Fall Back Tolerance for Server Farms

High Performance Networking

COEN 335

Winter 2017

Prajakta Deosthali

W1185340

pdeosthali@scu.edu

Audience

This report is for students learning about high performance networking. It is also written for professionals who are working in networking field dealing with servers, reliability engineering, various high availability systems and platforms.

This report is also suitable for enthusiasts of server farms, those who are interested in knowing more about how all the websites, applications provide continuous availability and reliable operations round the clock.

The readers are expected to have basic understanding of computers, networks. They should know how the internet works and fundamentals of client-server model.

This report covers information about different types of servers, server farms, and their role in today's connected world. It provides details about reliability, availability of the servers. It describes various ways by which these servers are made fault tolerant, how their performance is measured and monitored.

Table of Contents

Table of Figures

Introduction

Chapter 1 contains information about servers. It describes the functions and different types of servers. It briefly explains client-server model. It also provides information about hardware and software characteristics of servers.

Chapter 2 provides details about server farms, their utilities in today's world. This chapter talks about various performance metrics in the server farms. It also details about their impact on economy and environment.

Chapter 3 describes fault, fault tolerance, fall back, reliability and availability. It allows how these performance metrics are defined, measured and monitored.

Chapter 4 provides information about how availability and reliability is ensured in servers. It enlists various techniques used in practice, their advantages and disadvantages.

Chapter 1

Servers

1.1 Server

A server is a device which provides some sort of service to a set of clients. It is daemon process running on a computer. The word server is used interchangeably between the computer on which this process is running and the process itself.

There are many types of servers in use today. e.g. web server, application server, mail server, file server etc.

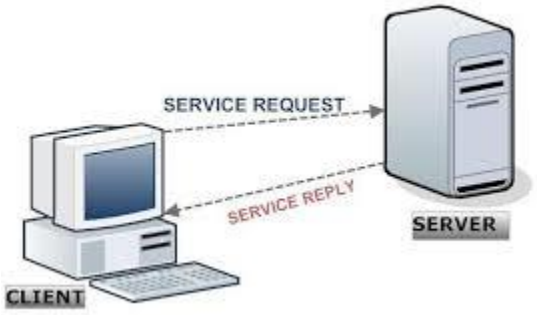

Fig 1.1 Client server architecture

In terms of computer networks, a server is one of the two main components of client-server model.

1.2 Client server model

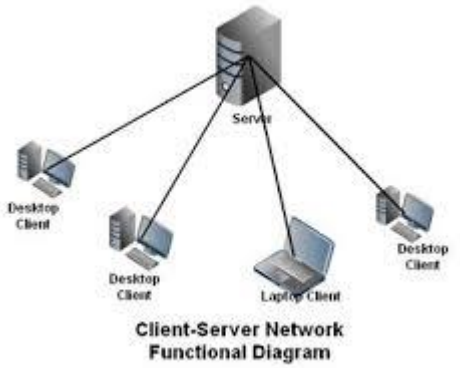

Fig 1.2 Client server network

The client-server model has two components, client and server. The client and server are both computer processes running on two different computers(in most cases). The client sends a request to the server; server processes the request and sends a response back to the client.

The client-server relationship is many-to-many type. One client can request many servers for different services. Server, in turn, can fulfill requests of many clients.

1.3 Types of servers

There are many types of servers depending on the functionality and requests they serve. Following are the more common types.

1.3.1 Application Server

An application server is part of web architecture. It provides facility to host and run web applications. This type of server acts as an intermediary between a user of a website and the database of that website.

Fig 1.3 Web server

Examples are JBoss, WebSphere Application Server.

1.3.2 Computing Server

This type of server provides additional and external computing power. When the client machine does not have enough computation ability, this can be requested to do additional calculations. A supercomputer is a type of computing server. It is arranged in grid computing fashion or a cluster fashion. This type of server uses large amount of energy and high speed networking components to communicate with participating computers.

