International Journal of Computing Science and Information Technology, 2013, Vol. 01 (01), 05-09
ISSN: 2278-9669, January 2013 (http://ijcsit.org)

Fail-over Clusters – A Review

Pabitra Mohan Khilar
Dept. of Comp. Sc. & Engg., NIT, Rourkela, India
pmkhilar@nitrkl.ac.in

Abstract

A cluster of workstations is increasingly used in many organizations including industries as well as academic institution for automation. As off-the-self components are widely available and becoming cheaper, the cluster of workstations (or simply clusters) is becoming popular compared to traditional high cost supercomputers. Due to their high scalability, clusters can be tuned to performance requirement of computational intensive applications. Nevertheless communication intensive applications (such as Google, Yahoo, Amazan clusters) can exploit the performance advantages of clusters as well. Development of highly reliable and available clusters has become indispensable as an alternative to supercomputers. In fact, many applications need increasingly higher levels of availability and reliability. As a result of this demand, the base server and storage components need to have number of features that help isolate single points of failure. It is good practice to structure a service so that it is not dependent on the availability of a single server. While certain faults in the environment can be entirely transparent to users, other faults may appear as a temporary dislocation of service. Cluster computing is becoming a widely accepted alternative method of providing higher availability and reliability using mainstream computing products than can be provided by a single system. The prime reason that most user organizations consider adoption of the cluster computing approach is high availability, which is the reason "non stop" systems were originally introduced. Specialized and proprietary fault tolerant systems have been used to provide high availability using sophisticated techniques to provide redundancy and resilience. However, the trade off is that the cost of such systems is typically high. In order for a cluster to provide high availability it must be able to recover from service failures. This paper's objective is to comment on the fail over characteristics of the various existing clusters found in industry. Here, we have surveyed some well known clusters that appear in top five hundred supercomputers/clusters used in the world, and presented a comparison among them.

Keywords:: workstations, cluster, cluster computing, supercomputing

1. Introduction

To start with, it is necessary to point out what is DNA, its usage in deferent areas of Biocomputing in general, and specifically point outwhy it was used in finite state machines. DNA (Deoxyribonucleic Acid) is a nucleic acid [1], which contains certain genetic instructions, used for the functioning of living organisms. DNA computing is fundamentally similar to parallel computing in that it takes advantage of the many different molecules of DNA to try many different possibilities at once. For certain specialized problems, DNA computers are faster and smaller than any other computer built so far. Furthermore, particular mathematical computations have been demonstrated to work on a DNA computer. [2, 3].

DNA nanotechnology uses the unique molecular recognition properties of DNA and other nucleic acids to create self-assembling branched DNA complexes with useful properties. DNA is thus used as a struct Clusters of computers are rapidly catching up with the processing capability of more costly supercomputers, traditional processors and proprietary mission-critical systems. Cluster-based systems are widely used in the commercial sector. A cluster is a group of independent computer systems called as nodes or hosts that work together as single system sharing a common workload. These nodes are used in running special computing tasks like high performance computing (HPC), High-Availability (HA), Load Balancing, etc. A HPC cluster performs computational tasks involving huge amount of processing requirements. The job is split, parallelised across the nodes by which complex problems which were not solvable or used to take more time on a single machine can be executed in less time on HPC clusters.

The vast computing potential of clusters is often hampered by their susceptibility to failures. Therefore, many fault-tolerant techniques have been developed to add reliability and high availability to cluster systems. The well-known unreliability in the OS or a failure in the hardware can be rectified by using clusters, where each node can monitor each other's activities and then be able to recover from system failures. High Availability (HA for short)

refers to the availability of resources in a computer system, in the wake of component failures in the system. This can be achieved in a variety of ways, spanning the entire spectrum ranging at the one end from solutions that utilize custom and redundant hardware to ensure availability, to the other end to the solutions that provide software solutions using off-the-shelf hardware components. These products survive single points of failure in the system. It provides fault tolerance, highly available servers with minimum down time. Failover clusters are designed so that there is always a second machine (and possibly more) that waits until the primary machine in the cluster fails. When the primary machine fails, the secondary assumes it's duties in a way transparent to the users.

