HIGH PERFORMANCE CLUSTER ARCHITECTURE

Prof. Anilkumar J. KAdam, Prof. S.U. Ghumbre

(AISSMS college of Engg. Pune)

ABSTRACT:

A computer cluster is a group of loosely coupled computers that work together closely so that in many respects they can be viewed as though they are a single computer. Clusters are commonly, but not always, connected through fast local area networks. Clusters are usually deployed to improve speed and/or reliability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or reliability.

Traditional cluster architecture consists of a server and some number of clients connected to each other via LAN cable. Everything takes place through this LAN cable. Here, everything refers to the broadcasting of messages, acknowledgements of these messages back to server and the data transfer. The route to every operation is a single LAN cable port.

In Advanced High Performance Cluster Architecture (HPCA), we are divided these tasks and they are done via different ports. For broadcasting messages, we are going to use serial port. For taking acknowledgements from multiple clients in parallel, we are going to use parallel port. For data transfer between the machines of cluster, we are going to use TCP/IP.

As we know, the computations are faster in C++ than in Java and coding power of Java is more than that of C++. Thus, we are going to write coding part in Java and computational part is given to C++. For maintaining connection between C++ and Java, the technology called JNI (Java Native Interface) is used. Because of all these modifications in the trivial cluster architecture, we make an Advanced High Performance Cluster Architecture (HPCA).

Keywords : Parallel processing, cluster computing, Dedicated Cluster

1.INTRODUCTION

In parallel computing, improving the performance of computation is very important issue. Collective communication functions are frequently called and their execution time accounts certain amount of whole execution time in parallel applications. Predicting the execution time of these functions is important to improve their performance. In order to formalize the behavior of communication, many communication models have been introduced. [5]

Clusters are not commodities in themselves, although they may be based on commodity hardware. A number of decisions need to be made (for example, what type of hardware the nodes run on, which interconnect to use, and which type of switching architecture to build on) before assembling a cluster range. Each decision will affect the others, and some will probably be dictated by the intended use of the cluster. Selecting the right cluster elements involves an understanding of the application and the necessary resources that include, but are not limited to, storage, throughput, latency, and number of nodes. [7]

A cluster is a type of parallel or distributed computer system, which consists of a collection of inter-connected stand-alone computers working together as a single integrated computing resource. The typical architecture of a cluster computer is shown in Figure 1. The key components of a cluster include, multiple standalone computers (PCs, Workstations, or SMP's), an operating systems, a high performance interconnect, communication software, middleware, and applications. [4]

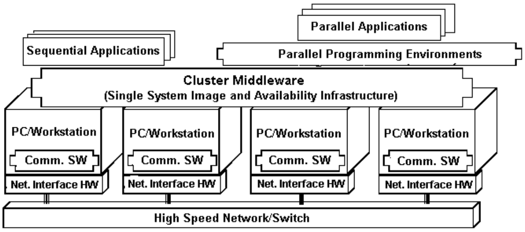

Figure 1.1 General Cluster Architecture.

A key component in cluster architecture is the choice of interconnection technology.

Interconnection technologies may be classified into four categories, depending on whether the internal connection is from the I/O bus or the memory bus, and depending on whether the communication between the computers is performed primarily using messages or using shared storage. Table 1 illustrates the four types of interconnections.[7]

Table 1.1 Categories of Cluster Interconnection Hardware.

	Message Based	Shared Storage
I/O Attached	Most common type, includes most high-speed networks; VIA, TCP/IP	Shared disk subsystems
Memory Attached	Usually implemented in software as optimizations of I/O attached message-based	Global shared memory, Distributed shared memory

Of the four interconnect categories, I/O attached message-based systems are by far the most common. This category includes all commonly-used wide-area and local-area network technologies, and includes several recent products that are specifically designed for cluster computing. I/O attached shared storage systems include computers that share a common disk sub-system. Memory attached systems are less common, since the memory bus of an individual computer generally has a design that is unique to that type of computer. However, many memory-attached systems are implemented in software or with memory mapped I/O, such as Reflective Memory. Hybrid systems that combine the features of more than one category also exist, such as the Infiniband standard, which is an I/O attached interconnection that can be used to send data to a shared disk sub-system as well as to send messages to another computer [7].

High performance clusters (HPCs) based on commodity hardware are becoming more and more popular in the parallel computing community. These new platforms offer a hardware capable of a very low latency and a very high throughput at an unbeatable cost, making them attractive for a large variety of parallel and distributed applications. However, for parallel applications that present a high communication/computation ratio, it is still essential to provide the lowest latency in order to minimize the communication overhead [4].

2.SIMPLE CLUSTER OMPUTING ARCHITECTURE:

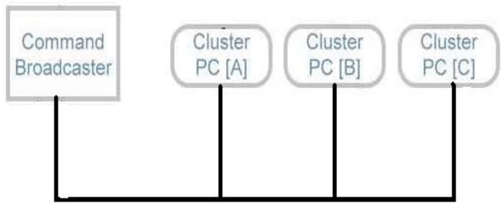

Fig 2.1 LAN (Data Transmission, Command, Acknowledgement)

•LAN is used to transmit data, command routing and acknowledgement.

