**International Journal of Digital Crime and Forensics**
Volume 10 • Issue 1 • January-March 2018

# Reliable Security Strategy for Message-Oriented Middleware

Guangxuan Chen, Zhejiang Police College, Institute of Software Application Technology, CAS, Guangzhou, China
Liping Ding, Digital Forensics Lab, Institute of Software Application Technology, CAS, Guangzhou, China
Guangxiao Chen, Universidad Carlos III de Madrid, Madrid, Spain
Panke Qin, Henan Polytechnic University, Jiaozuo, China

## ABSTRACT

This article explores a method to solve the security problems such as limited single-server processing power and single point of failure, and so on, a newly designed and developed Message-Oriented Middleware that supports clustering features. By improving the traditional reverse proxy cluster system, Message-Oriented Middleware has been made more applicable to message transmission services. A method of dynamic load balancing and load transfer was based on a variety of factors that was also designed according to the characteristics of a message service system. The method can solve problems like system instability and performance bottle-neck in Message-Oriented Middleware effectively and can increase the throughput of the system obviously.

### KEYWORDS

Data Synchronization, Load Balancing, Message-Oriented Middleware, Message Service, MOM

## 1. INTRODUCTION

Message-Oriented Middleware (MOM) can effectively resolve problems existed in the data transmission between network servers and clients, such as low system effectiveness and unreliability of transmission caused by different hardware platform, different network environment, mutual operations among different types of databases, and coexist of multi-application mode (Xiao & Wang, 2016; Wu, 2013). With the rapid growth of message service, message service system with single server can hardly solve problems brought by the sharp increase of client number and the increasing complicated business. The clusters link up multiple servers that work together and provide single mirror to the outside, which increased the number of servers transparently and achieved to solve the problems that single server unable to solve. These problems usually were by the sharp increase of client number, such as the server unable to respond to the users' requests. While, cluster makes full use of every effective service node and let all the active nodes handle the overall task jointly, so that in many respects they can be viewed as a single system. These nodes of the cluster usually connected to each other through fast local area networks and each node running its own instance of an operating system (Yang, Li, Qiu & Huang, 2010). Compared to single server system, cluster can significantly improve the throughput and parallel processing performance of the system.

The characteristics of the message and message service distinguished the cluster based on Message-Oriented middleware and the traditional cluster. In the traditional mode, the server-side bears a heavy burden of processing a lager mount of logical business, i.e., the clients urge the server

DOI: 10.4018/IJDCF.2018010102

to process certain logical business that clients can't handle with, and then get the process result from the server. Generally, the server need spend a certain amount of time handle with the clients' request, and in sometimes, it takes a long time to process the logical business.

While, Message-Oriented Middleware-based clusters possess with distinctive features compared to traditional clusters. Firstly, without heavy business processing work, the server only handles with the storage and retransmission of the message or creation and deletion of the queues and subjects. Secondly, the servers process many but simple tasks. Compared to the cluster systems which possess servers that can handle with complicated business, message clusters spend most of the time on network transmission and IO access. While for the former, the time spent on the storage of the shared database and network transmission can be negligible in regard to the time spend on the processing of the business (Zhao, Yin, Luo & Zhong, 2011).

The remainder of this paper is organized as follows: Section 2 introduces the integral structure of high reliability Message-Oriented middleware strategy we designed. Section 3 goes into particulars of the high reliability cluster strategy for Message-Oriented middleware respectively, i.e., load-balancing, data synchronization and fault handling, and a series of experiment results that come from different strategies are showed in order to enhance the strategy we proposed. Finally, Section 4 concludes with a summary of the new strategy and suggests future work.

## 2. INTEGRAL STRUCTURE OF HIGH RELIABILITY MOM STRATEGY

In order to guarantee the high reliability of the message transmission, we designed the message cluster system with four-layer structure (Figure 1). The four layers are client agent, load balancer, server and raid. The functions of the four components are as follows:

**Client Agent:** The client agent is responsible for the transparently transmission between different application layers, communication between server and client and load balancer, and maintaining the connection between the client and server (Li, Du, & Zhu, 2016).

**Load Balancer:** The load balancer is used for recording the location of the queue and subject, supporting the clients to establish connection to the specific server, maintaining the connection from the clients and choosing the servers with lighter load when new queue or subject created (Alakeel, 2010, pp. 153-160).

**Server:** Every server is responsible for the business conduction and maintaining the message index table of one or several specific queue and subject.

**Raid:** The raid is used for storing business data, queues and subject message.

Within the Message-Oriented middleware cluster, the high reliability is implemented through load-balancing, data synchronization and fault handling.

## 3. STRATEGY AND RESULTS ANALYSIS

Load balancing, data synchronization and fault handling are the three most important factors that could be used for describing the performance of a cluster strategy. The pros and cons of a cluster strategy can be directly reflected from the performance of the three factors. Thus, improving the algorithm and increasing the efficiency of the three factors can enhance the performance of this cluster strategy in a certain degree.

In the following text, we proposed a new strategy, i.e., improving the algorithm and increasing the efficiency of the three factors respectively, and setting the experiment results against those of different strategies, we will see a clear improvement.