The prime reason that most user organizations consider adoption of the cluster computing approach is high availability, which is the reason "non stop" systems were originally introduced. To provide an overview of availability issues across the various cluster offerings this paper focuses on issues such as fault tolerance, reliability and availability. In section 1, the demand of cluster technology for the current generation is introduced. Fault tolerance in clusters is discussed in section 2. Section 3 deals with various approaches for implementing failover clusters. The operating systems those are being used for governing the control over various resources in clusters are presented in section 4. High available clusters have been discussed in section 5. The comparison among different types of clusters is presented in section 6. Finally the paper concludes in section 7.

2. Fault – Tolerance

Fault-tolerance can be defined as the ability of a system to deliver the expected service even in the presence of faults. The fault in system can be: (i) permanent (ii) intermittent, (iii) transient and (iv) Byzantine. Permanent faults are caused by the total failure of a computing unit and are typically tolerated by using hardware redundancy, such as spare machines. Transient faults are temporary malfunctions of the computing unit or any other associated components, which causes an incorrect result to be computed. Intermittent faults are repeated occurrences of transient faults. The possible causes of transient faults include limitations in the accuracy of electromechanical devices, electromagnetic radiation received by interconnections, power fluctuations not properly filtered by power supply, etc. Several studies have indicated that transient faults are significantly more frequent than permanent faults. To counter such faults, fault tolerance mechanisms have typically relied on redundant hardware components with multiple replicas of essential applications running on separate hardware components. However, hardware redundancy is expensive and leads to wastage of resources.

Fault model will vary depending on the type of system and it's specification. It also varies with the applications requirements. For example, fault model for real-time applications will vary from that of non-real-time applications in terms of their specifications. In the event of a failure, the redundant components depend on the number of faults the system is designed to tolerate. Desired level of fault-tolerance can be achieved through temporal-redundancy or spatial redundancy. For non-real-time applications, optimizations of spatial-redundancy are desired where as for real-time applications, temporal redundancy is essential. To achieve high reliability, full redundancy is desirable, but it wastes the resources and is expensive especially on the non-occurrence of failure events. So, designers must balance the trade-offs to arrive at a correct mix.

Depending on the nature, faults can be classified as software and hardware faults. A software fault may arise due to a design or coding fault in a software system, while hardware faults refer to any deviation in the functioning of hardware component. Transient disturbances, environmental interference and failure of component may result in hardware fault. Sometimes hardware faults also force a software fault. For example, due to a transient fault, a processor may start executing a different code. For a system to be reliable, fault-tolerance must be built into the system to address all such faults.

Apart from redundancy, the following techniques are usually employed to mitigate the effect of faults. They are: (i) Fault Detection and diagnosis-the different methods for failure detection and diagnosis are (a) parity/CRC check (b) self-test i.e. periodically running a routine to check the status, (c) watchdog timers/heartbeat/keep-alive-all uses a time-out mechanism. (ii) Fault masking and Fault-Containments-Fault masking involves con-current masking and initiating mechanisms to correct the errors already generated. Two common types of fault masking are (i) hierarchical and (ii) group masking. To stop the propagation of erroneous results due to a fault before it is detected, fault containment technique is used. (iii) Roll back: In roll-back, the system rolls back to the last recorded state on detecting a fault. (iv) Recovery and Reconfiguration: Recovery mechanisms provide a means to bypass a fault. After recovery from a fault, the system states are again invoked after reconfiguration. One general technique for recovery through redundancy such as TMR (Triple Modular Redundancy), Voter based NMR etc.

Scheduling redundant tasks is another method to achieve fault-tolerance. This method of achieving fault-tolerance is called the primary-backup or primary-standby approach. Copies of a subtask are

scheduled on different machines to achieve fault tolerance. This approach tolerates a single fault occurring in the system when only one backup is scheduled. In general, to be able to tolerate n faults, we need to schedule n back-up copies on n different machines. Most of the clusters used to follow any one of the primary/back-up approach: (i) active-active or (ii) active-passive model [1].

