A Scalable Application Server On Beowulf Clusters

A thesis presented in partial fulfilment of the requirement for the degree of

Master of Information Science

At Albany, Auckland, Massey University
New Zealand

Supervised by: **Dr. Chris Messom**

Michael Zhiyong Zhou

2005

Abstract

Application performance and scalability of a large distributed multi-tiered application is a core requirement for most of today's critical business applications.

I have investigated the scalability of a J2EE application server using the standard ECperf benchmark application in the Massey Beowulf Clusters namely the Sisters and the Helix. My testing environment consists of Open Source software: The integrated JBoss-Tomcat as the application server and the web server, along with PostgreSQL as the database. My testing programs were run on the clustered application server, which provide replication of the Enterprise Java Bean (EJB) objects.

I have completed various centralized and distributed tests using the JBoss Cluster. I concluded that clustering of the application server and web server will effectively increase the performance of the application running on them given sufficient system resources. The application performance will scale to a point where a bottleneck has occurred in the testing system, the bottleneck could be any resources included in the testing environment: the hardware, software, network and the application that is running.

Performance tuning for a large-scale J2EE application is a complicated issue, which is related to the resources available. However, by carefully identifying the performance bottleneck in the system with hardware, software, network, operating system and application configuration, I can improve the performance of the J2EE applications running in a Beowulf Cluster. The software bottleneck can be solved by changing the default settings, on the other hand, hardware bottlenecks are harder unless more investment are made to purchase higher speed and capacity hardware.

Acknowledgement

My greatest appreciation goes to my supervisor: Dr Chris Messom, who has provided endless support both technically and morally. His broad and insightful understanding of the technology involved, his willingness to research emerging technology, his concise yet plain explanation to technically challenging concepts has made my learning process a pleasant experience.

Thanks to Dr Martin Johnson, Mr Andre Barczak and other people who have contributed greatly developing the Massey Beowulf computers: the Sisters and the Helix (the supercomputer) and made it available to us. Of course, thanks to the Allan Wilson Centre for Molecular Ecology and Evolution (AWC) for funding the development of the supercomputer.

Thanks to support from the system administrator James and the manager Lorri.

Also thanks to Massey University for the award of a Vice-Chancellor's Masterate Scholarship to support my study financially.

I must also thank the people working in the Open Source communities that have provided high quality open source software for my study, especially to the people working on Linux operating system, JBoss Group, Apache Tomcat Project and PostgreSQL database.

Finally, thanks to my family for their understanding and support. They are my source of energy and pleasure.

Table of Contents

Table of Tables

Table of Figures

Table of Abbreviations

ANSI	American National Standards Institute
API	Application Programming Interface
BBop	Benchmark Business Operation
BSD	Berkeley Software Distribution
CCM	CORBA Component Model
CICS	Customer Information Control System
CLR	Common Language Runtime
CMP	Container-Managed Persistence
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial off-the-shell
CPU	Central Processing Unit
CSIRO	Commonwealth Scientific & Industrial Research Organization
DBMS	Database Management System
DCOM	Distributed Component Object Model
EJB	Enterprise JavaBeans
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol
IMS-TM	Information Management System Transaction Manager
INRIA	French National Institute For Research In Computer Science And Control
J2EE	Java 2 Platform, Enterprise Edition
JDBC	Java Database Connectivity
JDK	Java Development Kit
JMS	Java Message Service
JMX	Java Management Extensions
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JSP	Java Server Pages
JVM	Java Virtual Machines
LAN	Local Area Network
MPI	Message-passing Interface

MPP	Massively Parallel Processing
ODBC	Open Database Connectivity
OMA	Object Management Architecture
OMG	Object Management Group
ORB	Object Request Broker
PBS	Portable Batch System
PHP	Hypertext Preprocessor
PVM	Parallel Virtual Machine
RAIDb	Redundant Arrays of Inexpensive Database
RAM	Random Access Memory
RDBMS	Relational Database Management System
RMI-IIOP	Remote Method Invocation Over Internet Inter-Orb Protocol (Rmi Over Iiop)
SARs	Storage Area Network
SFTP	Secure File Transfer Protocol Message-passing Interface
SMP	Symmetric Multiprocessing
SOAP	Simple Object Access Protocol
SPI	service provider interface
SSH	Security Shell
SUT	System Under Test
UDDI	Universal Description Discovery And Integration
UNIX	Uniplexed Information and Computing System. (It was originally spelled "Unics.")
WAN	Large Area Network
WSDL	Web Services Description Language
XML	Extensible Markup Language

Chapter 1: Introduction

1.1 Introduction

This thesis presents a study of the performance and scalability of J2EE applications. In particular, I concentrate on the application server, which is the core component of the J2EE architecture. I use a cluster of JBoss application servers to test how the scalability and performance of a J2EE application is effected.

In this introductory chapter, I start with the motivation of the scalability study and explain why it is important in the business world. Then I give some brief technical review about how scalability can be achieved using current available hardware and software. I explain why my particular study is useful and finally give an overview of the contents in each chapter.

1.2 Motivation of scalability study for distributed applications

Large-scale distributed systems are becoming increasingly important in the world, especially with online business activities. The Internet has greatly improved the accessibility to online businesses, and the increased accessibility has promoted ever-increasing e-commerce applications. The performance and scalability of an application is critical for a successful business, as a business application needs to have high performance to achieve competitive advantages over their competitors.

