Computing Clusters

Submitted by: Farniba Khan

farniba.khan@student.lut.fi

Course Title: Computing clusters, grids and clouds

Supervised by: Andrey Y. Shevel

Date: 06 June 2017

ITMO University, Saint-Petersburg, Russia.

TABLE OF CONTENT

INTRODUCTION

A computing cluster is a parallel or distributed computer system, which consists of a collection of interconnected stand-alone computers working together as a single integrated computing resource.
Basically, it usually consists of two or more servers, under one administrator. Users see the cluster as a united resource for the service but not as a group of servers. In general, computing clusters consist of

- Group of computers and servers which are connected together and that act like a single system.
- Each system called a Node.
- Node contain one or more Processor, Ram, Hard disk and LAN card.
- Nodes work in Parallel.
- We can increase performance by adding more nodes into the cluster.

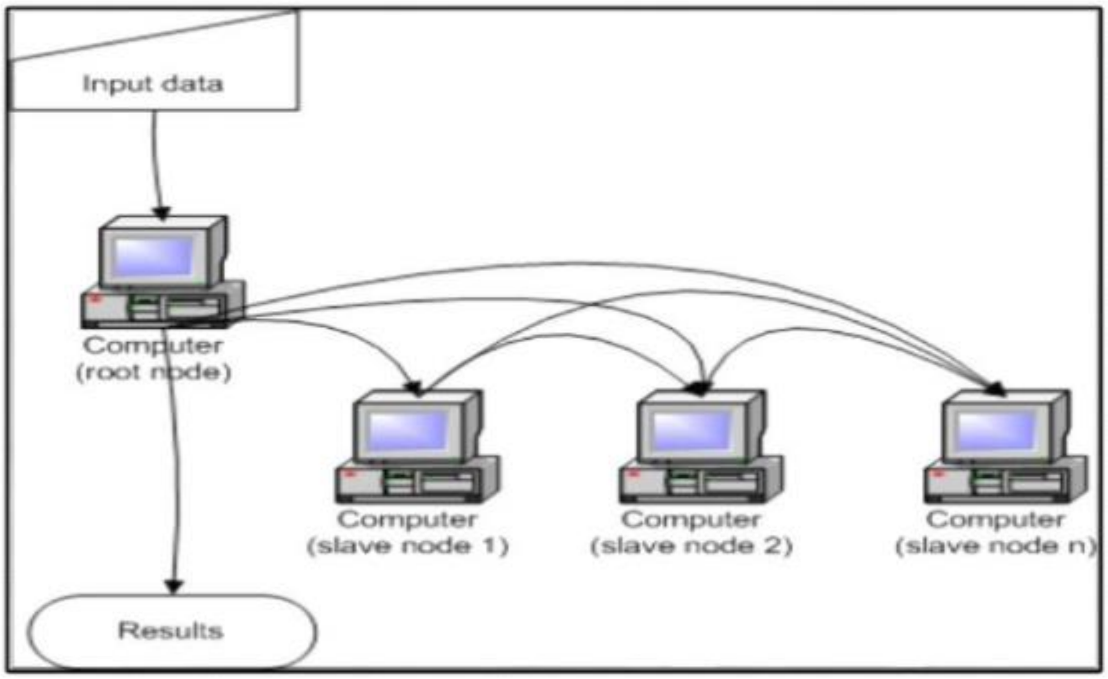

Fig: Simple layout of cluster computing

Above figure describes a simple layout of cluster computing. It consists of one root node and many slave nodes. The incoming requests from the user first comes to the root node and therefore the root node or root computer distributes the total request among all the slave nodes.

The major objective in the cluster is utilizing a group of processing nodes to complete the assigned job in a minimum time by working cooperatively. The main strategy to achieve such objective is by transferring the extra loads from busy nodes to idle nodes.