3. Approaches for implementing fail over clusters

There are several different ways of implementing the cluster. Four of them are as follows: [1], [2], [3].

(i) Idle Standby: There is a Master computer and a cluster of standby computers. As soon as the Master goes down, a computer from the cluster, which has the highest priority then takes over. If a machine with a higher priority is inserted into the cluster, it will take over as the master (causing a brief service outage).

(ii) Rotating Standby: This implementation is similar to the Idle Standby implementation, but the first machine in the cluster is the active one. When it goes down the second node will become the active node, and when the original node is re-inserted into the cluster, it becomes the standby node.

(iii) Simple Fail over: This strategy is similar to the Idle Standby strategy, but rather than having the second computer idle, the second computer will be a different server, for example a development computer, but if the master server goes down, it will take over.

(iv) Mutual Takeover: Basically this strategy is a combination of Idle Standby and Simple Failover. Two machines are serving different applications and each is designed to take over the other's duties plus it's own if the other node goes down.

4. Role of Cluster Operating System

The role of operating system on the fail over characteristics is equally important as like hardware and programming tools. The fail over role of four operating systems such as Solaris MC, GLUnix, UnixWare NSC, Linux are discussed.

4.1 Solaris MC

It provides high availability in many ways, including failure detection and membership service, an object and communication framework, a system reconfiguration service and a user-level program reconfiguration service. Some nodes can be specified to run as backup nodes, if a node attached to a disk drive fails, for example. Any failures are transparent to the user.

4.2 GLUnix- (Global Layer Unix For Network of workstations)

The GLUnix consists of a per cluster master, per node daemon, and per-application library. GLUnix is currently able to handle failures in the daemon and startup processes. The master continually monitors each daemon and startup process. If it detects that a daemon has failed, it looks up a process database and for any sequential processes running in that node, it marks them killed. It then notifies the startup process. If there were any parallel processes in the node, the master sends a SIGKILL to all the other associated processes. Finally the master removes the node from the node database and continues normal operation. When a startup process fails, the master sends a SIGKILL to all remote processes, removes the node from the node database, and continues normal operation. [4]

4.3 UnixWare NSC (Non Stop Computers)

It emphasizes fault tolerance, and was designed with continuous availability in mind. If a node fails or an application running on a node fails, processes are automatically migrated to the remaining nodes and resumed. Administrator tools allow any failed nodes to be restarted and returned to operation as quickly as possible. Most other clustering systems are simply "high availability clusters ", whereas UnixWare NSC is a fault-tolerant cluster with high-availability nodes. UnixWare NSC is ideal in environments where availability is an important requirement. It's support for application failover uses an "n+1" approach. The backup copy of the application can be restarted on any of the nodes in the cluster, allowing one node to act as a backup node for all the other nodes. Many other cluster architectures, such as the Hewlett-Packard Service Guard, apply a "1+1" approach, which requires a dedicated spare node for each active node in the cluster. The ability to migrate processes automatically means that nodes can be added and removed without disruption. UnixWare NSC takes approximately 2 to 10 minutes to migrate applications to different nodes. During this time, those applications are not available, but there are many situations whereas this acceptable, and are preferable to an hour of downtime. Traditional clusters typically force applications to fit their availability model. UnixWare NSC's SSI enables applications and upper levels of the operating system to run without modification. UnixWare NSC supports the ServerNet fault-tolerant interconnect technology. Any element in a ServerNet cluster-processor, disk or I/O device-can interact with any other element without processor intervention. ServerNet latency is at 0.3 microsecond as opposed to 200 microsecond for Fast Ethernet or 50 microsecond for SCI [6]. Comparable products are primarily RISC-based servers from Sun Microsystems, HP and IBM. These servers are also highly scalable and can provide significant quantity of memory and storage, but they cost more than standard

Intel-based PCs, and fault-zones increase as memory and storage increase at each node.