Examples are Cray-1, Blue Gene.

1.3.3 Database server

Fig 1.4 Database server

This is a commonly used server. This server is part of almost every website, web app. It is also used in enterprises and businesses to store vast amount of data like files, folders, images, videos, spreadsheets etc.

Examples are Oracle, DB2, Microsoft SQL Server.

1.3.4 File server

Fig 1.5 File server

This server stores large number of files or documents. It acts as a central location for accessing shared files through the clients connected to the network. This type of server is generally a component of multiuser systems and many users having different privileges and security levels can access the files on it.

File servers are widely used in institutions like schools, colleges and other small business organizations where the servers are connected using LAN. In addition to that, file servers connected to internet can be accessed using FTP.

1.3.5 Mail server

Fig 1.6 Mail server

This type of server acts as a post office for email communications. It acts as Mail Transfer Agent, i.e. transfers electronic mail between two communicating parties. It follows a protocol specifically designed for email communications, called Simple Mail Transfer Protocol.

1.3.6 Proxy server

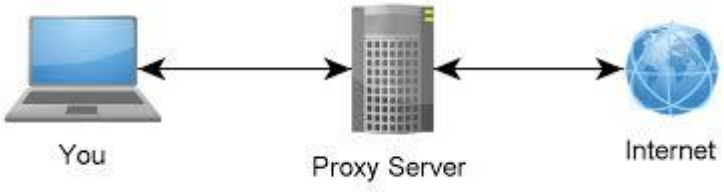

Fig 1.7 Proxy server

t is a type of server that acts as an intermediary between a client and any other server, provides fast, secure and more efficient access to the resources. There are different types of proxies depending on specific functionalities like forward, reverse proxy or open, closed proxies. A proxy is useful for caching, load balancing. It uses techniques like compression to reduce the load time and renders the content faster.

1.3.7 Web server

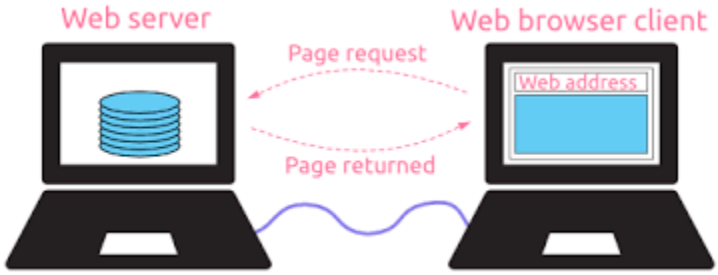

Fig 1.8 Web server

It is used to store, process and deliver web pages. It is connected to the internet and all the clients access the web server using HTTP protocol. The web server provides static as well as dynamic HTML content. It supports various scripting language for dynamic page generation like ASP, PHP etc.

The default client for any web server is a web browser like FireFox, Chrome etc.

1.3.8 Game server

This type of server makes multiplayer games possible. It acts as the source of truth for the state of the game, provides all the clients with internal state. It also processes each client's request, state in the game and overall progress.

1.4 Hardware of servers

As there are different types of servers the hardware used varies depending on the purpose and usage.

Fig 1.9 Rack mounted server

In most of the cases, servers do not have a GUI, these are accessed and controlled using a central console, a CLI or using a web browser. Servers are designed to be powered up and down, rebooted and configured remotely through these means.

The server is expected to run for long duration without any manual external monitoring or intervention. Due to this requirement, hardware reliability is an important criterion. Servers are designed with components having low failure rates, high availability and reliability.

Servers support built-in hardware redundancy, it includes dual power supplies, duplicate memory components, other important I/O drivers.

As servers run continuously, they are provided with uninterrupted power supply. They are also provided with more powerful and efficient cooling mechanisms.

Server casing are generally flat and wide, as the server is mounted on the rack.