HISTORY

The first inspiration for cluster computing was developed in the 1960s by IBM as an alternative of linking large mainframes to provide a more cost effective form of commercial parallelism.
The first commodity clustering product was ARCnet, developed by Datapoint in 1977.
The original PC cluster project, also called Beowulf project, was started at the Center of Excellence in Space Data and Information Sciences NASA in early 1994. It is a system which usually consists of one master or server node, and one or more client nodes connected together via Ethernet
Microsoft, Sun Microsystems and other leading hardware and software companies offer clustering packages.

CLASSIFICATION OF CLUSTERS:

- ### High Availability Clusters(HA):

These clusters are designed to provide uninterrupted availability of services to the end-user. With a high-availability cluster, nodes can be taken out-of-service for maintenance or repairs. Additionally, if a node fails, the service can be restored without affecting the availability of the services provided by the cluster.
High-availability clusters implementations are best for mission-critical applications or databases, mail, file and print, web, or application servers.

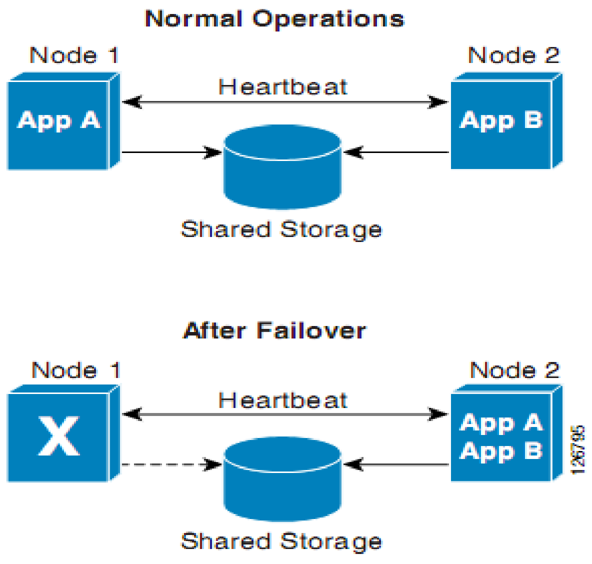

Fig: Failover Clusters

- Load Balancing Clusters:

This type of cluster distributes incoming requests for resources among multiple nodes running the same programs or having the same content. Every node in the cluster is capable to handle requests for the same content or application. If a node fails, requests are redistributed between the remaining available nodes. This type of distribution is typically seen in a web-hosting environment. These types of clusters are commonly used with busy file transfer protocol and web servers with large client base.

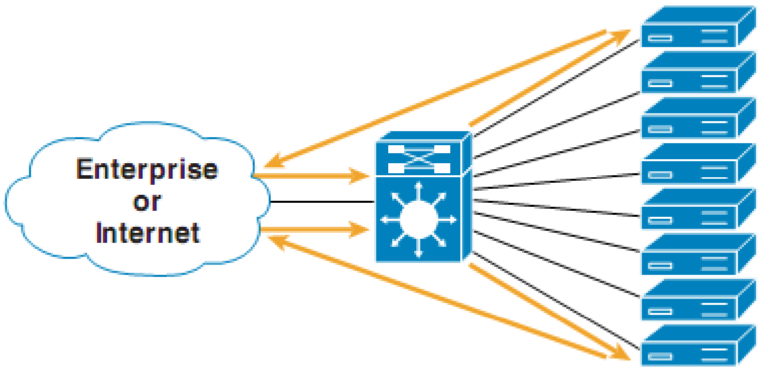

Fig: Load balancing clusters

- High Performance Clusters(HPC):

These clusters are designed to exploit the parallel processing power of multiple nodes. Applications like data mining, simulations, parallel processing, weather modeling etc. are used for HPC cluster. One of the example of this cluster is Beowulf cluster.

COMPONENTS:

The key components of a cluster include multiple standalone computers (PCs, Workstations, or SMPs), operating systems, high-performance interconnects, middleware, parallel programming environments, and applications.