4.4 Linux

Linux is known as a stable operating system. However, a Linux client/server configuration can have several points of failure, including the server hardware, the networking components, and the server-based applications. As more administrators choose Linux for critical applications, the demand for high-availability (HA) clustering for the Linux platform is increasing. In response, a number of Linux distributors have designed and implemented bundled HA solutions in their products, and numerous third-party add-ons are now available. HA clustering provides availability, performance, and capacity. What type of takeover strategy works best depends on local network topology. Three basic kinds are: (i) IP address takeover (ii) MAC address takeover (iii) Dynamic DNS reconfiguration.The High Availability Linux Project has another approach to clustering. The aim of the project is to create a cluster of servers, where one server will automatically take over the workload of another server if it goes down. This type of cluster is not aimed at parallel processing, or load sharing. Its one goal is to have a pool of computers to take over from a server if it stops working [2].

5. High Available Clusters

The following availability criterions are considered to compare the fail over characteristics of different clusters:(i) Long distance (>100 miles) cluster disaster fail over capability, (ii) Form of fail over to client-Totally transparent to user on client station, Application continues but in flight transactions are lost, User is disconnected from application (requiring intervention to reconnect to application), (iii) Form of action required by cluster administration staff to effect application fail over to other nodes within cluster-Nothing automatic fail over, Manual intervention required, Manual initiated fail over for any cluster resource including client, (iv) Form of preparation for fail over-No preparation required, Recovery scripts have to describe each application and the actions to be performed when various failures occur, Program requires change (APIs embedded, decomposition and recompilation, etc), Automatic distribution of failed node applications across the remainder of "good" nodes and resources [Load balancing], As above with manual override on and exception basis, Manual load balancing,

Some of the clusters that are chosen for discussion and comparison based on the above criteria are: Sun, IBM, Digital VMS and their variants. [2], [3].

5.1 Sun

Sun has two cluster offerings, Solstice High Availability (HA) Cluster and Sun Parallel Database (PDB) Cluster. Sun Solstice HA is aimed at providing high availability by application fail over from one node to the other. Sun HA clusters are set up in two ways: hot standby or mutual backup. Hot standby is characterized by the active workload being processed on one node while the second node is essentially idle, in case of failure the smart cluster agents in conjunction with pre-set fail-over specifications for each application and API's in Solaris, fail-over the applications to the second node. For fast restart, applications are pre-loaded in the second node but the state of the failed application is not saved and in flight transactions may be lost. Use of a transaction monitor in conjunction with Solstice HA can protect lost transactions and make the failover transparent to users connected to applications. Mutual backup works in a similar manner to hot standby with the exceptions that both systems are running active applications, thus providing a higher capacity than in hot standby, however in the event of fail-over, application termination and reload from one node to the other increase the fail-over time [3].

5.2 IBM

There are two separate and distinct IBM cluster offerings. IBM Parallel Sysplex based on enterprise S/390 nodes and HACMP Cluster based on RS/6000 and SP2 nodes. Strengths of HACMP include availability, configuration flexibility, connectivity and cluster management. IBM Parallel Sysplex is comprised of S/390 systems and their CMOS based predecessors, and is very much an enterprise class cluster system. If an organization is using S/390 systems and does not wish to change its applications, the step to higher availability is to implement Parallel Sysplex. The distance restrictions on the coupling facility allow node separation of a few kilometers, thus the Parallel Sysplex clusters provide continuous availability from the viewpoint of multiple component failures and single node failures but do not support long distance disaster recovery [7].

5.3 Digital OpenVMS Clusters

Cluster systems have been an inherent part of the VMS operating system concept since the early 1980s and VAX Clusters. Thus cluster operation is fundamental to VMS, unlike most systems, which consist of a UNIX operating system and add-on cluster software. In terms of high availability, it is better as fail-over is transparent to the user. There is no preparation required either by the programming or systems administration staff to enable regular single system style applications to fail-over to other nodes, and load balancing is automatic. The nodes in a cluster can be geographically separate to provide flexible distributed computing and disaster tolerance. Nodes can be separated by up to 500 miles with synchronous operation or up to several thousand miles

with asynchronous operation [5].