1.5 Monitoring the servers

A server should be and is continuously monitored using various tools provided by the server manufacturers. These tools track various metrics like CPU usage, number of requests, disk usage, disk I/O, network traffic etc.

These metrics are recorded and updated at regular intervals. The intervals and metrics are determined based on the type of server, usage and functionality. The server

administrator sets thresholds for all these parameters that help gain and maintain better control over the performance and uptime of the server; tuning and troubleshooting is done based on the fluctuations in the load and usage of the server.

The servers must provide reliability and high availability. They must be able to perform reasonably well in spite of fluctuations in load.

Chapter 2

Server Farms

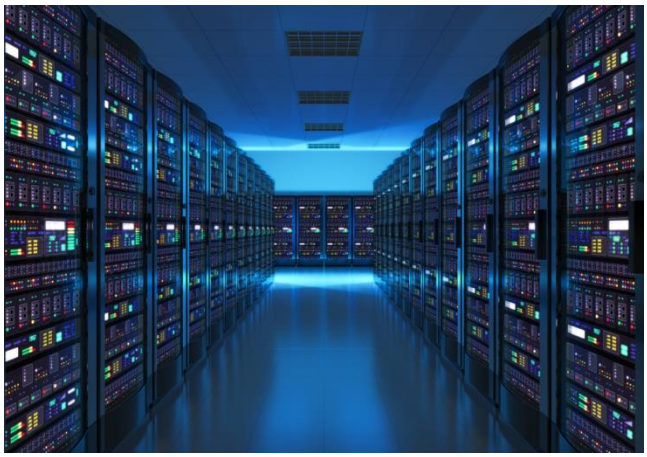

Fig 2.1 Typical server room

2.1. Server Farm

A server farm is also known as server cluster. It is a collection of two or more servers. A server farm is generally maintained by a single organization. A server farm may consist of thousands of servers, it requires lot of energy to keep these farms running and keep them cool. Server farms pose a financial as well as environmental burden.

Server farms have backup servers to provide better availability. These clusters are usually located near switches, routers and other networking components, to enable efficient and faster communication between different servers and users.

The computing servers that was discussed in previous chapter can be constructed as a server cluster, where a large number of high-speed servers are connected together using high speed networks.

Fig 2.2 Facebook Data center in Arctic

2.2 Capacity of Server farms

The server farms consist of thousands of servers, mounted on racks. These servers are connected using high speed networking components. The design of the farm is done keeping automatic failover and redundancy in mind. The server farm must provide reliability and high availability.

2.3 Performance of server farms

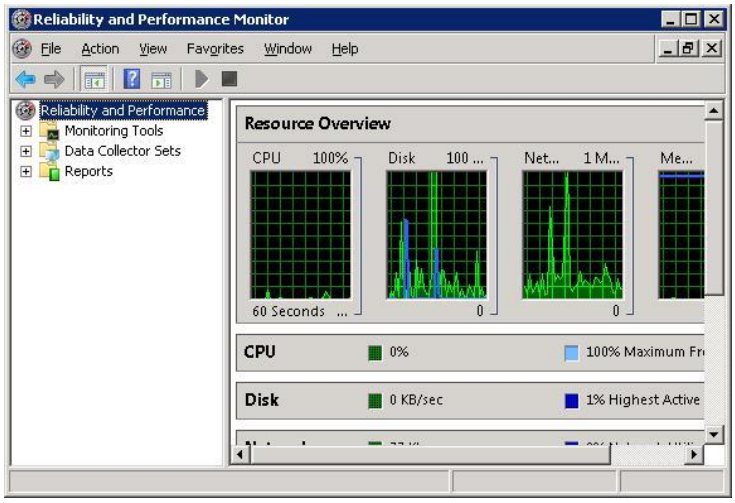

Fig 2.3 Monitoring performance

As the server farms consume large amount of electricity and are continuously up, lot of efforts are required to keep these centers cool. The most important performance metrics due to this is performance per watt. The power used by each rack is measured at power distribution unit. Total power consumption is then calculated based on the number of racks. The measure of performance could be number of operations (FLOPS) or number of requests served depending on the application of the server.

