized mechanisms. Hence it enables overlay traffic to exploit the alternative paths and quickly bypass congested or unavailable IP substrate paths and also enhances QoS support for applications. In addition, routing with overlay networks shows capacity to improve the end-to-end QoS even under normal circumstances. This is due to the limit in path selection mechanisms in the internet. For instance, the study [58] shows that in 2001 for almost 80% of the used paths in the Internet, there is an better alternate path with lower probability of packet loss.

The merits of overlay routing are more crucial to resilient messaging in the face of failures associated with large damage.

The Internet is composed of a federation of autonomous systems (AS). Each AS is normally controlled by a single entity, typically an Internet service provider or a very large organization, with connections to other AS domains. Within an AS equipment adheres to a single and clearly defined routing policy and allied forms of signalling. This common administration makes it possible to support resilience mechanisms within the domain at various levels as appropriate, or desired. These can include physical layer resilience within the optical transport network (such as 1+1 or 1:N backup path protection) and/or data-link and network layer resilience via MPLS (Multiprotocol Label Switching) or IP fault recovery mechanisms.

However, between AS domains reachability information is exchanged using the Border Gateway Protocol (BGP), which is a path vector routing protocol with the facility to implement policy restrictions. The level of trust required between operators is limited and the internal architecture of the AS domains

remains hidden. This low degree of transparency and cooperation is at least due in part to potential security concerns. However, it imposes severe limitations on how end-to-end pathways can be established and maintained with certain Quality of Service (QoS) guarantees. As things stand, the Internet simply provides a best-effort inter-domain delivery mechanism. To do more one could allow signalling mechanisms to reserve resources end-to-end across multiple domains using functional units such as the ITU-T next generation network (NGN) resource admission control function (RACF) typically in concert with path computation elements (PCE). However, this requires operators to process each other's signalling messages setting aside resources, as required. This is unlikely to happen. The problem remains that BGP, used for routing across ASs, is characterized by re-convergence time of several minutes or longer [122] so there is considerable demand to devise resilient solutions for mission-critical applications.

For example, in comparison with transient intra AS failures, inter AS failures which are often caused by BGP problems, are not easily to bypass with IP level routing since these BGP failures cost minutes or tens of minutes to converge; in other cases, some failures are hard to recovery due because their invisibility at routing level. The can be caused by configuration mistakes, malicious attacks, or hardware and software bugs and etc. Moreover if a major failure is caused by physical network outage, then on one hand, an arbitrary time is required to repair the physical outage, and on the network level it might takes the IP protocols unpredictable time to reconstruct a new route. With overlay routing, the alternate overlay paths and routing mechanisms provide better resilience and quicker response to these problems. In addition a single path may be insufficient to provide the resilient uninterrupted messaging required by mission critical applications. Hence multiple parallel end to end paths are in need. Routing with parallel paths is currently not supported by Transmission Control Protocol (TCP), but it is well supported by various overlays.

In inter-domain federation applications are often distributed across different networks and geographical locations. Hence, we focus on overlay-based data dissemination mechanisms and explore their ability to tolerate possible large geographically correlated failures. Due to the tight correlation between multiple overlay links and a single physical link, a few physical failures may affect lots of overlay links. To enable reliable dissemination under such conditions, we propose overlay multi-paths construction methods that incorporate proximity-aware neighbour selection methods to improve the performance of the overlay data dissemination, to the extent possible, in terms of reliability and latency. In this approach, the overlay nodes select neighbours that are most likely to be distinct in the presence of a geographical failure. By comparing with related approaches we show how an overlay MOM structure constructed using our proximity-aware route selection techniques can maintain a larger amount of uninterrupted overlay connections for disseminating data under various geographical failure conditions.

One application of RQMOM is the communication middleware for knowledge-based Decision Support System (DSS) for disaster prediction and reaction. The DSS requires different platforms and applications to communicate with each other, such as sensor data collection, fusion and analytics and controller command communications, backend processing applications, administration hosts, mobile messaging clients, social media platforms etc. Two scenarios from an EU project TRIDEC (Collaborative, Complex and Critical Decision-Support in Evolving Crises), demonstrate the need for such resilience. Both involve intelligent management of large volumes of data for critical decision-support. The first scenario concerns a large group of experts working collaboratively in crisis centres and government agencies using sensor networks. Their goal is to make critical decisions and save lives as well as infrastructural and industrial facilities in

evolving tsunami crises. The other scenario concerns a large group of consulting engineers and financial analysts from energy companies working collaboratively in sub-surface drilling operations. Their common objective is to monitor drilling operations in real-time using sensor networks, optimizing drilling processes and critically detecting unusual trends of drilling systems functions. This prevents operational delays, financial losses, and environmental accidents and assures staff safety in drilling rigs.

The geographic location of overlay and physical (underlay) nodes are available. In real world scenario, overlay nodes (brokers) can be located with GPS. However the accurate geolocation of physical underlay nodes require either information from an ISP or other geolocation databases such as googlemap or MaxMind. (MaxMind claims to map 85% IP addresses in US within a 40.2 km radius). With the support of ISP and user contributed geo-location data, companies like Google and Baidu have continuously improved accuracy of the geo-information for its business, such as IP based location services. This makes geo-location based routing possible. Besides information from geolocation mapping data bases, IP and geolocation mapping algorithms have been researched to estimate unknown locations. They usually have a much larger error margin, such as [123-125], claim map IP addresses to geolocations with median error distance of 134, 67, and 35 km radius in their experiment. By using additional information [126] manages to reduce the median error distance to 690m.

Variations of Overlay Routing

For a messaging service with stringent time constraints, two options of routing protocol are available to deliver a publisher's message to its subscribers. The first is to construct multicast trees that are rooted at the publishers and spans all the destination brokers where subscribers are attached to. This tree-based approach [76, 77, 127, 128] is found to be scalable; however it is more sensitive to node and link degradations. This is because a

faulty node or link in a tree will affect all its child nodes. Hence tree-based multicast needs additional maintenance mechanisms [127, 129, 130] to be resilient. The second is more appropriate in the scenarios with relatively small number federated overlay domains; and point-to-point overlay paths are constructed from the publisher's ingress broker to every destination brokers that subscribers are attached to [62]. Because it is easier to monitor and maintain point-to-point paths, the second approach is more responsive when managing faults and congestion. It is the method that is most employed in current enterprise messaging scenarios where a small number of federated domains are interconnected. Hence our research is focused on the second approach.

Basing on practical networking scenarios between federated enterprises systems, it is assumed that the set of local domains in the federation is known in advance, and the topology of the federated MOM overlay is also decided a priori. Nevertheless, these gateway brokers and overlay links may fail and recover at any time. This assumption that the overlay topology is known in advance is reasonable in many application scenarios because the federation members only change on a coarse timescale (e.g., once in a few weeks). However, when brokers do frequently join and leave, to adjust the overlay topology in runtime, dynamic topology design and maintenance scheme is needed. This case is left for future study.

The resilience and QoS of the messaging substrate plays a critical role in the overall system performance as perceived by the end users or applications. While some organizations may employ dedicated networks to provide QoS assurance, in general deployments of federated messaging this assumption of the underlying network does not always hold. In the federated overlay domain, since the different local domains are deployed over a large geographic area, and over heterogeneous networks, the link quality fluctuates due to the dynamic traffic load in the substrate underlay network. Such a dynamic network model poses challenges to our design as the messaging service must adapt to such network and system dynamics,

and ensure the end-to-end latency requirement is continuously satisfied. The example applications in this scenario include federated inter-domain messaging among a few separated enterprise data centers, and integrating distributed sensors and actuators with back-end processing capabilities in remote control centers.

5.1.2 Routing Design

Overlay routing mechanisms often work on different aspects or layers of a network. A complete routing mechanism may include these functional components: a path or multipath selection module; a transmission module which may employ a scheduling and traffic splitting function, or simply simultaneously send duplicated traffic along multipath; a path maintenance and recovery function etc. This work is motivated by the need to providing resilient uninterrupted messaging despite the potential damages caused by geographical correlated failures. The focus of this work is the path selection and, in particular, the geo-aware alternative path selection mechanism for a more resilient multipath routing against large scale geographical correlated failures. Many aspects of an overlay routing scheme, such as path maintenance and traffic scheduling have attract many research efforts. However the alternate path selection algorithm against geographical correlated failures has not been well reached. This also motivated this research work.

One advantage of the novel geo-aware alternative path algorithm (GAP), presented in the following sections, is that it is independent from the primary path selection mechanism. This means for a primary path selected with any mechanism, e.g. shortest path algorithms, predefined source routing mechanisms by administrators, an alternative path will be always be found to enhance the primary path. This flexible feature is very useful in real world application when overlay paths are sometimes selected using different independent mechanisms.

In RQMOM, once the primary and alternative paths are established, the

source node simultaneously sends replicated messages along both paths towards the target node. This approach is known as simultaneous multi-path routing, which improves the resilience of communication by using extra redundancy. Besides the simultaneous multi-path routing different schedule and traffic splitting techniques that can also be used on top of path selection algorithms to send messaging traffic among a multipath set. This work chooses the simple simultaneous multi-path routing to focus on maintaining uninterrupted messaging.

In the evaluation section in this chapter, it is assumed that the establishment of overlay paths is enforced by source nodes. It means after the alternative paths are found, source node commands other nodes along the path to set up overlay routes, and maintain this route or make other routing decisions. However to respond to unexpected networking faults, hybrid mitigation and recovery mechanisms are also often employed in addition to simple source routing. In real world systems, enterprise MOM systems are mostly deployed in a well-established network environment. Since quality of network is stable, the overlay networking is also stable. Source routing is used in many existing MOM implementations. Hence alternate path algorithms fits naturally to the routing functions that the existing MOM systems provide. These setting makes it convenient to focus on evaluating the effectiveness of this routing approach, under geographical correlated failure scenarios. Specifically the geographical correlated failure scenarios evaluate if alternative paths returned by the GAP algorithm provide the expected geographical separation that makes the multipath more resilient. Beyond these current settings here, like shortest paths algorithms, such as Dijkstra's algorithm, GAP can also be decentralized and deployed in each individual note. Hence the algorithm can work beyond source routing.