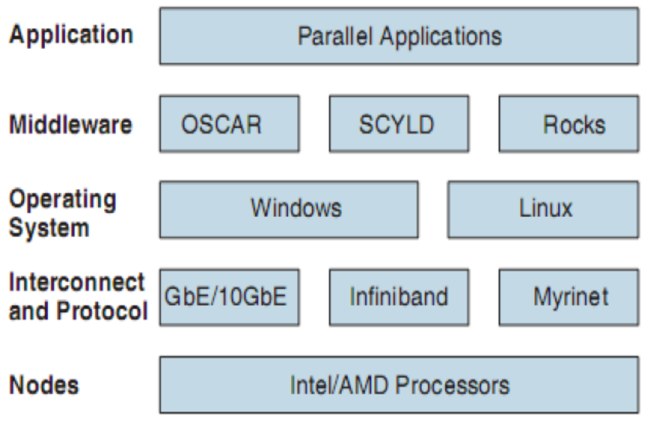

Fig: Components of computing clusters

- Application

It includes all the various applications either sequential and parallel applications. These applications run in parallel. These includes various query running on different nodes of the cluster. This can be said as the input part of the cluster component.

- Middleware

These are software packages which interacts the user with the operating system for the cluster computing. In other words, we can say that these are the layers of software between applications and operating system. Middleware provides various services required by an application to function correctly.
Node participation in the cluster falls into one of two responsibilities: master (or head) node and compute (or slave) nodes. The master node is the unique server in cluster systems.
It is responsible for running the file system and serves as the key system for clustering middleware to route processes, duties, and monitor the health and status of each slave node.
A compute (or slave) node within a cluster provides the cluster a computing and data storage capability. These nodes are derived from fully operational, standalone computers that are typically marketed as desktop or server systems that, as such, are off-the-shelf commodity systems.

The software that are used as middleware are:

OSCAR
Features:
_ Image based Installation.
_ Supported by Red Hat 9.0 and Mandrake 9.0.

_ Processors supported: x86, Itanium (in beta).
_ Interconnects: Ethernet, Myrinet.
_ Diskless support in development.
_ Opteron support in development.
_ High-availability support in alpha testing.

SCYLD
Features:
_ Commercial distribution.
_ Single system image design.
_ Processors: x86 and Opteron.
_ Interconnects: Ethernet and Infiniband.
_ MPI and PVM.
_ Diskful and diskless support.

Rocks
Features:
_ Processors: x86, Opteron, Itanium.
_ Interconnects: Ethernet and Myrinet.
_ Compute node management via Red Hat's kickstart mechanism.
_ Diskfull only.
_ Cluster on CD.

• Operating System

Clusters can be supported by various operating systems which includes Windows, Linux.etc.

• Interconnect

Interconnection between the various nodes of the cluster system can be done using Ethernet, Myrinet etc. In case of small cluster system these and be connected with the help of simple switches.

• Nodes

Nodes of the cluster system implies about the different computers that are connected. All of these processors can be of Intels or AMD 64 bit.

ARCHITECTURE:

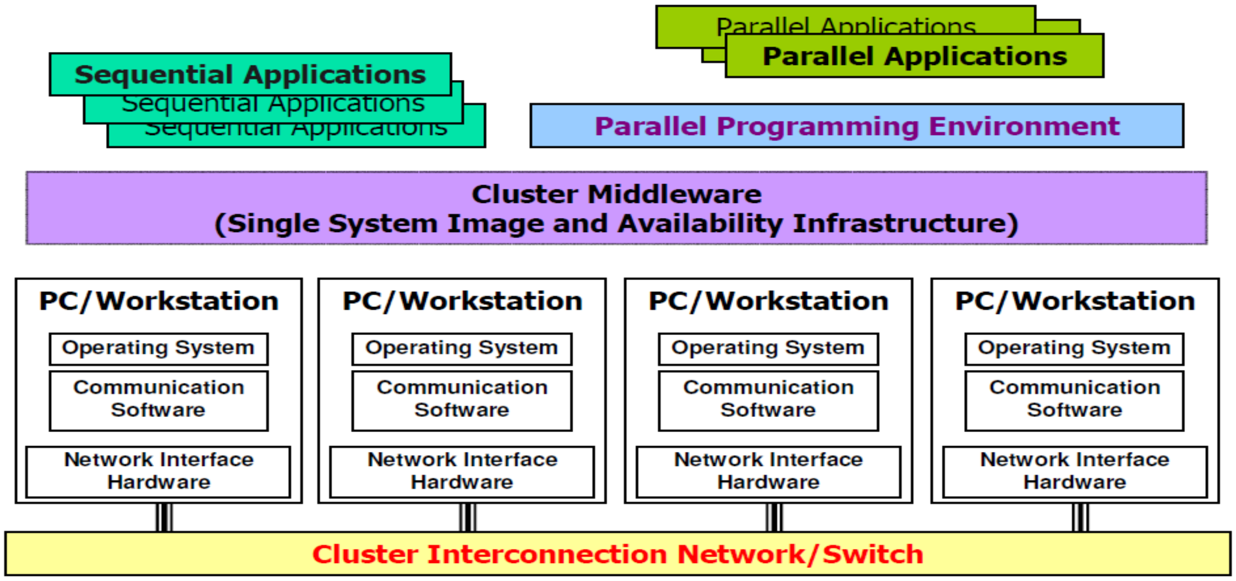

Fig: Overview of Computing Clusters

For the architecture, from the bottom is the Layer 2 level devices such as switches which is connected to several PCs or Workstations that have network interface hardware, communication software and operating systems running on those. These nodes or computers are run on middleware that are software packages which provides single system image and availability infrastructure. Finally, these nodes as clusters run on sequential or parallel applications.

Single System Image (SSI)

The Single System Image (SSI) represents the view of a distributed system as a single unified computing resource. This provides better usability for the users as it hides the complexities of the underlying distributed and heterogeneous nature of clusters from them. SSI can be established through one or several mechanisms implemented at various levels of abstraction in the cluster architecture: hardware, operating system, middleware, and applications.
The design goals for SSI cluster-based systems focus on complete transparency of resource management, scalable performance, and system availability in supporting user applications. Key SSI attributes that are generally considered desirable includes point of entry, user interface, process space, memory space, I/O space, file hierarchy, virtual networking, job management system, and control point and management.

The table below summarizes how SSI can be achieved at different levels of abstraction with examples.

Hardware	**Memory Channel – Distributed Shared Memory**
Operating Systems	MOSIX- Solaris MC-UnixWare
Middleware	Condor-Loadleveler - Load Share Facility (LSF)- Open Portable BatchSystem(OpenPBS) – Sun Grid Engine (SGE)- Libra
Application	PARMON-Linux Virtual Server- Problem Solving Environments
Programming	Linda- JavaSpaces-Message Queues- Parameter Sweep-Parallel Virtual Machine(PVM)-JavaGroups- Message Passing Interface(MPI)

APPLICATIONS:

- ### Compute Intensive Applications

Compute intensive is a term that applies to any computer application that demands a lot of computation cycles (for example, scientific applications such as meteorological prediction). These types of applications are very sensitive to end-to-end message latency. This latency sensitivity is caused by either the processors having to wait for instruction messages, or if transmitting results data between nodes takes longer. In general, the more time spent idle waiting for an instruction or for results data, the longer it takes to complete the application.

- ### Data or I/O Intensive Applications

Data intensive is a term that applies to any application that has high demands of attached storage facilities. Performance of many of these applications is impacted by the quality of the I/O mechanisms supported by current cluster architectures, the bandwidth available for network attached storage, and, in some cases, the performance of the underlying network components at both Layer 2 and 3.
Data-intensive applications can be found in the area of data mining, image processing, and genome and protein science applications. The movement to parallel I/O systems continues to occur to improve the I/O performance for many of these applications.