2.4 Impact on environment

Due to the amount of cooling required in a server farm, the location of the farm is of paramount importance. It is estimated that server farms consume 1.5% of world's total electricity.

More and more companies are shifting their data centers or server farms in the colder, arctic regions. Countries like Canada, Greenland are being targeted for building bigger and more energy-efficient data centers.

We have understood what servers are and how server farms are built. Let us now analyze and describe fall back tolerance.

Chapter 3

Fault Tolerance

The server farms and servers are important components in today's digital ear. Most of the web applications claim to provide 24X7 access to the users. For many organizations, it is imperative that the website, web app, mobile app or any other service provides efficient and fast access under any magnitude of user load.

As servers have these stringent performance constraints, fault tolerance is an important way to achieve high availability.

Let us understand more about fault and fault tolerance.

3.1 Fault

Fault – It is an error or a failure. A fault in a computer system can be of two types:

1. Software Fault

2. Hardware Fault

3.1.1 Software Fault

In case of software fault, the program goes out of intended or assigned memory boundary.

3.1.2 Hardware Fault

One of the hardware components of the computer system, fails, malfunctions, stops working or shows erratic behavior.

3.2 Fault isolation

It is the process of isolating or detecting the software or hardware component that caused the fault or error to occur in the system.

In case of hardware fault, the fault can be isolated by building various test circuits to verify the functionality of each component. For networks, network analyzers are used

to continuously monitor the network, these devices send alerts in case of fault occurrence.

For software fault, the process is similar wherein each module is tested separately. It is advised to divide the program into different modules or components, which makes it easier to test each part of the software separately.

3.3 Fail-safe

A system is designed to be fail-safe. In the event of a specific type of failure, the system is designed to respond in a way that will cause minimal harm to other components, equipment of the system.

In a fail-safe system, a failure is expected to happen. It is not a prevention mechanism. The system is designed with safeguards to deal with such failures.

Failing badly or failing well describes how a system reacts to failure, it could be software or hardware failure.

A system failing badly leads to catastrophic consequences for the overall system. Examples of such scenario is having single point of failure. Keeping a single copy of data in a central place.

On the other hand, a system that fails well is the one that can compartmentalize failure. By compartmentalizing it, the failure can be easily contained and restricted to a small area of the system. This approach allows graceful exit.

3.4 Graceful Exit

Graceful exit is when the system detects an error or a failure in any of its components and exits with minimal damage to the overall system, in a controlled and predictable way.

3.3 Fault Tolerance

It is the ability of a system to continue operating in the event of failure of some of its components. It is the quality of graceful degradation. The operational capability of a system should degrade proportional to the severity of the failure.

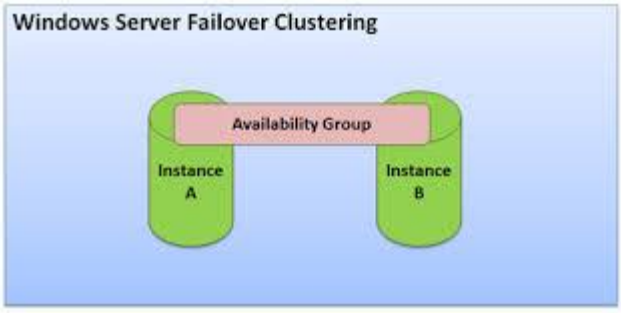

Fig 3.1 Availability

For computers, in case some components fail, it is expected that the system will be able to perform with reduced throughput and slower response. In fault tolerant systems, the programs are expected to continue even in presence of error, exception, incorrect data or any other failure condition.

The ability of a system to maintain functionality even when some parts of the system malfunction is known as **graceful degradation**.