The GAP selection with simultaneous multipath routing are designed to maintain uninterrupted messaging with best effort. As described previously, additional hybrid path maintenance and recovery mechanisms are often in place. This is because when these multiple paths between a source node and a

destination node are indeed simultaneously interrupted; propagating link states and re-establishing overlay paths put overheads to the recovery of connectivity of network. Such reactive path maintenance and recovery problems have been researched in relevant scenarios. These researches have shown good improvements to counter the network dynamics and faults and can be an effective addition to any resiliency-oriented network. The mechanisms include overlay based path probing, back up path maintenance, random forwarding and etc. However these are an independent area of research, not the focus of the thesis.

5.2 Background and Related Researches

5.2.1 Related Overlay routing protocols

One promising approach to providing this end-to-end resilience is the use of overlay networks, one example being Resilient Overlay Network (RON) [58]. This is proposed to provide better performance by actively monitoring the network among a group of participating nodes. The nodes in RON form a full mesh topology and use active probing to monitor the health of Internet paths included in the overlay. In contrast to the high convergence time of the Border Gateway Protocol (BGP), RON can achieve recovery time in the order of tens of seconds based on test-bed experiments. However, the overhead increases in the order of $O(n^2)$, where n is the number of overlay nodes. In [89], the authors discuss the relationship between the effectiveness of the overlay and the overhead consumption. The conclusion is that with a lower overlay node degree, a comparable Quality of Service (QoS) performance to that of full connectivity can be achieved.

Building on the benefits of overlay routing, studies have been conducted to use simultaneous multipath overlay routing or overlay backup paths to improve resilience of overlay network, and also to further optimize the

performance with available overlay paths. In the case of greedy overlay best path routing, where each node make the routing decision basing on solely its own interests, the studies [68-70] find that some restraints over greedy routing are crucial for obtaining better performance of route selection. Specifically, three forms of restraint, namely randomization of route selection, utilizing an appropriate hysteresis threshold when switching routes, and increasing the time intervals between route-change decision assessments has been researched. Their results indicate that firstly randomization can significantly reduce loss-rates, reduce flip flopping of path selections and more importantly, it is sufficient to utilize information from a small subset of overlay paths to obtain such results. Secondly, it is shown that appropriate values of the hysteresis threshold for switching to better routes, can be heavily dependent on the parameters of the system. The algorithm proposed by [70] determines hysteresis thresholds dynamically. Other works have been attempt to adapt traffic splitting and scheduling to the changing conditions. For example [131] proposes a biological inspired approach to maintain multipath routing. This approach is built on a nonlinear mathematical model, called the attractor selection model. The attractor selection model imitates adaptive behaviour of biological systems, e.g. bacteria, which adaptively adjust the rate of nutrient synthesis to keep alive and growing in dynamically changing nutrient condition. In the case many bacteria co-exist in a reactor, cooperative behaviour emerges through interaction among the shared nutrients in the reactor environment. By regarding bacteria as overlay networks, selection of nutrient to synthesize as selection of path to use, and the growth rate as the performance, the proposed mechanism is believed by authors to improve the adaptive and stable multipath routing without centralized coordination. This work adopts a simple simultaneous multipath routing to introduce redundancy for uninterrupted messaging as described previously in the ***5.1.2.***

Overlay topology construction has been researched from various aspects. For

example, in [132] and [133], different types of overlay topologies are analyzed for given overlay node locations. However, there remains debate on where to place the overlay nodes [134], [135]. For instance, [135] considers the question of how many ISPs and the number of routers inside one ISP network is enough for overlay routing. The correlations between direct and overlay paths between source and destination pairs using different available nodes are calculated. Then, the intermediate nodes are categorized into different performance clusters for overlay construction use. Both of them show that the diversity of virtual links is indispensable for providing better performance in overlays. However, they are both categorized as provider-dependent overlay architectures. Moreover, customers only use the overlay services in the case of internet path failures, which means they should be equipped with the ability to diagnose the health of the original path in a timely manner.

QoSMap [92] has been shown to provide high overlay resilience in a provider-independent overlay. It maps an overlay topology with specific QoS requirements onto a physical network topology by sequentially selecting the Planet-Lab nodes that can provide the best QoS performance. However, it strategically chooses the physical network to provide a diversified set of paths for the overlay construction. The possibility of overlapping between the virtual links that may result in concurrent failures is not addressed.

Most existing work on overlays focuses on improving QoS performance using a single intermediate node for detours. In [136], two availability models are proposed to define an overlay that can still be fully connected in case of no more than three physical node failures. They assume that all the physical nodes are overlay candidates and physical nodes with a lower degree than three are not considered. Moreover, they focus on proving the NP-completeness of the two models [137] and how to construct an overlay with high availability in a scalable distributed way [136].

More recent work [138] considers network provider independent resilience overlay networks where the overlay nodes are constrained to ASes with low connectivity as would be expected of smaller tier-3 stub networks. It shows that a near optimal overlay topology can be constructed using heuristic methods and is verified using different failure models. Traceroute is used to provide underlying physical resource diversity between the virtual links of the overlay topology and MPLS provides separate working and backup label switched paths between the resilient ingress and egress points.

5.2.2 Alternate path selection

Alternate paths as well as a primary path may significantly improve reliability and performance of network. Conventionally the classic shortest path algorithms are widely used in routing protocols to select the primary path between a source and target node. Besides the primary paths, additional alternate paths may significantly improve reliability and performance of network. Hence algorithms computing and deploying alternate paths in routing protocols have been researched, such as variants of K-shortest path algorithms (KSP), disjoint path algorithms, and etc.

Each of these algorithms has some drawbacks. For the K shortest path algorithms, after all paths are sorted, an additional path-checking procedure has to be performed to choose two alternate paths that satisfy the desired constraints for acceptable paths. Therefore, when the number of paths from a source to a destination is large, the K shortest path algorithms become very inefficient for selecting alternate paths. For the disjoint path algorithms, on the other hand, the alternate paths have to be determined by applying flow theorems. The final path set calculated by disjoint path algorithms is totally disjoint, but the primary shortest path of the network may not be included.

5.2.3 Related Algorithms

For the problems that can be formulated into a graph representation, search algorithms are widely used to traverse the graph and find solutions. The shortest paths algorithm and its variations are fundamental ones among search algorithms. One major usage of shortest paths algorithm is for routing entities to find the best path between a source node and its destination node(s).

Dijkstra's shortest path algorithm [139] solves the single source shortest path problem. Single source shortest path problem is finding the path with minimum distance from a given source node to all the other nodes in the network. It is a classic problem in graph theory and the solution algorithm of single source shortest path problem is a foundation to solve many other problems in graph theory. Dijkstra calculates the shortest path tree with the source node as the root and destinations as leaves.

Similar works to compute disjoint alternate paths have been researched. However their approaches are under the assumption that a complete disjoint path always exits, which may not be true in many circumstances. Furthermore, they do not address the scenarios where only partially disjoint paths are available. In these approaches, deleting the edges is equivalent to increase the weight of edges by $\Delta W = \infty$. A pre-trip alternate route planning scheme [140] by Rouphail uses a similar scheme basing on increasing the weight of edges on primary path P_0. Each edge of P_0 is increased by 20%, 50% and 100% of its original weight and the candidate alternate paths are recalculated with Dijkstra's algorithm. In contrast our GAP algorithm searches for an alternative path using geographical separation based penalty, and the details are described in 5.3.2.

5.2.4 Failure Models

This section addresses models of failures that damage reliability of message

dissemination over wide area. Independent random failures and geographical correlated failures are two models of failure. Independent random failures are caused by random events such as node, link failures and intra or inter domain routing failures. Because of the independent nature of these failures, the disruption of communication is localized in a single point or restricted in a relative small geographical area. Different from independent random failures, disruptions caused by correlated failures often have a large geographical span and hence can cause massive severe disruptions to the underlying physical infrastructure. The characteristics and observations for this correlated failure model motivate the design in this research. The alternate path selection mechanism in this work is assessed in geographical correlated failure scenarios.

The simultaneous geographical correlated physical failures can be caused by geographical events such as natural disasters such as earthquakes, flood, tornados; or blackouts from sudden power distribution outages. The geographical effects of a natural disaster are intuitive. Some aspects of them have been further researched. For example, in the case of earthquake, seismological studies [141] found that given two points on a plane, an exponential function with respect to the geographical distance between them are used to describe the spatial correlation of intensities of earthquakes.

Blackouts are caused by a continuous, cascading failure in the energy distribution grids. Energy distribution grids are in an interconnected network. Stations in the network usually serve their sub-regions that are in proximity. When an energy shortage occurs in one sub-region, the grids will redistribute its load to nearby grids. However in a major failure, such load redistribution may induce overload in those nearby grids and causing a cascading failure that propagates from the geographical start point toward near-by regions. The blackout in US [142] was an example of such cascading damage.

Some research work is based on reliability assessments, which means the probability of a path working. This is also called availability. It is assumed

that the probability of failure of every link in the overlay network is acquired through monitoring over time. Then the *resiliency of* an overlay path or multi-path is expressed as the probability that there is at least one QoS-satisfying path available between the connected gateway brokers. A path is considered available only when all links on the path are available. Thus, the resiliency of brokers along a path can be computed only with the assuming the availability and accuracy

Note that some random major failures within an IP infrastructure of one organization, that is geographically distributed, may also cause network outage in a wide spatial area. However damages from such failures are contained within the failed infrastructure and other network infrastructures in the same area are not likely affected by it.

5.3 *Path computation*

It is a common approach for the overlay network to support pre-establishment of multiple explicit end-to-end overlay paths between each source and destination node pairs.

In both overlay and underlay network, alternate paths of good quality can significantly improve the reliability of routing. This is because, at the time of congestions, faults or failures, switching to a pre-calculated alternative path is faster than determining a new path responsively. For simultaneous multipath routing, an active alternative path can enable uninterrupted overlay messaging in the face of primary path failure. To determine a new path, the process of probing, calculating and establishing a new path takes up time. Hence if a good alternate path survives the failure of the primary path, the disrupted messaging time can be avoided or reduced depending on the upper-level route maintenance mechanisms employed.

The focus here is to find the type of alternate paths that are geographical separated to a necessary extent from the primary path. In the previous work [62] a probability model of earthquake impact is incorporated into the path

selection algorithm, other than more abstract the proximity factor itself. Because an earthquake is only one cause of geographical correlated failures, many other human or nature causes result in geographical correlated failures, such as power grid failures, hurricanes, fires and malicious attacks. The general isolation is used here. It is difficult to model many of these faults and their dependencies with accurate models. However, proximity of networking elements is a key factor of damage magnitude in the above failure scenarios; hence in this work proximity is used to capture correlation between two geo-locations.