Performance can refer to many aspects, such as scalability, availability, fault tolerance and load balancing. I am particular interested in the scalability of an application. A scalable application has the capacity to serve additional users or transactions without fundamentally altering the application's architecture or program design. If an application is scalable, you can maintain steady performance as the load increases simply by adding additional resources such as servers, processors or memory.

The two most common types of scalability that can be applied to affect the overall application performance are:

- Horizontal scalability: Adding more servers (web, application or database servers) to improve performance.
- Vertical scalability: Adding more physical resources (memory, processors or network cards) to a existing server to improve performance.

The key point of scalability is to decide how well an application will perform when the size of the problem increases. Scalability is not only critical to maintain current system functionality in a changing workload, but also a key factor to guarantee the system can keep up with the growth potential and has the ability to scale to meet future user's demand.

1.3 Today's technology support for scalable application

Today's technology has provided high-quality hardware and software to support the development and deployment of applications with good scalability and high performance.

For the computer hardware, we have consistently increasing computing power with the CPU speed doubling every 18 months, while the price of a personal computer is gradually getting cheaper. Various architectures built on PCs have provided fundamental support for high performance computing.

Supercomputers, which are the most powerful computers in the world, are getting more powerful. Beowulf Clusters, which are built using the Commercial off-the-shelf (COTS) components such as PCs, are gradually becoming more important in the supercomputer field [7]. A major merit of a Beowulf Cluster is its significant cost advantages over traditional mainframe supercomputers with similar computing capacity.

The wide adoptions of fast network connections, for either local or large area networks, as well as the Internet technology have enabled reliable communication facilities to support high performance applications. Combined with the supercomputer and the reliable Internet connections, it is much more practical to build a GRID [20], a network of supercomputers using today's technology.

Software has been developed to take advantage of the hardware architecture to achieve high performance and scalability. Using a cluster of application servers for a J2EE application, the application server components such as an EJB can be replicated across a cluster of application server machine. By load balancing the client request to members in the application server cluster, each client can interact concurrently with local copies of the same EJB component. This results in increased accessibility to computing power, thus, an increased application performance and scalability can be achieved.

I am going to investigate the J2EE application performance using open source software. JBoss, the leading open source application server has recently introduced cluster support, which I will use for my study.

1.4 Significance of my study

I am going to investigate application scalability using open source software running in the Beowulf Cluster. The advantage of this approach is that I have total control of the resource, because the Beowulf Cluster was built and maintained by our department in the Massey University, and the open source software can be used free of charge with access to the source code.

From a business point of view, I am using one of the most cost effective combined hardware and software for running J2EE applications. Building a Beowulf Cluster cost only 5% to 20% of total cost compared with traditional mainframe supercomputers with the same computing power [21]. The JBoss application server is free of charge but with most of the features a leading commercial application server provides. PostgreSQL is the most advanced open source database. By running J2EE applications using a Beowulf cluster as the hardware, the JBoss cluster, PostgreSQL as software, I can expect good scalability results. A good scalability result means that the system could be very useful for developing and deploying cost effective commercial applications.

1.5 Overall structure of the thesis

The thesis is organised into the following nine chapters:

Chapter 1 introduces the overall structure of the thesis. I start with the motivation for the scalability study, followed by current hardware and software technology that can be used to build scalable applications. I then give reasons why my particular approach is useful, and finish with the overall thesis structure.

Chapter 2 presents some of the background knowledge necessary for understanding my study. Three of the most important architectures for building large-scale distributed applications are presented and compared, this information helps to identify why I chose the J2EE architecture for my study. I give some of the related literature review and also state my research hypothesis.

Chapter 3 gives a detailed description of the hardware architecture of my study. I use the Beowulf Cluster computers in Massey University for my performance study. Starting with the general architectures of various high performance supercomputers, the advantages of cluster-based system are discussed. At last, the helix and sisters clusters in Massey University are introduced in details.

Chapter 4 covers the software used in the study. I have chosen all software from open source, which I am particularly interested in. I cover the software for running a distributed applications based on J2EE technology. To be more specific, I give some detail about why I choose the integrated JBoss-Tomcat as the application server and web server, the PostgreSQL as the database and the ECperf as my testing application.

Chapter 5 describes the details of the test design for both the Sisters and Helix. I start with different types of hardware architecture I will use, followed by some detailed information about how to run various test programs in the system. The last part gives my preliminary design selections on the type of test, and briefly identifies the reason for that selection.

Chapter 6 gives detailed testing procedures for my study in Sisters. I have done various tests based on different hardware architecture, software and the application configurations. For each type of test, I present with details about the test design and implementation procedures. Followed by a test result, Analysis and discussion on these results reveal several important conclusions.

Chapter 7 gives detailed testing procedures for my study in Helix. Again, I have followed a similar approach used for the Sisters. But the Helix concentrates on some different aspect of my study and reveals some different results as compared with the Sisters.

Chapter 8 covers the further analysis and discussions based on the results obtained on both Sisters and Helix. I have given a broader view on how to further improve the performance and scalability of my current system. A higher level of discussions about how to improve the current implementation and use better software features and hardware architecture are discussed.

Chapter 9 gives the conclusion to my study. Based on the analysis of testing results in the previous chapters, I make my final conclusions and show some of the contributions made by my study. I also anticipated the future work in my research field.

1.6 Summary

I have introduced the overall structure of the thesis. Firstly, I gave the motivation for the scalability study, followed by the technical support that can be used for building a scalable application. I then show why my particular approach is useful. Finally, I listed the major topics of each chapter in the thesis.