IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Volume-4, Issue-4 E-ISSN: 2347-2693

A Survey on Availability and Scalability Requirements in Middleware Service Platform

Varsha V[1*] and Anooja Ali[2]

[1*, 2] Department of Computer Science,
Reva Institution of Technology and Management, Karnataka, India

Available online at: www.ijcseonline.org

Received: Mar/22/2016 Revised: Apr /02/2016 Accepted: Apr/14/2016 Published: Apr/30/2016

***Abstract*- Middleware is a software that resides on top of the operating system. It provides a development environment to the applications on top of it. It is used in diverse areas like distributed systems; networks embedded systems and so on. In telecommunication industries, middleware is also used to provide platform management functions such as context management, fault management, process and node management and so on. Availability and scalability play an important role in these industries that are customer oriented. This paper is a survey on how middleware based on symmetric multiprocessing and cluster systems provide better availability and scalability.**

Keywords: Availability, Middleware Platform, Scalability, Symmetric Multiprocessing

1. INTRODUCTION

Middleware is a software that resides on top of the Operating System. It ensures that the services offered by the Operating System are available to the application. Middleware platform includes WEB servers, application servers, tools and content management systems that support application development and delivery. Many of the failures will be handled by the middleware, without the programmer ever coming to know about them. The Quality of Service requirements improves with middleware component hiding the complexity of the underlying architecture.

Middleware is used in diverse areas and some of them include distributed system, networks, and embedded systems and so on.

Middleware in distributed system

"*A distributed system is a computer system where components of the system are held on physically separated, autonomous computers*. [1]"

Applications written in different languages, executing on different Operating Systems are supported by the distributed system. Here, middleware comes into the picture. It helps communication between two or more applications. Therefore, having a single common development and runtime environment is helpful. Middleware ensures that complexity is moved out of the applications while balancing the load, managing security and supporting different languages and platforms.

With middleware, the hardware and operating system are very stable (or fault tolerant) and the applications need not worry about the errors.

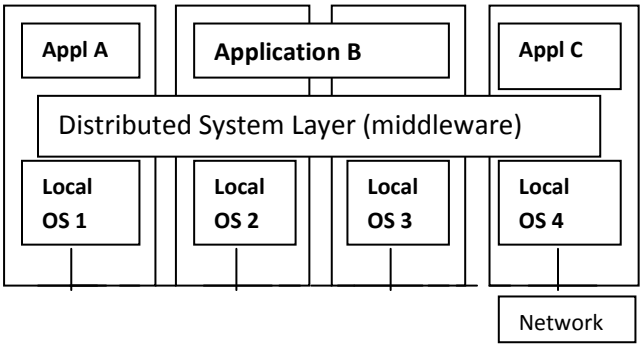

Figure 1: Implementation of middleware in distributed system

The middleware layer provides similar functionalities as the distributed OS. To offer additional functionalities required for zero downtime newer and customized collection of APIs. The advantage of middleware is that it supports usage of standards as much as possible:

- standard hardware
- standard UNIX-operating system
- software portfolio available on standard UNIX-system [2].

2. LITERATURE SURVEY

In the earlier days, middleware was not popular as computer themselves were expensive so attention was not paid for software reuse and developing of applications from existing software [3]. Over the years due to economic factors and technical advances middleware has gained importance. The need for distributed systems and

distributed applications rose at the emergence of the internet in the 1970s. Due to lack of methods, platform and tools, developing applications was difficult.
In the past twenty years, various technologies have emerged making the development of distributed systems and applications easier. Now there are various types of middleware for various fields; some of them are as follows:

- **Procedure oriented middleware** uses "serial communication protocol". Here, data is sent in a stream at a constant rate. The advantage is that it supports exception and different types of data formats. The disadvantage is that it is not scalable.
- **Object-oriented middleware** supports synchronous, asynchronous and deferred synchronous communication. It supports multiple transactions. There is a broker that facilitates communication between client and server components. The advantage is that it provides scalability and load balancing. The disadvantage is that legacy systems require wrapper code.
- **Message-oriented middleware** supports communication through message exchange. The message provides functions like notification of events and service execution requests. Message queues are used to provide asynchronous service. Advantages include better fault tolerance and scalability. The disadvantage is the lack of access transparency.