To provide uninterrupted messaging, it is proposed as an effective method to select an alternate path different from primary shortest path. Intuitively, when searching for an alternate path, this routing algorithm should consider geographical proximity between elements of the candidate paths and the primary path. Although there are many works researching the mechanisms and criteria of determining effective alternate paths, there has been little research on how to selecting geographically separated paths.

The following introduces the geographical proximity-aware routing algorithm.

5.3.1 Path Selection Criteria

A source broker uses a path selection algorithm to compute the best path from candidate overlay paths according to delay and other constraints. After choosing the best path, the source broker calculates the back-up path from the remaining candidates. To make the back-up path resilient to geographical failures, a novel back-up path selection algorithm is needed to take into account the factor of geographical correlation.

5.3.1.1 Correlation Factor and Proximity Factor

A geographical proximity measure is used as a backup path selection heuristic and represents a degree of geographical correlation between two

paths. Proximity needs to be modeled so that the quality of solutions from different algorithms can be evaluated. The measure can also be used by path selection algorithms as a potential heuristic.

The Proximity Factor *PF* between an alternative path P_n and its primary path P_m is defined in the following description:

Algorithm Find the Proximity Factor between two overlay paths $PF(P_n, P_m,)$ connecting overlay nodes *a, b,* with path P_n being the alternative path and P_m being the primary path

Require: $T_{distance}$:= distance threshold below which nodes are counted as being geographically close.

Require: $P_m := \{a, R_{m1}, R_{m2}, \ldots, R_{mr}, b\}$ physical nodes along the overlay path where R_{mi} is the i_{th} node along the path

Require: $P_n := \{a, R_{n1}, R_{n2}, \ldots, R_{nk}, b\}$ physical nodes along the overlay path where R_{ni} is the i_{th} node along the path

```
ProximityFactor ←0
for all R_mi ∈ P_m do
   for all R_nj ∈ P_n do
      if D(R_mi , R_nj)< T_distance then
         ProximityFactor=ProximityFactor+1
ProximityFactor= ProximityFactor/Hops(P_m)
return ProximityFactor
```

Figure 5-1 Calculating the Proximity Factor

If the distance between any pair of nodes along two overlay paths is below the prescribed distance threshold, then the proximity factor is increased by one. Before return, the proximity factor is normalized by the number of hops in the primary path P_m. Hence the smaller the Proximity Factor the better.

5.3.1.2 Other Requirements for Alternative Paths

The last section explains the rationale behind using geographical proximity as a metric to find resilient alternate path to cope with geographical correlated failures. Besides geographical diversity, i.e. the converse of proximity and minimum delay, several other practical requirements are often considered for the calculation of the alternate path. These requirements may be set by the end users, messaging service providers, or routing protocol itself. In the path calculation algorithms, these requirements are often expressed as constraints. These practical constraints are introduced in this section to describe a more complete picture for this kind of problem.

A. Delay or length of the path

Maximum path delay or length represents the maximum acceptable length of a path between source and target overlay node. It is defined as the sum of edge weights on the path. The path length may represent various physical properties, such as distance, cost, delay, or failure probability. It can be represented by an integer or a float value. If $w_i = 1$ is true for all edges, then the path length is the hop number from a source to a destination.

B. Number of hops on the path

Maximum acceptable number of hops of an overlay path represents the number of nodes that a message has to travel through to get to target node. It can be define either in the overlay level, or underlay level, or a combination of both. If a path contains k nodes, then its hop number is k-1. It is an integer value.

5.3.2 Computing Resilient Proximity-aware Multipath

In solving the shortest path problem, an edge *(i, j)* with a larger weight *w(i, j)* has less probability to be chosen in the calculated shortest path P_0. In the extreme case, if the weight of an edge *(i, j)* is increased to ∞, then this edge

will not be presented. Hence if the weights of edges are changed then the path computed, P_1 say, will be different. Increasing the edge weights is applied here in the process of selecting geographically separated alternative paths. The weights of edges near a primary path are increased by a large value ΔW, and hence they are less likely to be chosen in a recalculated alternative path.

The problem of finding geographical separated alternate path is described as follows. Given a primary path P_0, and pre-defined distance threshold r between the primary and alternative paths, this problem is to find an alternative path that is as geographical separated as possible with P_0, while also satisfying the other constraints such as delay and hop count. The pre-defined distance threshold r is an administrative parameter. Its purpose is to specify what geographical distance is desired for the separation between elements of the primary and alternate paths.

This proximity-aware algorithm searches for alternate paths by extending the edge weight-increment method and invoking Dijkstra's algorithm. The algorithm uses an edge weight-increment method as the heuristic to search for alternate paths that are geographical separated defined by distance threshold r. With this chosen heuristic, the weight of an edge w is augmented with a computed penalty ΔW, before shortest path algorithm is invoked, i.e.:

$$w := w + \Delta W$$

The algorithm starts with an initial large penalty $\Delta W = W_0$ on the edges which are within the distance threshold r to any of the router or overlay nodes on P_0. This is to ensure that the algorithm can find the alternate path satisfying the proximity constraint between the newly calculated the path P_1 and the primary path P_0, if such solution exists. If a P_1 does not exit, or it fails to satisfy other pre-defined constraints, then ΔW is decreased on corresponding edges and they form a changed graph. The alternate path search algorithm repeats on this newly changed graph.

The decrease of ΔW relaxes the heuristic for geographical separation. It is because if a perfect alternative path does not exist, the algorithm then will need to compromise for an alternate path with good quality in terms of

proximity and other constraints. In the new graph after the ΔW decrease, every edge within distance *r* from the primary path has a lighter weight, and thus has a higher probability to be chosen in the alternate path selection algorithm.

The detailed steps of this algorithm is summarised as below.

The initial value of ΔW is set to the sum of weights of all edges in the original undirected graph, i.e.:

$$\Delta W = W_0 = \sum w_{ij}.$$

Here, *i and j* are the ID of vertex V_i and V_j , and the w_{ij} correspond to the delays in the overlay edges along overlay path P_{ij} that connects V_i and V_j.

Algorithm Geographical proximity aware alternate path selection algorithm

Require: r := distance threshold,

P_0 := {a, R_{01}, R_{02},… , R_{0k}, b} primary path in the overlay from source a to target b where R_{0i} is the i_{th} overlay node along the P_0

doSearch:

for all E_{ij} *in the overlay with source Vi and destination Vj*

if $\exists V_a \in \{V_i, V_j\}$: $distance(V_a, P_0) < r$

then *adjust*(w_{ij});

P_1 = *searchShortestPath;*

if *satisfyConstraints*(P_1)

return P_1

else *doSearch*

Figure 5-2 Geographical proximity aware alternate path selection algorithm (GAP)

In the above algorithm, $distance(V_a, P_0)$ returns the shortest distance from node V_a to any of the router or overlay nodes on P_0.

Alternate path calculation constraints

In each loop, the adjustment function *adjust*(w_{ij}) changes the weight of edge (*i,j)* by adding a penalty value, if (*i,j)* is within the range threshold:

$$w_{ij} := w_{ij} + \Delta W$$

where the initial value of penalty in the first loop is

$$\Delta W = \Delta W_0$$

In a next loop, the penalty is calculated by decreasing its previous value by a coefficient α ($0 < \alpha < 1$).

$$\Delta W = \alpha\, \Delta W$$

Steps of the algorithm

Step 1. The shortest path P_0 is selected as the primary path.

Step 2. Every edge that is within r distance has its weight increased to $\Delta W = \Delta W_0$

Step 3. The weight on every edge within r is decreased to relax the proximity constraint. That is every edge the weight of which has been changed in the last step.

Variations of adjustment function:

The original adjustment function is tuned in two aspects, in order to reduce the total number of loops and improve the end results

When attempting to make the algorithm converge in fewer loops, the penalty is decreased logarithmically, instead of by a multiplicative coefficient α, i.e.

$$\Delta W_{n+1} = \log \Delta W_n$$

Here after the n^{th} iteration, the penalty value ΔW_{n+1} for the next iteration is decreased to $log\, \Delta W_n$. Then to increase the probability of selecting less proximate paths, the penalty added to a path *(i,j)* within distance threshold *r* is further adjusted according its distance d_{ij} from the primary path (the more distant paths are given smaller penalty):

$$\Delta W_{ij} = \frac{r - d_{ij}}{r}\, \Delta W_{ij}$$

5.3.3 Computational Complexity

Assume a graph with *N* nodes and *M* edges. The complexity of Dijkstra's

shortest path algorithm is $O(N^2 + M)$.

If the path found does not fit the other constraints defined, then another search loop will run with decreased penalty value to relax the geographical separation constraint. In each loop the penalty is decreased by multiplying by α until the penalty reaches the threshold T, lower than which no change is considered worthwhile. Let $\Delta W_o = \sum w_i$ be the initial penalty, where $\sum w_i$ is the sum of weights of all the edges in the searched graph.

With an initial penalty of ΔW_o, the algorithm stops the n^{th} loop, when the penalty falls below the prescribed threshold T, i.e. $\frac{1}{\alpha^n} \cdot \Delta W_0 \leq T$. Therefore in the worst case, the algorithm will run until $n = [\log_\alpha \frac{\Delta W_0}{T}]$. In each loop, the weight of a maximum of *M* edges needs to be adjusted by the penalty value. Hence the complexity is:

$$O\left([\log_\alpha \tfrac{\Delta W_0}{T}] \cdot \left((N^2 + M) + M\right)\right) = O\left(\log \tfrac{\Delta W_o}{T} \cdot (N^2 + M)\right)$$

With the implementation based on a min-priority queue implemented by a Fibonacci heap, the complexity is reduced to

$$O(\log \frac{\Delta W_o}{T} \cdot (N \cdot \log N + M))$$

When the searched graph is a sparse graph where $N^2 \gg M$ the previous complexity is approximately

$$O(\log \frac{\Delta W_o}{T} \cdot N^2)$$

5.4 *Evaluation of GAP and an Enhanced K-shortest Path Algorithm*

This part evaluates the performance of GAP against other alternative algorithms. The metrics evaluated are designed to represent the resilience of GAP and bench mark algorithms, and the effectiveness of them of achieving geographical separation between paths. Resilience is the ability of network to

maintain service in the face of failures. The geographical separation between paths is evaluated by introducing a proximity factor.