- ## Transaction Intensive Applications

Transaction intensive is a term that applies to any application that has a high-level of interactive transactions between an application resource and the cluster resources. Many financial, banking, human resource, and web-based applications fall into this category. There are three main care about for cluster applications: message latency, CPU utilization, and throughput. Each of these plays an important part in improving or impeding application performance. This section describes each of these issues and their associated impact on application performance.

- ## Examples:

➢ ### Google Search Engine.

A widely used search engine, Google uses cluster computing to meet the huge quantity of worldwide search requests that comprise of a peak of thousands of queries per second. A single Google query needs to use billions of processing cycles and access a few hundred megabytes of data in order to return satisfactory search results.

Google uses cluster computing as its solution to the high demand of system resources since clusters have better price-performance ratios than alternative high performance computing platforms, and use less electrical power. Google focuses on 2 important design factors: reliability and request throughput.
Google is able to achieve reliability at the software level so that a reliable computing infrastructure can be constructed on clusters of 15,000 commodity PCs distributed worldwide. The services for Google are also replicated across multiple machines in the clusters to provide the necessary availability. Google maximizes overall request throughput by performing parallel execution of individual search requests. This means that more search requests can be completed within a specific time interval.

A typical Google search consists of the following operations:

1. An Internet user enters a query at the Google webpage.
2. The web browser searches for the Internet Protocol (IP) address via the www.google.com Domain Name Server (DNS).
3. Google uses a DNS-based load balancing system that maps the query to a cluster that is geographically nearest to the user to minimize network communication delay time. The IP address of the selected cluster is returned.
4. The web browser then sends the search request in Hypertext Transport Protocol (HTTP) format to the selected cluster at the specified IP address.
5. The selected cluster then processes the query locally.
6. A hardware-based load balancer in the cluster monitors the available set of Google Web Servers (GWSs) in the cluster and distributes the requests evenly within the cluster.

7. A GWS machine receives the request, coordinates the query execution and sends the search result back to the user's browser.

- ### Petroleum Reservoir Simulation

This application demands intensive computations to simulate geological and physical models.

- ### Protein Explorer

This application provides Molecular dynamics simulations.

- ### Earthquake Simulation

This simulation is used to model and forecast strong ground motion during earthquakes.

- ### Image Rendering

This clustering application uses a swapping mechanism to manage datasets that are too large to load into the available texture memory, resulting in low performance and interactivity.

BEOWULF CLUSTER

Beowulf cluster uses parallel processing across multiple computers to create cheap and powerful supercomputers. For clustering, at least two computers with Linux based distribution with Ethernet segments, Myrinet, Infiniband can be used. It allows Operating System to run on every node and still allow parallel processing.

A cluster has two types of computers, a master computer, and node computers.
When a large problem or set of data is given to a Beowulf cluster, the master computer first runs a program that breaks the problem into small discrete pieces; it then sends a piece to each node to compute. As nodes finish their tasks, the master computer continually sends more pieces to them until the entire problem has been computed.

REFERENCES

http://www.cloudbus.org/papers/ic_cluster.pdf
http://www2.sscc.ru/News/Prezent/final-White_paper.pdf
https://www.slideshare.net/poojakhatana1/cluster-computing-28886643?next_slideshow=4
https://www.slideshare.net/nikhil9177/cluster-computing?next_slideshow=2
https://www.slideshare.net/Kajal_Thakkar/cluster-computing-59182598?next_slideshow=3
https://www.techopedia.com/definition/6581/computer-cluster
https://www.slideshare.net/rajamohd2/cluster-computing-56275811
http://www.exforsys.com/tutorials/clustering/clustering-security.html