## Optimizing Load Distribution in JBoss Clusters through Enhanced

## Messaging Services

## Dr. Rajendra Kachhava, Sangam University, Bhilwara, Rajasthan

## Email: rajendrakachhava@gmail.com

### *Abstract—* This research paper presents a solution aimed at enhancing the operational efficiency of small to medium-sized Value Added Service (VAS) providers by establishing a highly available and scalable messaging platform for telecom operators. The proposed architecture leverages JBoss clustering in conjunction with a load balancer to ensure robust service delivery while minimizing costs and maintenance efforts. The first phase of the project involves the creation of a JBoss cluster comprising two nodes, integrated with a load balancer to support multiple Java messaging web applications. The second phase focuses on customizing existing applications, including subscription management, pre-subscription, mobile-terminating messages, and service message processes, to cater to specific client and node configurations. This approach aims to streamline day-to-day management, enhance automation, improve security, and reduce the need for additional manpower, ultimately facilitating a more efficient service environment for VAS providers.

***Key words: Computing, Cluster Computing, Jboss Clustering, Load Balancing, Messaging Service, Nodes.***

1. Introduction

Parallel computing has seen many changes since the days of the highly expensive and proprietary super computers. Changes and improvements in performance have also been seen in the area of mainframe computing for many environments. But these compute environments may not be the most cost effective and flexible solution for a problem. Over the past decade, cluster technologies have been developed that allow multiple low cost computers to work in a coordinated fashion to process applications. The economics, performance and flexibility of compute clusters makes cluster computing an attractive alternative to centralized computing models and the attendant to cost, inflexibility, and scalability issues inherent to these models [1].

Sports Service, a subscription based SMS application for sports fans delivering sports updates and information covering games fixtures, results, match report, standings, players-clubs- teams and statistics. The application offers Telecom operators to give update of most loved game in the world to their subscriber. Company developed Sports service as a tool/utility for end users to facilitate the flow of information of the game

covering all major sports championships. Users subscribe for the service via various mediums such as SMS, Web, social networking sites and USSD.

As subscriber and client base increased company feels improvement in existing technical resources. So there are multiple options available like client specific configuration, going for cluster computing or improving the existing infrastructure with jboss clustering. Improving existing JBoss system to cluster level looks feasible, secure, tested, low cost and high availability option for the company [4].

### Literature Survey and Related Work

In recent times, the up-to-date concept of cluster computing has turn into a main theme in the IT business. On the other hand, the nature of the idea is not state-of-the-art technology. Indeed, cluster computing is formed from a range of computing paradigms and it endeavour to offer convinced capabilities and deliver IT goods as services. The relevant computing concept is as below:

### Network Computing

In the early 1990s, Network Computing was developed and it thoroughly changed IT communications. This approach means that applications placed on servers in the network can be accessed remotely by client machines. The development of the World Wide Web (WWW) is the best practical example of network computing.

### Grid Computing

For most computers in America, on average, only 25 percent of the time of the central processing units is taken up by tasks in hand. Hence, the objective of Grid Computing was to reduce the geographical boundaries and connect remote computers into a network, in order to mingle idle resources of all computers on the grid and to build a virtual supercomputer. Similarly, a most important goal of cluster computing is to connect a great number of computers in a data processing centre through the networks, and to produce a high- performance computing power with the same technologies, such as distributed computing, parallel computing and cluster computing [6].

### Utility Computing (On-demand Computing)

Utility Computing could be described as a new concept for the stipulation of the grand computing power. The inaccessible and large-scale data processing centers, such as IBM, HP and Oracle suggest utility computing services and these services are accessible whenever consumers require them. Furthermore, these computer utility services are similar to electricity and gas in that consumers just need to pay for the quantity of their usage of computing power. Therefore, companies can reduce the costs of their investment in IT infrastructure by purchasing utility services on demand [2].

### Pervasive Computing

Pervasive computing refers to an emerging mobile technology. Most recently, it is plain to see that the broad usage of mobile devices such as laptop, Notebook, Personal Digital Assistant, Smartphone or some future device has caused calculation to become part of the computing environment. Pervasive software is frequently embedded in these devices and it is designed to automatically adjust to the surrounding computing environment.