A failure is the simultaneous disconnection in both the primary path and alternative path. The number of failures is a measurement of effectiveness of an alternate path algorithm; however its values also depend on topology of the network graph in which the failures are generated. The proximity factor is a direct measurement of geographical separation. The algorithm for generating alternative paths using geographical distance is called GAP (Geo-aware Alternative Path based on Proximity) and two variants were tested. The difference between the variants is in how to adjust penalties in the path-finding process. A penalty function approach is used in the genetic algorithm to handle constraints. One variant uses a decrease the penalty by dividing the penalty by a prescribed number (GAP Div), and the other is based on the logarithm of the penalty from last iteration (GAP Log). For multipath routing and alternative path finding, the k-shortest path algorithms are widely used, with different enhancements to enforce constraints. Thus in comparisons, an enhanced k-shortest path algorithm is used to evaluate against the GAP algorithms. The simulation results will show a decrease in the proximity factor on average relative to the new alternative path algorithm. It demonstrates that the new algorithm is pushing the alternate path further apart from the primary path geographically.

5.4.1 A Benchmark Algorithm:

K-shortest path algorithms and disjoint path algorithms have been used as the foundation for different search algorithms. Disjoint path algorithms is a solution to another class of similar problems, which is to find fully disjoint paths between a source node and a destination node. However in the disjoint paths problem, the final path set calculated may not include the desired primary path, i.e. the shortest path or other pre-defined paths. Disjoint paths algorithm is not suitable a solution to the alternate path finding problem and

is not considered in the evaluation. Moreover, traditional disjoint path problem do not consider the geo-proximity of primary and alternate paths. Hence an enhanced K shortest path algorithm is implemented for the purpose of comparison with the proximity-aware alternate path finding algorithm. K shortest path algorithms calculate and rank a predefined number of shortest paths according to various preferences. After finding the K shortest paths between overlay node *s* and *t*, to find the desired alternate path, an additional path-filtering process is carried out to compare the candidate paths with desired constraints and finally the best one is chosen by predefined heuristic criteria.

For the problem of finding alternate path that satisfies some separation heuristic, a correlation indicator $CORR(P_m, P_n)$ is often used to model such separation heuristic. For example the correlation indicator $CORR(P_m, P_n)$ may be defined between the paths P_m, P_n, in terms the degree of overlapping edges or nodes, or by the availability value according to probability models.

Here an enhanced *K* shortest path algorithm is used for comparison with our geo-aware alternate path algorithm. After the K shortest paths are calculated between overlay node *s* and *t*. One path that is most distant spatially (in terms of proximity factor) to primary path is chosen an alternative path. The proximity factor represents the proportion of nodes along a path P_m that is within the distance *r* to another path P_n. It is calculated as below. Here the correlation indicator $\mathrm{CORR_m}$ is based on geographical proximity of two paths $\mathrm{P_a}$ and $\mathrm{P_b}$. Formally it equals to proximity factor *PF* between nodes and paths along these two overlay paths.

$$CORR(P_m, P_n) \quad = PF(P_m, P_n)$$

Algorithm overlay paths selection algorithm with extended *K* shortest path algorithm
Find the *K* shortest paths Select the alternate path p_k with smallest correlation factor with the shortest path $CORR(p_0, p_k) = PF(p_0, p_k)$

Figure 5-3 Enhanced K-shortest Path Algorithm

Algorithm overlay paths selection algorithm with extended maximally disjoint path algorithm
For find the shortest and maximally disjoint paths, If $CORR(p_0, p_k) = PF(p_0, p_k) < threshold$ then return p_k else relax the weight along p_0 and continue the search return the p_k with largest $PF(p_0, p_k)$

Figure 5-4 Calculating the Proximity Factor

5.4.2 Validation of GAP Implementation

In order to ensure the correctness of simulation, the simulator is built on a step-by-step manner. The function of each step is verified with a known input and the outputs are compared with the expected theoretical results, which are pre-calculated.

As the key algorithm, the correctness of Geo-aware Alternative Path algorithm is validated in a test graph of 8 nodes within a 10x10 plane with hand crafted tests. The test consists of a few cases. In each test case, GAP algorithm can be configured with a spatial distance threshold value R_{th} as the constraint of geographical separation and a maximal delay threshold D_{th} as temporal constraint. The GAP algorithm searches for an alternative path for the same primary path between node s and t in the graph. The solutions of GAP are then verified with known resulting alternative paths for each case. After the algorithms are validated, several crafted failure regions are employed to test the statistics collection function. A series of sample cases are shown as below. More details of this test can be found in the appendix.

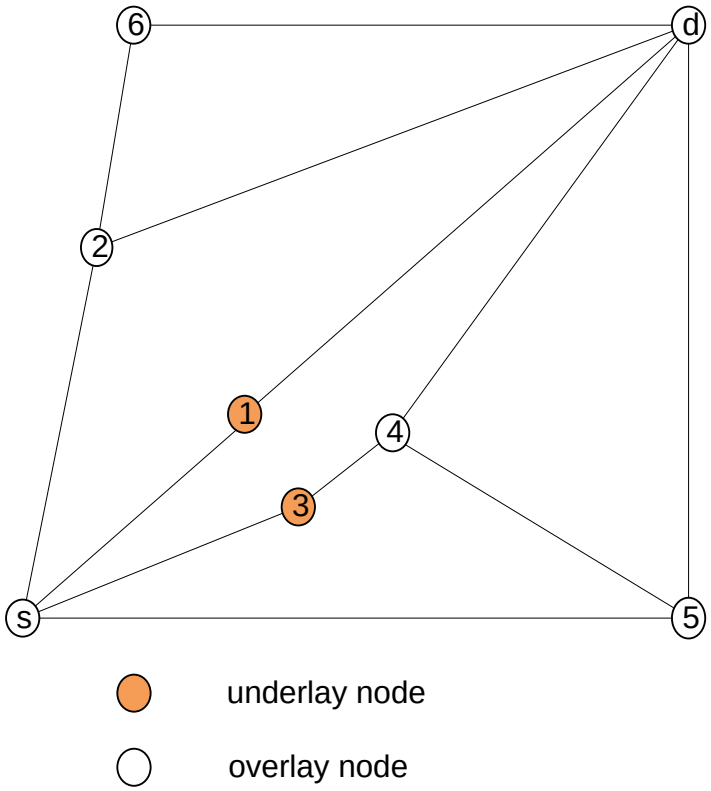

Figure 5-5 An example network to validate the GAP algorithms

The above graph is crafted to validate of the algorithm implementation against known results. In the graph node 1 and node 3 are only underlay nodes and overlay brokers are not directly attached to them. The other nodes are overlay nodes. Each overlay node represents an overlay broker and the nearest underlay node that it is attached to. The evaluation in next section uses different synthetic topologies that are generated with a well-known internet topology generator BRITE[143]. Here in the above graph the weights of edges equals to the geographical distance. The GAP algorithm with multiplicatively decreased penalties is configured to use the sum of weights of edges W_0 in the graph as the initial penalty, i.e. $\Delta W = W_0$. And then the penalty is decreased by 1/10 in subsequent loops. The primary path P_{pri} between source s and target t is *{s – 1 – t}*. The spatial distance threshold R_{th} = 5 and corresponding outputs of alternative paths with different delay constraint are as following:

- With spatial distance threshold R_{th} = 5, without delay constraint D_{th} =∞, the alternative path *{s – 5 – t}* is found in the initial loop.

- R_{th} = 5, and delay constraint Dth =15 which is the maximum length allowed of path is applied as a hard constraint. In the third loop, the alternative path {s – 2 – t} is found.

Three circular failure regions are generated, with different centre and radius: (5, 5) r=5, (3, 3) r=5, (6, 2) r=4. In the network the failures in the selected paths between s and t is shown in the following table:

centre, radius	{*s – 1 - t*},	{*s – 2 – t*}	{*s – 5 – t*}
(5, 5), r=5	0	0	1
(3, 3), r=5	0	0	0
(6, 2), r=4	0	1	0

Table 5-1 Statistics collection validation

Overall in the three cases, for primary path {s – 1 - t} the total number of failures is 3; and for multi-path pair {s – 2 – t} {s – 1 - t} and {s – 5 – t} {s – 1 - t} the total number of failures is 2.

Pseudo random number generator:

The well known Java implementation of Mersenne Twister is used, which has passed statistical tests related to the first two, and were efficient enough for the simulations in hand. The Mersenne Twister and tests for assuring the required statistical properties are described in [144]. It also had a large cycle length. This generator was adequate for the experiments in terms of efficiency and had a sufficient period to match the required duration of the experiments

5.4.3 Evaluation of GAP against EKSP

For testing a routing algorithm or overlay routing protocol, the two usual ways are to emulate the application in test bed systems such as PlanetLab, or to incorporate an underlay network topology in simulation. The correlated failures are rare and difficult to capture in real world test bed. Hence we

conduct evaluation through simulation. To evaluate the GAP and EKSP algorithms, the underlay graph in the test is generated using the BRITE internet topology generation tool [143]. The graph represents the underlay network. It has 1,000 underlay nodes within a 10,000 by 10,000 Euclidean plane. For overlay network, 100 overlay brokers are randomly generated, the coordinates of which are drawn from uniform distribution. Then they are attached to the nearest underlay nodes. 200 overlay source node and target node pairs are randomly selected. Between every pair of source and target node, a primary path is established by Dijkstra algorithm, and then the alternative paths are found with the evaluated algorithms.

After the algorithms found the alternate path solutions, 50 circular failure regions are randomly generated with uniform distribution. Each group of failure regions have a predefined different failure radius of R_f, to explore the different impacts of the failure radius parameter. The primary and alternate path pairs are evaluated against these failures. The tests are repeated for different values of R_f.

In the evaluation, the quality of alternate proximity-aware paths and its effectiveness as an enhanced protection mechanism against geographical correlated failures are the essential aspects of concern. Some parameters are in need, as metrics to describe the quality and effectiveness of alternate path finding algorithm. The metrics compared here is the *Proximity Factor* and the resiliency between two algorithms.