6. Comparison

More nodes can participate in a cluster from the viewpoint of extra computing power and level of resilience in case of failure in one or more nodes. Digital VMS and IBM Parallel Sysplex systems exceed the capability of the other vendors in terms of the number of nodes that can participate in a cluster system, Digital VMS can have up to 96 nodes and IBM Parallel Sysplex 32. The other systems cannot configure more than 8 nodes in their clusters, however both Sun's high availability Solstice offering and the Parallel Database cluster have a maximum of 2 nodes. In the OpenVMS environment, the operations staff don't take any action to cause the applications running on the failed node to fail-over to another node. Most of the reviewed cluster systems enable the application to failover and for the client user to continue to use the application with only a potential minor interruption if the last action was not committed to the database or file system at the point of failure. OpenVMS provides transparent fail-over from the viewpoint of a user connected to an application, without loss of in-flight transactions. OpenVMS stores the state of applications which are reapplied after failover. IBM Sysplex requires users to re-log onto the node where the failed-over application has been established, as a matter of security. HP MC/Service Guard, HP X Class and Sun PDB Clusters all require the user to re-establish the session after node failure. Digital OpenVMS is the only cluster system which currently allows nodes in the cluster to participate over long distances (upto 500 miles), providing both a disaster fail-over capability and a mechanism for load-sharing across geographically separate computing centers with little performance impact. Sun PDB Cluster and HP MC/LockManager do not currently implement fail-over in the normal sense, they rely on copies of the business application and parallel database software running in each node. If a particular business application is only active in one node, then in a node failure situation, that application will require human intervention to reinstate it on a remaining node. In the Sun PDB environment, API have recently been introduced which, if embedded in the user applications, can automate the fail-over process.In terms of preparation for fail-over, most cluster systems require scripts to be written to describe the applications, their associations in terms of physical or logical files, and the actions required in cases of failure. Only Digital OpenVMS clusters requires no preparation in terms of scripts or changes to the applications. Other properties considered include the ability to have multiple and different versions of the operating system running within the cluster, the range of storage devices supported and effectively shared, and the flexibility to have any node as a failover resource.

7. Conclusion

Many studies have been conducted on the impact of computer system downtime in business. The results of these studies point to the conclusions that the business impact of downtime includes: lost productivity, customer dissatisfaction, lost revenues and lost customers. High availability is important for many applications. In many cases, multiple servers will be operated to achieve high availability. During normal operation, the nodes within the cluster offer the defined services to clients. In the event of a planned shutdown, or failure of a node, the services that are being offered by this node are transferred over to one of the remaining nodes in the cluster and then offered to the clients once again through the same logical entity. As soon as the service is restored, the client is able to address the service as before the fail over. To the client, therefore, the service appears to be identical after fail over. There is, however, a period of time during the duration of the fail-over, when the service is not available to the client.

High availability is the primary characteristic that causes most users to consider adoption of cluster computing. The most significant elements are: the way in which a failure affects an application user, the amount of action that the cluster operations staff must perform to enact applications fail-over, and the amount of preparation that the user organization's MIS department must undertake to make applications fail-over as part of the cluster operation. The normal form of fail-over reviewed is where a given application is running in a node experiences a failure condition which causes the node to shut down or crash. Most of the systems reviewed are capable of automatic node fail-over without intervention from the cluster administration staff.

REFERENCES

[1] Understanding fault-tolerant distributed systems http://www.portal.acm.org/ citation.cfm?id=102801

[2] http://linux-ha.org/

[3] http://www.sun.com/software/white-papers/wpsunclusters.

[4] http://now.cs.berkeley.edu/Glunix/glunix.html

[5] http://h71000.www7.hp.com/availability/index.html.

[6] http://www.sco.com

[7] http://www.rs6000.ibm.com/sp.html

[8] " Sankhya Cluster Mannual ", Taurus India (Pvt) Ltd, Bangalore,India.