•To execute the complex process, it requires more time (Time-consuming).

When the network is a shared hub, all nodes connected to the hub share the same collision domain and only one computer on the network can send a message at a time. With a switched hub each computer is in a different collision domain. In this case, two sets of nodes may be communicating at the same time as long as they are not sending to the same receiver. For cluster computing, switched networks are preferred since it allows multiple simultaneous messages to be sent, which can improve overall application performance [6].

3. PROBLEM OF TCP AND UDP PROTOCOL:

TCP supports only peer-to-peer connections (that is, one transmitter talking to one receiver) so a communication from, say, one transmitter to ten receivers requires ten connections and ten independent transmissions. There is no concept of *multicast.* In
cluster computing, it is increasingly common for many applications on different hosts to share a common data stream [1].

Since TCP do not support multicast, they have no concept of multicast group management, i.e., there is no ability to define a group of transmitters and receivers or for the application program to define the reliability requirements of the group members. Cluster multicast applications should have the option of knowing the identity and/or the status of the multicast receivers [1].

TCP supports only fully reliable transmission (i.e., it repeats missing data until it finally arrives at the destination, no matter how long it takes) which is too strong for some applications. UDP supports only a "best-effort" service (i.e., the transmitter never knows if the receiver got the data) which is too weak for some applications. Data reliability is defined by the protocol, not by the application. Cluster applications should be able to define their own reliability paradigm on a connection-by-connection basis

To send a single message reliably (as with transactions), TCP requires the exchange of six packets (two to set up and acknowledge the connection, two to send and acknowledge the data, and two to close the connection). This slows the exchange of data for shortlived connections since connection setup and teardown are accomplished separately from data transfer [1].

For cluster applications, transactions need to be fast and efficient. When errors do occur, some TCP implementations recover using a go-back-n algorithm that retransmits all data beginning at the point of loss [1].

TCP, UDP all implement fixed policies defined by the protocol, not the application. For example, all connections are governed by flow control, which restricts the amount of data the transmitter can send [1].

4. PROPOSED SOLUTION

This product is specially tailored by considering the huge computation needs of the complex applications. The basic and most important factor is to have a high performance and low cost super scalar architecture.

The project is based on a concept of improving performance by splitting the task over the network across multiple clusters. Clusters thus provide for a super scalar architecture with improved performance, faster and at a low cost alternative.

This is a platform independent product and doesn't depend on any other product for its implementation and is self-sufficient. Only dependency to illustrate its great many benefits is that of the application that will operate on the distributed environment. It will be able to optimize the workload of usually huge tasks by dividing them across the network node (clusters).

- The projects aim is to combine hardware connectivity features of cluster computing and network connectivity features to produce faster parallel networks.
- Additional ports like Serial and Parallel ports are used to communicate with all the PCs in the network apart from LAN.
- This enables us to use TCP/IP protocols used in grid computing.
- The data transfer is done using TCP/IP protocols and command routing is done using serial and parallel ports.
- Task-division architecture is used to distribute the tasks among various computers.
- A central server is maintained to command all the subordinate PCs.

5. CONCLUSION

By implementing cluster based parallelism and making efficient utilization of available hardware resources, we attempt to provide cost effective solution for small & medium scale businesses and research institutes. Instead of using conventional network support to communicate between clusters we propose the use of a dedicated hardware and cabling to speed up processes than existing systems. Use of hardware legacy ports speeds up the performance exponentially and helps in good utilization of available resources. From performance evaluation, we confirmed that High performance clustering Architecture is effective in improving the throughput.

6. REFERENCES

[1]Alfred C. Weaver, University of Virginia, (June 2002), “A Network Protocol for Cluster Computing”,Page (1-2).

[2] Christopher Batty(Editor), (Oct 2003) “ Using The Java Native Interface”,Page (2-5)

[3] Evgeniy Gabrilovich Lev Finkelstein(Editor), “ JNI – C++ integration made easy ” ,Page(1-2).

[4] Mark Baker, Amy Apon, Rajkumar Buyya, Hai Jin (September 2000) “Cluster Computing and Applications”,Page (2-3).

[5] Nomura, A. Matsuba, H.Ishikawa, Y.Dept. of Computer Sci., Tokyo Univ., (Sept. 2008), “Network Performance Model for TCPIP Based Cluster Computing”, Page(1).

[6]Pham, C.D. Albrecht, C.Univ. Claude Bernard Lyon I, France (June 2002), “Optimizing Message Aggregation for Parallel Simulation on High Performance Clusters”page(1-2).

[7] San Jose(Editor) ,(Jan 2004), “Cluster Computing”, CA 95134-1706 USA,Page(1).

[8] Xilink(Editor), The DCT/IDCT Solution (February 2000),”Customer Tutorial”, Page(1-16).

[9] Yunhong Gu and Robert Grossman, University of Illinois at Chicago(Jan 2000), “Using UDP for Reliable Data Transfer over High Bandwidth-Delay Product Networks , USA” , Page(1).