Here the two aspects are measured and compared as the followings. First, the degree of proximity between the primary path and alternate path found by different methods are compared to evaluate the quality of alternate path-finding algorithms. The degree of proximity between two paths is described with the proximity factor of the chosen path sets.

Second, for mission critical applications, the resiliency of overlay routing methods is essential in the face of network failures. Here the uninterrupted messaging is the key concern and used to describe the resilience, (e.g. at least

one path between two the source and destination is working). Here if all paths between source and destination node are disconnected, messaging will be interrupted until a path recalculation find another solution and establishing another path.

5.4.4 Results from the Evaluations

After above tests, two sets of statistics are compared in the results. The first is the *Proximity Factor,* which measures the degree of spatial separation of alternative path algorithms. The second is the number of failures caused by a number of random generated failures. This evaluates the resilience of alternative path algorithms against geographical correlated failures. Both of these are shown in the results bellow:

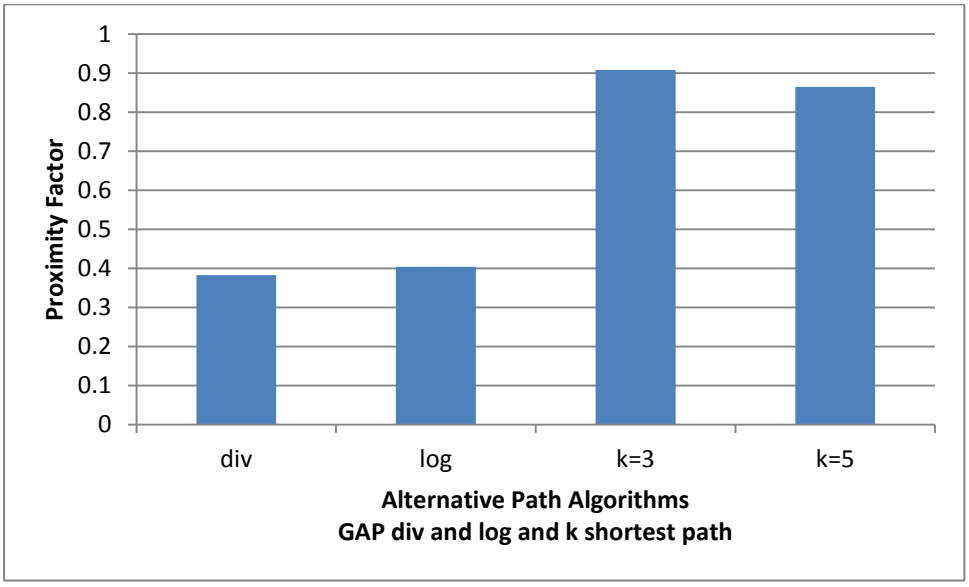

Figure 5-6 Proximity Factor between Alternative Paths and Primary Paths (A low PF indicates greater separation)

In comparison with k-shortest path algorithm, the sample results new alternative path algorithm shows an over 0.4 decrease of proximity factor in average. It demonstrates that the new algorithm is pushing the alternate path further apart from primary path geographically. The more geographical separation further results in less overall number of failures as shown below.

	Number of Failures				
Radius	primary path	gap_div	gap_log	k=3	k=5
3000	9495	8528	8569	9117	9048
2000	6187	4672	4708	5739	5645
1000	2291	1313	1337	1922	1841
500	731	327	333	532	500
200	109	45	47	71	67

Table 5-2 Failure numbers under geographical correlated failures

	Improvement over primary path only			
radius	gap_div	gap_log	k=3	k=5
3000	0.101843	0.097525	0.03981	0.047077
2000	0.244868	0.23905	0.07241	0.087603
1000	0.426888	0.416412	0.161065	0.196421
500	0.552668	0.54446	0.27223	0.316005
200	0.587156	0.568807	0.348624	0.385321

Table 5-3 Proportionate improvements of uninterrupted connections over primary path only

The failures are presented as the proportionate reduction in failures relative to the benchmark EKSP algorithm. In the figure above proportional reduction is defined as:

$$(\text{Failures(GAP or EKSP)} - \text{Failures (Primary Path)})/\text{Failures (Primary Path)}$$

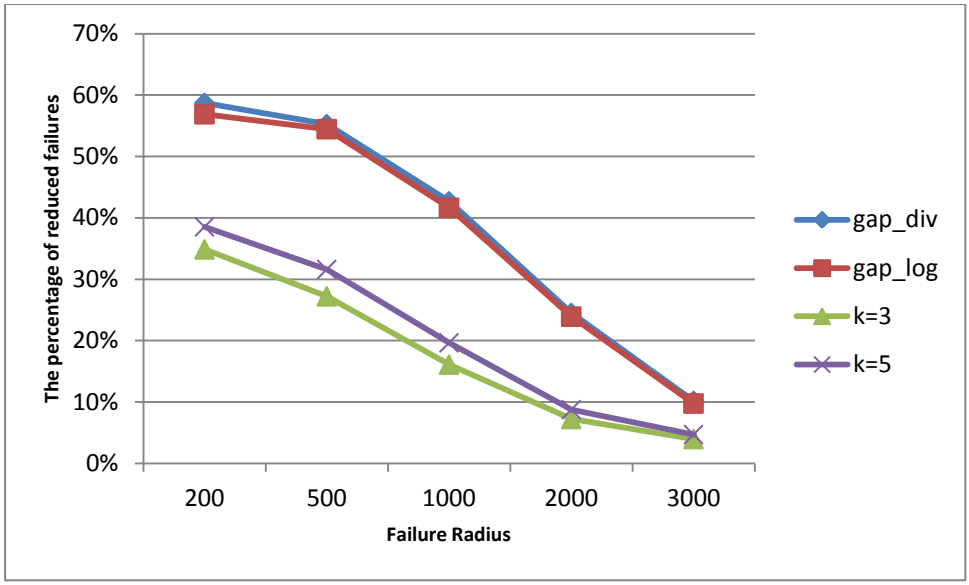

Figure 5-7 The percentage of reduced failures of GAP and EKSP over only using a primary path

In the above results, under the scenarios of different failure radiuses, multi-path routing basing on GAP and EKSP are compared with a simple shortest overlay path. Both GAP and EKSP based multi-path routing have improvements over using only a shortest primary path. These improvements are more significant when failure radiuses are relatively smaller. The improvements decrease as the failure radius becomes larger.

The proximity constraint threshold P_{th} is set to 2000 in all cases. For a smallest failure radius *r=200*, the improvements are over 58% and 56% for GAP approaches and over 34% and 38% for EKSP. When the failure radius *r* is comparable with $P_{th,}$ (*r=1000*), the GAP approaches maintain 42.7% and 41.6% improvements, while improvements drop to a less significant 16.1% and 19.6% for EKSP approaches. As the failure radius *r* rises above proximity constraint threshold P_{th} , the resilience of the two variants of multi-path routing significantly drop. When the failure radius *r* reaches *3000*, the improvements are 10.2% and 9.75% for GAP algorithms and 3.98% and 4.7% for KSP-based algorithms.

Overall in these tests, both of the geo-aware algorithms improve the resilience of shortest path based routing. Also GAP algorithms perform better than the

EKSP-based ones in all cases.

5.5 *Summary*

In this chapter, we presented the design of federated overlay level routing components that will provide resilience and QoS-awareness. The specific focus is to mitigate the damage geographical large scale failures which may impose significant threats to the federated overlay of RQMOM. A novel overlay routing mechanism is designed to improve the resilience of messaging in such large geographical scale failures. As a practical improvement to single path overlay routing, this novel algorithm is developed to search for geographical separated alternative path, with any given primary path. It algorithm is evaluated against an enhanced k-shortest path algorithm on a synthetic internet topology generated with BRITE [143], with random generated geographical correlated failures. The graph in the test is generated with BRITE uses the flat topology generation option. This setting marks the geolocation of each node. The graph represents the underlay network. It has 1,000 underlay nodes within a 10,000 by 10,000 Euclidean plane. The average node degree in the graph is eight, which is close to current internet. The testing network is chosen because it provides the geographical locations to be effective in evaluating the algorithms and it also maintains the average node degree to be representative to realistic networks. In the evaluation both the degree of separation of multi-paths and their resilience against geographical correlated failures are compared in the results. In comparison with the enhanced k-shortest path algorithm the novel algorithm shows improvement in terms of the degree of spatial separation measured by a proximity factor with the given primary path. As a result the novel algorithm also increases the percentage of uninterrupted messaging connections in the face of the generated geographical correlated failures.

Chapter 6 Conclusion and Future Work

Conclusions

This thesis introduces novel mechanisms to provide resilient messaging in a hierarchical MOM overlay structure. The following contributions of the thesis have been explored.

In the local domain, the busty correlated surge of workload is a problem that causes performance degradation or system fault in messaging brokers. To address this, a proactive risk-aware workload allocation and mirroring mechanism is developed as an enhancement to current reactive load balancing mechanisms. The problem of risk-aware workload allocation has been formulated. The allocation algorithm minimizes the risk in a quantified way and mitigates correlated bursty workload to different brokers. The mirroring solution adapts the system to tolerate broker faults and faults and failures in the application end points in local domains.

In the risk-aware workload allocation problem, a GA-based approximation algorithm is developed. This is evaluated against the widely used Round Robin allocation. The test shows the GA-based approximation algorithm shows consistent improvements over a widely used random Round Robin allocation. Especially in the case when a low quality Round Robin allocation solution occurs, the GA-based approximation algorithm shows further improvements, reducing the risk of overloading brokers.

In the overlay domain, a novel overlay routing mechanism is researched to improve the resilience of messaging in the face of large geographical scale failures. As a practical improvement to single path overlay routing, a novel algorithm is developed to search for geographical separated alternative path, with any given primary path. This algorithm is evaluated against an enhanced k-shortest path algorithm on a synthetic internet topology generated with BRITE, with random generated geographical correlated

failures. In comparison with the enhanced k-shortest path algorithm the novel algorithm shows improvement in terms of the degree of spatial separation with the given primary path. Hence the novel algorithm is shown to increase the percentage of uninterrupted messaging connections in the face of the generated geographical correlated failures.

Future Work

Some current research results in this thesis can be extended into further publications. There are also a few research directions worth further exploration based on the works presented.