All of them have their own advantages and disadvantages and all of them strive to provide quality of service requirements.

3. MIDDLEWARE IN TELECOMMUNICATION INDUSTRIES

In telecommunication industries, managing platform efficiently prevents errors significantly. A platform would consist of hardware, operating system, and middleware. Since, these systems are used in conjunction rather than standalone, managing them remotely is vital. If the management is done well, it enhances the availability of the entire network [4].

Some of the management functions are as follows:

- Configuration management
- Fault management
- Performance management
- Process and node management
- Security management.

The most commonly used protocol is SNMP (Simple Network Management Protocol). The middleware service platform is expected to provide the following set of requirements: high availability, scalable performance, distribution of data and services, programming environment, development environment and so on. Among all the requirements, availability and scalability play an important role. One of the most promising approach to obtain these requirements is the combination of symmetric multiprocessing and clustering.

Symmetric multiprocessing includes multiprocessing system, where two or more processors that are identical and are connected to shared memory.

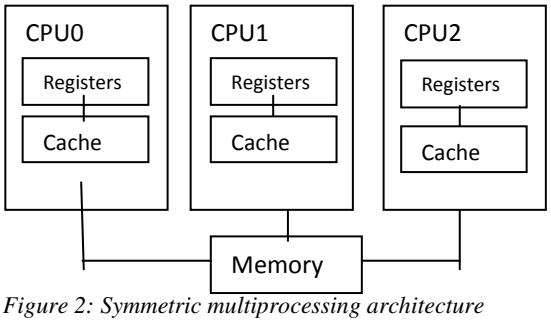

Figure 2: Symmetric multiprocessing architecture

These processors have full access to all I/O devices and are under the control of single Operating System instance. In a multi-core system, separate cores are treated processors, and SMP architecture applies to them. Symmetric multiprocessing can also be used with distributed systems.
The cluster systems consist of nodes which are basically autonomous systems based on Symmetric Multiprocessing
In cluster systems, if a there is a failure in a cluster node, another node takes over its tasks, thus increasing availability. With the standby systems readily available, downtime is bare minimum .
Middleware based on this system provide better availability and scalability.

4. AVAILABILITY CONCEPTS

Redundancy is the most common method used to increase reliability and availability of a system, by duplication of critical components, usually in the form of a backup or fail-safe. It is less expensive and easier to manage [5]. A cluster network provides the highest level of redundancy, and thereby availability in a telecommunication system.

- **Redundancy in a cluster**

"*A cluster is a system consisting of two or more nodes (hosts) having separate processors and memories that are interconnected.*" Since each component is duplicated at least once in a cluster, redundancy is achieved.
The nodes in the cluster network contain redundant hardware and software components. In the case of failure, these components can perform the functions of the failed node and reduces the effect of the errors. It is plausible to remove the flawed component without having to interrupt

the functionalities of operating system [6].
In active/active cluster mode scenario, the nodes do not wait for the failure of another node; rather they perform their respective functionalities. Hence, the nodes are always available.
If the application functions are distributed within the network, better availability and performance can be achieved. Workload balancing and a strong cluster-awareness of applications are essential.

5. SCALABILITY CONCEPTS

Scalability means to adapt the capacity and performance of a system according to the needs of an application.
Using cluster technology, one has a variety of possibilities to realize the scalability of a system which affects both its hardware and software components.

- **Hardware**

From a simplifying perspective, with a cluster there are basically two ways to scale up:

- Increase the number of cluster elements
- Scale up the nodes themselves.