### Service-Oriented Architecture (SOA)

In the last three years, IT professionals have started to give notice to SOA or service computing. Strictly speaking, SOA is a idea similar to Web 2.0 and cluster computing and it offered a network-based way for organizations to combine different systems suitably. In detail, the approach of SOA is to follow some principles such as interoperability, componentization, and reprocess and standards observance for modeling, creating, operating and managing company services [3].

### Evolution of computing environment

In recent years, Internet has been developing very speedily. The expenditure of storage, the power consumed by computer and hardware is increasing. The storage space in data center can’t meet our requirements and the system and service of original internet can’t resolve above questions, so we need new clarification. At the same time, big enterprises have to study data source fully to support its business [16].



# Figure 1: Evolution Chart for Computing

1. CLUSTERCOMPUTING

Cluster computing is best characterized as the integration of a number of off-the-shelf commodity computers and resources integrated through hardware, networks, and software to behave as a single computer. Initially, the terms cluster computing and high performance computing was viewed as one and the same. In this environment a computer system acts as the *backup* system to one or more *primary* systems. When there is a failure in a primary system, the critical applications running on that system are *failed over* to its designated backup system.

A cluster is set of computers which are connected to each other, and are physically located close to each other, in order to solve problems more efficiently. These types of clusters are also referred to as High Performance Computing (HPC) clusters, or simply Compute clusters [7].

### Architecture of Cluster Computer

In such cluster architecture, the network interface hardware is responsible for transmitting and receiving packets of data between nodes. The communication software should offer a fast and reliable means of data communication between nodes and potentially outside the cluster. For example, clusters with a special network like Myrinet use communication protocol such as Active Messages for fast communication among its nodes. This hardware interface bypasses the operating system and provides direct user-level access to the network interface, thus remove the critical communication overhead [8].



Figure 2: Architecture model of cluster computing.

### Benefits of Clusters

The benefits of the Recommended Basic Architecture are:

### Ease of administration

Because a single cluster hosts static HTTP pages, servlets, and EJBs, you can configure the entire Web application and deploy objects using the Web Logic Server Console.

You do not need to maintain a separate bank of Web servers (and configure Web Logic Server proxy plug-ins) to benefit from clustered servlets [4].

### Flexible load balancing

Using load balancing hardware directly in front of the Web Logic Server cluster enables you to use advanced load balancing policies for accessing both HTML and servlets content. For example, load balancer to detect current server loads and direct client requests appropriately [9].

### Robust security

Placing a firewall in front of your load balancing hardware enables you to set up a De-Militarized Zone (DMZ) for your web application using minimal firewall policies.

### Optimal performance

The combined tier architecture offers the best performance for applications in which most or all of the servlets or JSPs in the presentation tier typically access objects in the object tier, such as EJBs or JDBC objects.

1. Need of Jboss Clustering

This section will introduce the main concepts that form the foundation of the clustering features in JBoss. In particular, will see how clusters are defined, how they are built and how nodes communicate together. JBoss currently provides full clustering support for stateless session beans, state full session beans, entity beans and JNDI. Replication of HTTP sessions for web applications is also available [10].

### Architecture of Jboss clustering

****

Figure 3: Jboss Clustering Architecture

**JGroups** is a toolkit for reliable point-to-point and point-to- multipoint communication. JGroups is used for all clustering- related communications between nodes in a JBoss AS cluster. **JBoss Cache** is a highly flexible clustered transactional caching library. Many AS clustering services need to cache some state in memory while 1) ensuring for high availability purposes that a backup copy of that state is available on

another node if it can't otherwise be recreated (e.g. the contents of a web session) and 2) ensuring that the data cached on each node in the cluster is consistent. JBoss Cache handles these concerns for most JBoss AS clustered services information processing power distribution system can help to realize real-time monitoring and information collection [11].

V.MESSAGINGSYSEM

### System Overview

This section introduces the system context and design considerations summarizing the choice of specific technology and the architecture.

### System Design

**Technology driving the Soccerazzi Solution:**

* + Sports service has been developed on ***Java*** using

***Eclipse IDE3.2***.

* + Linux based operating environment.
	+ JBOSSApplicationserverversion5.X
	+ MySQL version 5.X, to provide the RDBMS support to application.
	+ Hibernate 3.0, providing an Object Relational Mapping Framework for java that maps an object- oriented domain model to a traditional relational database.
	+ Enterprise Application development environment
	+ A Short Message Service (SMS) gateway. The gateway is used to send and receive SMS using SMPP 3.4 towards the SMSC [17].