In the local domain, the workload allocation mechanism could be extended with context-aware techniques such as Case-based Reasoning and then can be automatically applied. The fine-grained allocation solutions and adjustment actions made can be stored in case bases, so that the MOM system can quickly find the solution and make automated reaction according to changing situations.

In federated overlay domain, the GAP algorithm can be evaluated further in real world network topologies and failure scenarios. Through the evaluation, the practical performance of GAP can be further studied, and also the configuration of GAP algorithm analysed. For example, the effective distance threshold value in KM, can be determined to mitigate real world geographical correlated failure problems based on category. The assumption that the overlay topology is known in advance is sound in many application scenarios because the federation members only change on a coarse timescale (e.g., once in a few weeks). However, when brokers do frequently join and leave, a dynamic topology design and maintenance scheme is needed to adapt the overlay topology.

Appendix

6.1 Author's Publications

1. Jinfu Wang, John Bigham, Bob Chew, Jiayi Wu "**Enhance Resilience and QoS Awareness in Message Oriented Middleware for Mission Critical Applications**", 2nd Symposium on Middleware and Network Applications, April 11-13, 2011, Las Vegas, Nevada, USA
2. Jinfu Wang, John Bigham, Beatriz Murciano "**Towards a Resilient Message Oriented Middleware for Mission Critical Applications**", The Second International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE 2010), November 21-26, 2010 - Lisbon, Portugal
3. Habtamu Abie, Reijo Savola, Jinfu Wang, and Domenico Rotondi (2010): "**Advances in Adaptive Secure Message Oriented Middleware for Distributed Business Critical Systems"**, 8th International Conference of Numerical Analysis and Applied Mathematics**,** (ICNAAM 2010), 19-25 September 2010, Greece
4. Jinfu Wang, Peng Jiang, John Bigham, Bob Chew, Beatriz Murciano, Milan Novkovic, Ilesh Dattani, **"Adding Resilience to Message Oriented Middleware"**, in Proc. Serene 2010 ACM, 13-16 April, 2010 London, UK
5. Jinfu Wang, John Bigham, "**Anomaly detection in the case of message oriented middlewar**e". 1st International Workshop on Middleware Security (MidSec 2008), December 2, 2008 Leuven, Belgium.

   The other publications

6. Jinfu WANG, John Bigham, "**Enforcing Application Security - Fixing Vulnerabilities with Aspect Oriented Programming",** Joint 2008 European Safety and Reliability Association and 17th European Society

for Risk Analysis Annual conference (ESREL 2008)

7. Bob Chew, Jinfu Wang, Athen Ma, John Bigham, "**Anomalous Usage in Web Applications**", 2008 Networking and Electronic Commerce Research Conference (NAEC 2008), Sep 2008, Lake Garda, Italy
8. Usman Naeem, John Bigham, Jinfu Wang: **"Recognising Activities of Daily Life Using Hierarchical Plans"**. EuroSSC 2007: 175-189

6.2 Description of GAP Sample Test Data Log

A sample data log is provided. In the data log, the geo-aware alternate path (GAP) algorithm using division penalty (div=10) decrease and k-shortest path (k=3) algorithm are used to search for alternate path against a primary path which is determined using Dijkstra shortest path algorithm. The graph in the test is generated using the BRITE internet topology generation tool. It has 1,000 underlay nodes within a 10,000 by 10,000 Euclidean plane. 100 overlay brokers are randomly generated and attached to the nearest underlay nodes, where each (x, y) coordinate in the Euclidean plane is sampled from the uniform distribution. After the algorithms found the alternate path solutions, 100 circular failure regions are randomly generated each with a radius of 2000. An extracted sample data log is shown as below. Note some rows in the middle are omitted only for saving the space.

Header of the log file:

Date:		Wed.2014.07.30_04.05.59
10k_1k_m8.brite		
DIV.10.0	K3	
C		2000

Body of the log file:

ID_{path}	S	T	Con_{pri}	Con_{gap}	Con_{ksp}	PF_{gap}	PF_{ksp}	Reg
1	100	548	67	82	70	0.285714	1	100
2	100	631	64	73	73	0.333333	0.666667	100
3	100	695	76	80	76	0.25	1	100
4	106	278	72	79	75	0.333333	0.666667	100
5	106	46	69	81	69	0.25	1	100

6	106	695	76	79	76	0.285714	1	100
7	171	631	62	75	65	0.285714	0.875	100
8	171	706	69	85	79	0.333333	0.857143	100
9	171	877	73	79	73	0.333333	0.857143	100
10	188	106	59	75	65	0.25	1	100
11	188	407	76	76	76	1	1	100
12	188	73	57	80	61	0.4	1	100
13	188	857	76	76	76	1	1	100
14	197	489	69	80	69	0.133333	0.833333	100
15	197	714	78	82	80	0.333333	0.6	100
……								
189	944	810	68	72	68	0.333333	1	100
190	944	892	60	67	72	0.25	1	100
191	956	137	73	77	79	0.285714	0.75	100
192	956	217	66	83	69	0.4	1	100
193	956	487	82	82	82	0.285714	1	100
194	956	730	81	82	81	0.2	1	100
195	957	337	67	81	68	0.285714	0.833333	100
196	971	341	73	82	73	0.285714	1	100
197	979	249	61	75	73	0.333333	0.428571	100
198	979	489	66	76	69	0.285714	1	100
199	999	310	80	80	80	0.222222	1	100
200	999	607	80	80	80	0.166667	1	100

Tail of the log file:

total_Alive	13813	15328	14261	
percent_Alive	0.69065	0.7664	0.71305	

Table A-0-1 A sample log file from the test of overlay routing test

In the above log file the contents of the header, body and tail are further explained:

The information from header of the above log file is as bellow:

Date and Time when the test is done	
The topology file name used	
GAP configuration (DIV or LOG)	EKSP configuration (K=3 or …)
Shape of failure region (e.g. c for circular)	Failure radius (numeric value)

In the body of the above log file, each column name represents a parameter or a counter, and the meanings of the column names are:

ID_{path}	ID of the path (1~100)
S	underlay Node ID of source
T	underlay Node ID of destination

Con_{pri}	counter of primary path that remain connected during generated failures
Con_{gap}	counter of GAP paths that remain connected during generated failures
Con_{ksp}	counter of EKSP paths that remain connected during generated failures
PF_{gap}	proximity factor of the multipath selected by GAP algorithm
PF_{ksp}	proximity factor of the multipath selected by EKSP algorithm
Reg	number of failure regions generated

The two rows of outcome values shown in the tail of the log file are total counter of paths that remain connected and percent of paths that remain connected during the tests. The columns represent the overlay routing method used (primary path only, GAP or EKSP):

Total counter of paths that remain connected	Primary path only	GAP	EKSP
Total percent of paths that remain connected	Primary path only	GAP	EKSP

Table A-0-2 The description of overlay routing sample test log file

6.3 GAP Validation Testing Case

The detailed coordinate of nodes in the network to test routing algorithms of section 5.4.2 are as following:

node	x,y
s	1,1
d	9,9
1	3,3
2	2,7
3	3,2
4	4,3
5	9,1
6	2.5,9

$W_0 = \sum w_{ij} = 66.083829.$

The GAP algorithm and the explanation of above values are illustrated in section 5.3.

References

1. Reumann, J., *Pub/Sub at Google*, O. Lecture at EuroSys & CANOE Summer School, Norway, Editor. 2009.
2. Lee, E.A., *Cyber physical systems: Design challenges, In Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing. IEEE Computer Society*.
3. Venkatasubramanian, K.K., *Security solutions for cyber-physical systems*. 2009, Arizona State University.
4. Albano, M., et al., *Message-oriented middleware for smart grids.* Computer Standards & Interfaces, 2015. 38: p. 133-143.
5. Zhou, L. and J.J. Rodrigues, *Service-oriented middleware for smart grid: Principle, infrastructure, and application.* Communications Magazine, IEEE, 2013. 51(1): p. 84-89.
6. Peng, Z., Z. Jingling, and L. Qing. *Message oriented middleware data processing model in Internet of things*. in *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*. 2012: IEEE.
7. *SOS: System of systems*. Available from: http://www.sosece.org/
8. *Tibco Rendevzous*. Available from: http://www.tibco.com.
9. Liu, H., et al., *Client behavior and feed characteristics of RSS, a publish-subscribe system for web micronews*, in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. 2005, USENIX Association: Berkeley, CA.
10. Petrovic, M., H. Liu, and H.A. Jacobsen, *G-ToPSS - fast filtering of graph-based metadata*, in *In the 14th International World Wide Web Conference*. 2005.
11. Papazoglou, M.P. and W.-J. Heuvel, *Service oriented architectures: approaches, technologies and research issues.* The VLDB Journal, 2007. 16(3): p. 389-415.
12. Li, G., V. Muthusamy, and H.-A. Jacobsen, *A distributed service-oriented architecture for business process execution.* ACM Trans. Web, 2010. 4(1): p. 1-33.
13. Grefen, P., et al., *2000).Crossflow: cross-organizational workflow management in dynamic virtual enterprises.* Int. J. Comput. Syst. Sci. Eng, (15): p. 227--290.
14. Comitz, P., et al. *The joint NEO Spiral 1 program: Lessons learned operational concepts and technical framework*. in *Integrated Communications, Navigation and Surveillance Conference, 2008. ICNS 2008*. 2008.
15. Dolin, R.A. *Deploying the "Internet of things"*. in *Applications and the Internet, 2006. SAINT 2006. International Symposium on*. 2006.
16. Gershenfeld, N., R. Krikorian, and Cohen, *D.: The Internet of Things. Scientific American*. 2004.
17. Choi, J. and C. Yoo, *Connect with Things through Instant Messaging*, in *The Internet of Things*. 2008. p. 276-288.
18. Ananda, A.L., B.H. Tay, and E.K. Koh, *A survey of asynchronous remote procedure calls.* SIGOPS Oper. Syst. Rev., 1992. 26(2): p. 92-109.
19. Birrell, A.D. and B.J. Nelson, *Implementing remote procedure calls.* ACM