The first possibility works well in clusters, where the overhead of communication between the nodes can be kept low. More performance is needed if there are more messages that have to be passed between the nodes to coordinate and synchronize the processes [6, 7].
Since the messaging overhead has an impact on system performance, the performance of a cluster with fewer nodes is better than one with a number of nodes. This means, that a two-node cluster cannot reach twice the performance of a single node cluster, and a four node cluster has less than four times the single-node performance.
In order to be able to choose the second possibility, the node itself must be scalable. Hence, the hardware to realize the required server must offer the possibility to extend hardware components such as

- Number of CPUs
- RAM capacity
- Disk capacity
- Controllers (e.g. Ethernet boards, SS7 boards)

- **Software**

Once the hardware is scaled up, the software should be capable of increasing the capacity. For example, a larger number of CPUs can be used by an application. The platform must offer mechanisms to hide the complex process and communication management from the application that are needed to implement scalability within a cluster.
Further, application load distribution and balancing among the nodes can be realized using communication mechanisms thus improving the scalability.

6. CONCLUSION

Middleware based on the combination of symmetric processing and clustering provides availability and scalability significantly.

Many methods have been introduced to obtain quality of service requirements and one such method is using clustering over the years. Clustering has become popular as it provides better reliability, scalability, availability and also maintenance can be done at ease.

Although clustering provides better QoS requirements, it is expensive; also a limitation that might occur is failures in applications due to dependency. This can be sorted out by using “hierarchical middleware architecture”. This architecture uses *overlay-network* to isolate faults in the application and eliminates the single point of failure in symmetric clustering middleware [8].

In the recent years, with the development of cloud computing, the concept of cloud middleware has come into the picture, which seems like the future of middleware in terms of quality of service requirements.

ACKNOWLEDGEMENTS

I would like to take this opportunity and thank Computer Science department, Reva Institute of Technology, for encouraging and supporting me throughout.

REFERENCES

[1] Gu, Tao, Hung Keng Pung, and Da Qing Zhang. "A middleware for building context-aware mobile services." Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th. Vol. 5. IEEE, 2004.
[2] Qilin, Li, and Zhou Mintian. "The state of the art in middleware." Information Technology and Applications (IFITA), 2010 International Forum on. Vol. 1. IEEE, 2010; pp 83-85
[3] Bishop, Toni A., and Ramesh K. Karne. "A Survey of Middleware." In Computers and Their Applications, pp. 254-258. 2003.
[4] Nokia white paper on telecommunication service platform
[5] Kanso, Ali, Maria Toeroe, and Ferhat Khendek. "Comparing redundancy models for high availability middleware." Computing 96.10 (2014): 975-993.
[6] Seshadri, S. Ling Liu ; Cooper, B.F. ; Chiu, L ; Gupta, K ; Muench, P, A Fault-Tolerant Middleware Architecture for High-Availability Storage Services, 2007,IEEE
[7] C. Engelmann, S. L. Scott, C. Leangsuksun, and X. He. Symmetric active/active high availability for high-performance computing system services. 2006, Journal of Computers (JCP).

[8] Galán-Jiménez, Jaime, and Alfonso Gazo-Cervero. "Overview and challenges of overlay networks: A survey." Int J Comput Sci Eng Surv (IJCSES) 2 (2011): 19-37.

Authors Profile

Varsha V is a student pursuing M.Tech in Computer Science from VTU, Belgaum, in Reva Institute of Technology and Management, and completed BE from Brindavan College of Engineering, in 2014.

Anooja Ali has received M.Tech in Computer Science from VTU, Belgaum in 2013 and BE from Cochin University of Science & Technology (CUSAT), Kerala in 2004. She is an Assistant Professor in School of Computing and Information Technology, Reva University, Bangalore, since August 2013. She has 8 years of experience in teaching. Her research is about "Key Agreement Protocols for Wireless Sensor Networks". She has published 4 papers in International Journals.