### System Architecture

Architecture provides a view of the system as whole comprising of structural elements of software and their externally visible properties along with the relationship between them. It also provides an indication about the external interfaces that exists [15].



Figure 4: System Architecture of Messaging System

### System External Interface

* + Interfacing with the **SMSC** using **SMPP 3.4** to send and receive SMS’s for subscription and application download etc.
	+ Interfacing with the **Charging Systems**. This depends on the Charging system available in the network. Currently supported protocols include **UCIP, SCAP UCAP and XML based API.** There are provisions to add more protocols to the service.
	+ Interfacing with USSD menu gateway for giving comprehensive soccer information to the subscribers Interface using **HTTP / SOAP** for integration to self care portals [12].

VI. Load Balancing

Now that we have a common vocabulary, we can begin to examine the basic load balancing transaction. As depicted, the load balancer will typically sit in-line between the client and the hosts that provide the services the client wants to use; like most things in load balancing, this is not a rule, but more of a best practice of the typical deployment. We will also assume that the load balancer is already configured with a virtual server that points to a cluster consisting of two service points. In this deployment scenario, it is also common for the hosts to have a return route that points back to the load balancer so that return traffic will be processed through it on its way back to the client [13].

### Structure of Load Balancing

Figure 5: Structure of Load Balancing

Load balancing or Network Load Balancing (NLB), which distributes incoming traffic through a network of connected servers, should be seen as a complement to clustering.

Load balancing balances the load of incoming network traffic and distributes the requests to the servers that best can handle them. Load balancing is mainly used for scalability and performance reasons [14].



Figure 6: Load balancing distribution of servers

VII. Proposed Architecture of System

Sports Messaging Service, a subscription based SMS application (service) for sports fans delivering real time sports updates and information covering games fixtures, results, match report, standings, players-clubs-teams and statistics. This application offers to Telecom operators to provide messaging updates of most loved games in the world to their subscriber. Company developed Sports service as a tool/utility for end users to facilitate the flow of information of the game covering all major sports championships. Users subscribe for the service via various mediums such as **Short Message Service** (**SMS**), Unstructured Supplementary Services Data (USSD), WAP [18].

Client1

Client2

Client3

Client4

Database

JbossServerNode1

JbossServerNode1

MessagingService

ApplicationsHibernate

MessagingService

ApplicationsHibernate

3rdPartyservers(Content/

Charging/SMSC

KANNEL(SMSgateway)

Loadbalancer

Figure 7: Proposed Architecture for Messaging Cluster

### Performance Evolution

Provide a suitable solution for small to medium level companies which are highly available, secure, scalable, low cost, low maintenance, already tested environment. So after study for separate server for each client and sport, cloud computing and JBoss clusters. We have decided to go for JBoss clustering option as this is fulfilling the companies’ entire requirement.

VIII. RESULT ANALYSIS

Based on our research purpose and objective, we tried to examine the development of messaging as a service which applied in other platform to messaging system. By doing this, we will implement cluster computing with messaging system using the JBoss java platform which evaluate the customer and system working efficiency s that concerned to effect to messaging as a service solution. Our main research question will be based on:

* How service can improve the messaging system?
* How better messaging system improve and describe using cluster computing and compared with other computing?

### Test Results

Table1 shows the metrics we obtained for each of the tested scenarios. Clearly, TCP load balancing out performs the other scenarios in both the mean transfer delay and the number of requests served per second.



Table 1: Performance of Various Policies

The three graphs shown in Figures 4, 5, and 6 capture the behavior of the cluster under the three scenarios tested. They show how many connections each machine serves per second. When we use no load balancing, we can see that Buick served the majority of requests. When we use Random load balancing or TCP load balancing we can see that the three servers are serving approximately the same number of connections per second leading to a *better* response time and throughput.



Figure 8: Total numbers of Connections served by each connection with no Load Balancing



Figure 9: Total Connections of Random Load Balancing

Figure 10:Total numbers of Connections served by each connection with TCP Load Balancing

CONCLUSION

High-performance cluster computing is enabling a new class of computationally intensive applications that are solving problems that were previously cost prohibitive for many enterprises. The use of commodity computers collaborating to resolve highly complex, computationally intensive tasks has broad application across several industry verticals such as chemistry or biology, quantum physics, petroleum exploration, crash test simulation, CG rendering, and financial risk analysis. However, cluster computing pushes the limits of server architectures, computing, and network performance.