Trans. Comput. Syst., 1984. 2(1): p. 39-59.
20. Sun. *Java Remote Method Invocation Specification*. 2000; Available from: http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html.
21. OMG. *CORBA Event Service Specification*. 2001.
22. OMG. *The Common Object Request Broker: Core Specification.* . 2002.
23. Banavar, G., et al., *A Case for Message Oriented Middleware*, in *Proceedings of the 13th International Symposium on Distributed Computing*. 1999, Springer-Verlag.
24. BLAKELEY, B., HARRIS, H., LEWIS, J., *Messaging and Queuing Using the MQI*. 1995: McGraw-Hill, New York, NY.
25. Setty, V., et al. *PolderCast: fast, robust, and scalable architecture for P2P topic-based pub/sub*. in *Proceedings of the 13th International Middleware Conference*. 2012: Springer-Verlag New York, Inc.
26. Rahimian, F., et al. *Vitis: A gossip-based hybrid overlay for internet-scale publish/subscribe enabling rendezvous routing in unstructured overlay networks*. in *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*. 2011: IEEE.
27. Yingwu, Z., *Ferry: A P2P-Based Architecture for Content-Based Publish/Subscribe Services.* IEEE Transactions on Parallel and Distributed Systems, 2007. 18: p. 672-685.
28. Gupta, A., et al., *Meghdoot: content-based publish/subscribe over p2p networks*, in *In Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. 2004, Springer-Verlag. p. 254--273.
29. Kramer, J., *Advanced message queuing protocol (AMQP).* Linux Journal, 2009. 2009(187): p. 3.
30. *DDS: Data distribution service for real-time systems.* Available from: http://www.omg.org/technology/documents/formal/data_distribution.htm.
31. Hakiri, A., et al., *Supporting end-to-end quality of service properties in OMG data distribution service publish/subscribe middleware over wide area networks.* Journal of Systems and Software, 2013. 86(10): p. 2574-2593.
32. *JMS: Java messaging service*. Available from: http://java.sun.com/products/jms/.
33. Dias, D.M., et al., *A scalable and highly available web server*, in *Proceedings of the 41st IEEE International Computer Conference*. 1996, IEEE Computer Society.
34. Shirriff, K., *Building distributed process management on an object-oriented framework*, in *Proceedings of the annual conference on USENIX Annual Technical Conference*. 1997, USENIX Association: Anaheim, California.
35. Aleksy, M., A. Korthaus, and M. Schader, *Design and Implementation of a Flexible Load Balancing Service for CORBA-based Applications*, in *In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 欽?1), Las Vegas*. 2001, IEEE. p. 2140--2144.
36. Barth, T., et al., *Load Distribution in a CORBA Environment*, in *Proceedings of the International Symposium on Distributed Objects and Applications*. 1999, IEEE Computer Society.
37. Ho, K.S. and H.V. Leong, *An Extended CORBA Event Service with Support for Load Balancing and Fault-Tolerance*, in *Proceedings of the International Symposium on Distributed Objects and Applications*. 2000, IEEE Computer Society.

38. Lindermeier, M. *Load management for distributed object-oriented environments*. in *Distributed Objects and Applications, 2000. Proceedings. DOA '00. International Symposium on*. 2000.
39. Hui, Z., et al. *Optimal Load Balancing in Publish/Subscribe Broker Networks Using Active Workload Management*. in *Communications, 2008. ICC '08. IEEE International Conference on*. 2008.
40. Berman, F. and R. Wolski, *Scheduling From the Perspective of the Application*, in *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing*. 1996, IEEE Computer Society.
41. Othman, O. and D.C. Schmidt, *Issues in the Design of Adaptive Middleware Load Balancing.* SIGPLAN Not., 2001. 36(8): p. 205-213.
42. Cheung, A.K.Y. and H.-A. Jacobsen, *Dynamic load balancing in distributed content-based publish/subscribe*, in *Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware*. 2006, Springer-Verlag New York, Inc.: Melbourne, Australia.
43. Chen, Y. and K. Schwan, *Opportunistic overlays: efficient content delivery in mobile ad hoc networks*, in *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*. 2005, Springer-Verlag New York, Inc.: Grenoble, France.
44. Casalicchio, E. and F. Morabito. *Distributed subscriptions clustering with limited knowledge sharing for content-based publish/subscribe systems*. in *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*. 2007.
45. Riabov, A., et al. *Clustering algorithms for content-based publication-subscription systems*. in *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*. 2002.
46. Wong, T., R. Katz, and S. McCanne. *An evaluation of preference clustering in large-scale multicast applications*. in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. 2000.
47. Li, M., et al. *A Scalable and Elastic Publish/Subscribe Service*. in *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*. 2011.
48. Ma, X., et al., *Scalable and elastic event matching for attribute-based publish/subscribe systems.* Future Generation Computer Systems, 2014. 36: p. 102-119.
49. Chand, R. and P. Felber. *XNET: a reliable content-based publish/subscribe system*. in *Reliable Distributed Systems, 2004. Proceedings of the 23rd IEEE International Symposium on*. 2004.
50. Esposito, C., D. Cotroneo, and A. Gokhale, *Reliable publish/subscribe middleware for time-sensitive internet-scale applications*, in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. 2009, ACM: Nashville, Tennessee.
51. Kazemzadeh, R.S. and H.A. Jacobsen. *Reliable and Highly Available Distributed Publish/Subscribe Service*. in *Reliable Distributed Systems, 2009. SRDS '09. 28th IEEE International Symposium on*. 2009.
52. Jafarpour, H., S. Mehrotra, and N. Venkatasubramanian. *A Fast and Robust Content-based Publish/Subscribe Architecture*. in *Network Computing and Applications, 2008. NCA '08. Seventh IEEE International Symposium on*. 2008.
53. S.D, M.K. and U. Bellur, *An underlay aware, adaptive overlay for event broker networks*, in *Proceedings of the 5th workshop on Adaptive and*

reflective middleware (ARM '06). 2006, ACM: Melbourne, Australia.
54. Peter, R.P. *Hermes: A Distributed Event-Based Middleware Architecture*. 2002.
55. Cao, F. and J.P. Singh, *MEDYM: match-early with dynamic multicast for content-based publish-subscribe networks*, in *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*. 2005, Springer-Verlag New York, Inc.: Grenoble, France.
56. Young Yoon, V.M.a.H.-A.J., *Foundations for Highly Available Content-based Publish/Subscribe Overlays*. 2011.
57. Bhola, S., et al., *Exactly-once delivery in a content-based publish-subscribe system*, in *In DSN*. 2002.
58. Andersen, D.G., et al., *Resilient Overlay Networks*, in *in Proc. of ACM SOSP*. 2001.
59. Opos, J.M., et al. *A Performance Analysis of Indirect Routing*. in *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*. 2007.
60. Sung-Ju, L., et al. *Bandwidth-Aware Routing in Overlay Networks*. in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. 2008.
61. Weidong, C., I. Stoica, and R.H. Katz. *Backup path allocation based on a correlated link failure probability model in overlay networks*. in *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*. 2002.
62. Meersman, R., et al., *Overlay Routing under Geographically Correlated Failures in Distributed Event-Based Systems*, in *On the Move to Meaningful Internet Systems, OTM 2010*. 2010, Springer Berlin / Heidelberg. p. 764-784.
63. Fraiwan, M. and G. Manimaran. *Localization of IP links faults using overlay measurements*. in *Communications, 2008. ICC'08. IEEE International Conference on*. 2008: IEEE.
64. Suto, K., et al. *An overlay network construction technique for minimizing the impact of physical network disruption in cloud storage systems*. in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. 2014: IEEE.
65. Beitollahi, H. and G. Deconinck. *An overlay protection layer against denial-of-service attacks*. in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*. 2008: IEEE.
66. Lua, E.K., et al., *A survey and comparison of peer-to-peer overlay network schemes*. Communications Surveys & Tutorials, IEEE, 2005. 7(2): p. 72-93.
67. Chen, C., H.-A. Jacobsen, and R. Vitenberg, *Algorithms based on divide and conquer for topic-based publish/subscribe overlay design.* 2015.
68. Qiu, L., et al., *On selfish routing in internet-like environments*. IEEE/ACM Trans. Netw., 2006. 14(4): p. 725-738.
69. Roughgarden, T., \#201, and v. Tardos, *How bad is selfish routing?* J. ACM, 2002. 49(2): p. 236-259.
70. Seshadri, M. and Y.H. Katz, *Dynamics of simultaneous overlay network routing, tech. rep*. 2003.
71. DiPalantino, D. and R. Johari. *Traffic Engineering vs. Content Distribution: A Game Theoretic Perspective*. in *INFOCOM 2009, IEEE*. 2009.
72. Wardrop, J., *Some theoretical aspects of road traffic research.* Proceedings of the Institution of Civil Engineers, Part II, 1952. 1(36): p. 352-362.
73. Haiyong, X., et al. *On self adaptive routing in dynamic environments - an evaluation and design using a simple, probabilistic scheme*. in *Network*

Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on. 2004.
74. Calzarossa, M.C., et al., *Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges*, in *Performance Tools and Applications to Networked Systems*. 2004, Springer Berlin Heidelberg. p. 209-234.
75. Andersen, D., A. Snoeren, and H. Balakrishnan, *Best-Path vs. Multi-Path Overlay Routing*, in *in Proc of IMC*. 2003.
76. Cheng, W.C., et al. *Large-scale data collection: a coordinated approach*. in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. 2003.
77. Ganguly, S., et al. *Fast replication in content distribution overlays*. in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. 2005.
78. Shu, T., et al. *Exploring the performance benefits of end-to-end path switching*. in *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*. 2004.
79. Wang, B., et al., *Application-layer multipath data transfer via TCP: Schemes and performance tradeoffs.* Performance Evaluation, 2007. 64(9-12): p. 965-977.
80. Leung, J.Y.T., *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. 2004: Chapman & Hall/CRC.
81. Bui, V., et al., *A Markovian Approach to Multipath Data Transfer in Overlay Networks.* Parallel and Distributed Systems, IEEE Transactions on, 2010. 21(10): p. 1398-1411.
82. Cetinkaya, C. and E.W. Knightly. *Opportunistic traffic scheduling over multiple network paths*. in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. 2004.
83. Liu, X., E.K.P. Chong, and N.B. Shroff, *Opportunistic transmission scheduling with resource-sharing constraints in wireless networks.* Selected Areas in Communications, IEEE Journal on, 2001. 19(10): p. 2053-2064.
84. Puterman, M., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1994: Wiley-Interscience.
85. Bui, V., et al., *A markovian approach to multipath data transfer in overlay networks.* Parallel and Distributed Systems, IEEE Transactions on, 2010. 21(10): p. 1398-1411.
86. Wu, J., et al., *A novel scheduling approach to concurrent multipath transmission of high definition video in overlay networks.* Journal of network and computer applications, 2014. 44: p. 17-29.
87. Di Caro, G. and M. Dorigo, *AntNet: Distributed stigmergetic control for communications networks.* Journal of Artificial Intelligence Research, 1998: p. 317-365.
88. Leibnitz, K., N. Wakamiya, and M. Murata, *Biologically inspired self-adaptive multi-path routing in overlay networks.* Commun. ACM, 2006. 49(3): p. 62-67.
89. Rewaskar, S. and J. Kaur, *Testing the scalability of overlay routing infrastructures*, in *Passive and Active Network Measurement*. 2004, Springer. p. 33-42.
90. Nakao, A., L. Peterson, and A. Bavier, *Scalable routing overlay networks.* ACM SIGOPS Operating Systems Review, 2006. 40(1): p. 49-61.
91. Young, A., et al. *Overlay mesh construction using interleaved spanning trees*. in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE*