FUTURE SCOPE

Cluster Computing is a vast concept and load balancing plays a very important role in case of VAS services. There is a huge scope of improvement in this area. We have discussed only two divisible load scheduling algorithms that can be applied to cluster, but there are still other approaches that can be applied to balance the load in cluster. The performance of the given service can also be increased by varying different parameters. In system and application migration challenge, issues such as organization politics and ownership, system and application dependencies which may affect how applications are to be migrated to the cluster. In service level agreement

management the issue of lack of proper SLA or inadequate SLA impacts on how organizations will use cluster computing while ensuring quality service to customers and without breaching any legal and security compliance.

References

[1]. C. Feisst, D. Schlesinger, and W. Frye, "Smart Grid, The Role of Electricity Infrastructure in Reducing Greenhouse Gas Emissions",*Cisco internet business solution group*, white paper, October 2008.

[2]. V. C. Gungor, B. Lu, and G. P. Hancke, “Opportunitiesand challenges of wireless sensor networks in smart grid,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557– 3564, Oct. 2010.

[3]. A. Bestavros, M. Crovella, J. Liu, and D. Martin "Distributed Packet Rewriting and it Application to Scalable Web Server Architectures," in *Proceedings of ICNP ‘98: the 6th IEEE International Conference on Network Protocols*, (Austin, TX), October 1998.

[4]. A. Bestavros, M. Crovella, J. Liu, and D. Martin, "Distributed Packet Rewriting and its Application to Scalable Server Architectures," Tech. Rep. BUCS-TR-98-003, Boston University, Computer Science Department, February 1998.

[5]. DanielM.Dias,WilliamKish, RajatMukherjee,and Renu Tewari, “A Scalable and Highly Available Web Server”, Proceedings of IEEE COMPCON'96.

[6]. R. Buyya (ed.), High Performance Cluster Computing: Systems and Architectures, Prentice Hall, 2007.

[7]. B. Burke and S. Labourey “JBoss Clustering,” JBoss Inc. 2004.

[8]. B. Burke, and S. Labourey “Clustering with JBoss 3.0,” O’Reilly 2002.

[9]. Y.V.Natis “JBoss Application Server Passes J2EE ‘Certification’ FT-23-5133,” Gartner pp. 1-2, 2004.

[10]. Y.V. Natis and M.Pezzini, et al. “Magic Quadrant for EnterpriseApplicationServers,2Q04M-228073,”Gartnerpp. 1-8, 2004.

[11]. H.P. Corporation “JBoss Application Server in an HP Proliant blade server architecture,” Hewlett Packard Corporation 2004.

[12]. B. Golden “Hands-On: Is JBoss Ready for Your Enterprise?,” Integration Developer News 2004.

[13]. Luis Aversa, Azer Bestavros “Load Balancinga Cluster of Web servers” Using Distributed Packet Rewriting, Computer Science Department Boston University funded by NSF research grant CCR-9706685..

[14]. Jeffery Mogul. “Network behavior of a busy Web server and its clients”. Research Report95/5, DEC Western Research Laboratory, October 1995.

[15]. Carpenter,V.Getov,G.Judd,T.Skjellum,andG.Fox, “MPI For Java: Position Document and Draft API Specification,”JavaGrandeForumTechnicalReportJGF-TR- 03, Nov. 1998.

[16]. A. Skjellum, N. E. Doss, and P. V. Bangalore, “Writing Libraries in MPI ,”Proc. Scalable Parallel Libraries Conference, Mississippi State, USA, Oct. 1993, pp. 166-173.

[17]. W. Gropp, E. Lusk, N. Doss, and A. Skjellum, “A High- Performance, Portable Implementation of the MPI MessagePassingInterfaceStandard,”ParallelComputing,vol.22,no.6, Sep. 1996, pp.789-828.

[18]. Foster, and N. Karonis,“ A Grid- Enabled MPI :Message Passing in Heterogeneous Distributed Computing Systems,”Proc.1998IEEE/ACMSupercomputingConference(SC98), Orlando,USA,Nov.1998.

.