Computer and Communications Societies. 2004: IEEE.
92. Brockmeyer, J.S.a.M., *QoSMap: Achieving Quality and Resilience through Overlay Construction*. 2010.
93. Shamsi, J. and M. Brockmeyer. *Efficient and dependable overlay networks*. in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*. 2008: IEEE.
94. Baud, L., N. Pham, and P. Bellot. *Robust overlay network with Self-Adaptive topology: Protocol description*. in *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*. 2008: IEEE.
95. Baud, L. *Robust overlay network with self-adaptive topology: The reliable file storage layer*. in *Computing and Communication Technologies, 2009. RIVF'09. International Conference on*. 2009: IEEE.
96. Pietzuch, P. and S. Bhola, *Congestion control in a reliable scalable message-oriented middleware*, in *In Proceedings of ACM/IFIP/USENIX International Middleware Conference (Middleware*. 2003.
97. (INRIA), S.B., et al., *Selfware Self-management of JMS-based application*. 2008.
98. AlZain, M., et al. *Cloud computing security: from single to multi-clouds*. in *System Science (HICSS), 2012 45th Hawaii International Conference on*. 2012: IEEE.
99. Bessani, A., et al., *DepSky: dependable and secure storage in a cloud-of-clouds*. ACM Transactions on Storage (TOS), 2013. 9(4): p. 12.
100. Veeravalli, B. and M. Parashar, *Guest editors' introduction: Special issue on cloud of clouds.* IEEE Transactions on Computers, 2014(1): p. 1-2.
101. Jayalath, C., J. Stephen, and P. Eugster, *Universal Cross-Cloud Communication.* Cloud Computing, IEEE Transactions on, 2014. 2(2): p. 103-116.
102. Esposito, C., et al., *Interconnecting federated clouds by using publish-subscribe service.* Cluster Computing, 2013. 16(4): p. 887-903.
103. Abie, H., et al., *Advances in Adaptive Secure Message-Oriented Middleware for Distributed Business Critical Systems*, in *8th International Conference of Numerical Analysis and Applied Mathematics*. 2010: Rhodes, Greece.
104. Wang, J., et al., *Adding Resilience to Message Oriented Middleware*, in *Proceedings of 2nd International Workshop on Software Engenering for Resilient Systems*. 2010, ACM London, UK. p. 89-94.
105. Abie, H., R.M. Savola, and I. Dattani. *Robust, Secure, Self-Adaptive and Resilient Messaging Middleware for Business Critical Systems*. in *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*. 2009.
106. Abie, H., et al. *GEMOM - Significant and Measurable Progress beyond the State of the Art*. in *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*. 2008.
107. Wang, J. and J. Bigham, *Anomaly detection in the case of message oriented middleware*, in *Proceedings of the 2008 workshop on Middleware security*. 2008, ACM: Leuven, Belgium.
108. G. Marsh, A.P.S., S. Potluri, and D.K. Panda, , *Scaling Advanced Message Queuing Protocol (AMQP) Architecture with Broker Federation and InfiniBand*. 2009
109. Mi, N., et al., *Burstiness in multi-tier applications: symptoms, causes, and new*

models, in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. 2008, Springer-Verlag New York, Inc.: Leuven, Belgium.
110. Yoon, Y., V. Muthusamy, and H.-A. Jacobsen, *On-demand Replication for Failover in Content-based Publish/Subscribe Overlays*, in *Middleware Systems Reserch Group Technical Report*. 2009, University of Toronto.
111. Henjes, R., *Performance Evaluation of Publish/Subscribe Middleware Architectures*. 2010.
112. Marsh, G., et al., *Design and Evaluation of Benchmarks for Financial Applications using Advanced Message Queuing Protocol (AMQP) over InfiniBand*. 2008.
113. Gregory Marsh, A.P.S., Sreeram Potluri, and Dhabaleswar K. Panda, *Scaling Advanced Message Queuing Protocol (AMQP) Architecture with Broker Federation and InfiniBand*. 2009, Department of Computer Science and Engineering, The Ohio State Universit.
114. Cheung, A.K.Y. and H.-A. Jacobsen, *Load Balancing Content-Based Publish/Subscribe Systems.* ACM Trans. Comput. Syst., 2010. 28(4): p. 1-55.
115. Tran, P., P. Greenfield, and I. Gorton. *Behavior and performance of message-oriented middleware systems*. in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*. 2002.
116. Chu, P.C. and J.E. Beasley, *A genetic algorithm for the generalised assignment problem.* Computers & Operations Research, 1997. 24(1): p. 17-23.
117. Sharkey, T.C. and H.E. Romeijn, *Greedy approaches for a class of nonlinear Generalized Assignment Problems.* Discrete Applied Mathematics, 2010. 158(5): p. 559-572.
118. Michalewicz, Z. *A Survey of Constraint Handling Techniques in Evolutionary Computation Methods*. in *4th Annual Conference on Evolutionary Programming*. 1995.
119. Coelho, R.F., *Multicriteria Optimization with Expert Rules for Mechanical Design*, in *Computational mechanics department*. 2004.
120. Montes, E.M., *Alternative Techniques to Handle Constraints in Evolutionary Optimization*, in *Electronic Engineering Department Computer Science Section*. 2004.
121. Qi, H. and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix.* SIAM journal on matrix analysis and applications, 2006. 28(2): p. 360-385.
122. Sahoo, A., K. Kant, and P. Mohapatra, *BGP convergence delay after multiple simultaneous router failures: Characterization and solutions.* Computer Communications, 2009. 32(7): p. 1207-1218.
123. Katz-Bassett, E., et al., *Towards IP geolocation using delay and topology measurements*, in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. 2006, ACM: Rio de Janeriro, Brazil.
124. Eriksson, B., et al., *A learning-based approach for IP geolocation*, in *Proceedings of the 11th international conference on Passive and active measurement*. 2010, Springer-Verlag: Zurich, Switzerland.
125. Bernard Wong, I.S., and Emin Gün Sirer,. *Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts*
in *4th USENIX Symposium on Networked Systems Design & Implementation*. 2007.
126. Wang, Y., et al. *Towards Street-Level Client-Independent IP Geolocation*. in *NSDI*. 2011.

127. Birrer, S. and F.E. Bustamante. *Resilience in Overlay Multicast Protocols*. in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*. 2006.
128. Costa, P. and D. Frey, *Publish-Subscribe Tree Maintenance over a DHT*, in *Proceedings of the Fourth International Workshop on Distributed Event-Based Systems (DEBS) (ICDCSW'05) - Volume 04*. 2005, IEEE Computer Society.
129. Frey, D. and A.L. Murphy. *Failure-Tolerant Overlay Trees for Large-Scale Dynamic Networks*. in *Peer-to-Peer Computing , 2008. P2P '08. Eighth International Conference on*. 2008.
130. Fei, Z. and M. Yang, *A proactive tree recovery mechanism for resilient overlay multicast.* IEEE/ACM Trans. Netw., 2007. 15(1): p. 173-186.
131. Ijspeert, A., et al., *Resilient Multi-path Routing Based on a Biological Attractor Selection Scheme*, in *Biologically Inspired Approaches to Advanced Information Technology*. 2006, Springer Berlin / Heidelberg. p. 48-63.
132. Li, Z. and P. Mohapaira. *The impact of topology on overlay routing service*. in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. 2004: IEEE.
133. Li, Z. and P. Mohapatra, *On investigating overlay service topologies.* Computer Networks, 2007. 51(1): p. 54-68.
134. Yuan, B., et al. *Improving chinese internet's resilience through degree rank based overlay relays placement*. in *Communications, 2008. ICC'08. IEEE International Conference on*. 2008: IEEE.
135. Han, J., D. Watson, and F. Jahanian, *Enhancing end-to-end availability and performance via topology-aware overlay networks.* Computer Networks, 2008. 52(16): p. 3029-3046.
136. Madhu Kumar, S. and U. Bellur. *A distributed algorithm for underlay aware and available overlay formation in event broker networks for publish/subscribe systems*. in *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on*. 2007: IEEE.
137. S D, M.K. and U. Bellur. *Availability models for underlay aware overlay networks*. in *Proceedings of the second international conference on Distributed event-based systems*. 2008: ACM.
138. Zhang, X., C.I. Phillips, and R.J. Mondragon, *Topology Construction of Provider-Independent Overlays to Improve Internet Resilience.* Communications Letters, IEEE, 2012. 16(11): p. 1876-1879.
139. Dijkstra, E.W., *A note on two problems in connexion with graphs.* Numerische mathematik, 1959. 1(1): p. 269-271.
140. Rouphail, N.M., et al. *A decision support system for dynamic pre-trip route planning*. in *Applications of Advanced Technologies in Transportation Engineering (1995)*. 1995: ASCE.
141. Wang, M. and T. Takada, *Macrospatial correlation model of seismic ground motions.* Earthquake spectra, 2005. 21(4): p. 1137-1156.
142. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. 2003
143. Alberto Medina, A.L., Ibrahim Matta, John Byers, *BRITE: Universal Topology Generation from a User's Perspective*. 2001.
144. Dutang, C. and D. Wuertz, *A note on random number generation.* 2009.