The basic characteristics of a fault tolerant system are as follows -

1. There should not be a single point of failure.

2. The system should be able to operate during the repair process. This can be achieved by providing a secondary system or adding a level of redundancy to the system.

3. It should be easy to detect and isolate the point of failure, be it hardware or software. This can be achieved by various failure detection techniques.

4. The fault once detected, should be contained. It must be prevented from affecting other working components of the system.

Any system having high availability as the major performance criteria must be able to provide fault tolerance. There are various ways to achieve fault tolerance. In cases where the consequences of system failure are catastrophic, the system must be able to fall back to safe mode.

A system can be analyzed to anticipate failures of components that are likely to occur. Based on this analysis, prevention mechanisms are used to avoid failures. In some

cases, it is cost prohibitive to implement such prevention techniques. For such scenarios, redundancy is the solution.

In most of the fault tolerance system, redundancy is used to provide high availability.

3.5 Redundancy

This is one of the ways to provide fault tolerance. It means providing backup options or secondary systems to fall back on in case the primary system fails. It is duplication of critical components. This is done to increase reliability of the system.

Redundancy, if not properly designed, can make the system more complex.

Following are the common ways of achieving redundancy.

3.5.1 Space redundancy

It is classified into hardware, software and other types depending on the type of system. It involves inclusion of multiple counts of a different software as well as hardware components to ensure fault free operation.

3.5.2 Time redundancy

In this type of redundancy, the processing on data is done again and stored. The simplest example is creating a checkpoint in a distributed data store.

3.5.3 Hardware redundancy

Having replicas of hardware components e.g. servers, Name node in Hadoop ecosystem.

3.5.4 Information redundancy

It is achieved using error detection and correction codes while transmitting the signals in the network.

3.5.6 Functions of redundancy

3.5.6.1 Passive redundancy

It uses additional hardware components to provide graceful degradation in case of failure. Reduction in performance levels is expected in this type.

3.5.6.2 Active redundancy

In this case, the performance is continuously monitored. If there is a visible decline in any of the hardware components, that component is automatically reconfigured.

3.6 Fall Back

Fall Back means to retreat or withdraw. It is a contingency option if and when the preferred option suffers from a fault.

Fallback system is a system of hardware or software components which act as a backup to the primary system. In case of failure of the primary system, the functionality falls back on the secondary or backup system. The services of primary are provided by secondary system as the primary is being restored back to health.

Chapter 4

Fault Tolerance in Server farms

4.1 Fault and servers

Fault tolerant is an important design consideration in server farms, as the applications require the data center to be available round the clock.

As the servers are used for multitudes of purposes, the holy grail of fault tolerance can be achieved in various ways. The application, requirement and critical component can differ considerably for each server farm. Based on this following are some of the techniques used to provide fault tolerance.

4.2 Intra-server redundancy

One of the more common ways, already discussed in previous chapter is redundancy. It is used as a mechanism to resolve intra-server failure. A error detection system is used to find the malfunctioning CPU. One detected that particular server is removed from operation, replaced with another redundant machine.

4.3 Data mirroring technique

For a server farm responsible for storage and easy access of files of the users, availability is provided by replication. In this scenario, a first and second server is established. The client sends a request to write a file to the first server. This server provides the same data to the second server; this is achieved through a specific asynchronous mechanism. When the client makes the request again for the same resource, either first or second server can fulfill the request. This technique avoids the single point of failure by providing Dual Mode redundancy. This architecture is useful in Network File System (NFS) servers.

4.4 Temporal redundancy

As explained in previous chapter, in temporal redundancy a checkpoint is created. In the event of a failure, the application can be re-run from the checkpoint. The checkpoint can be created at two levels based on the type of service: software stack level and application type level.

Various libraries are available to implement software stack level checkpoint. For the application type checkpoint creation, it needs to be taken care of by the programming team of the application.

4.5 Spatial redundancy

It involves duplication of the data or recalculation of the task. This technique is common in database servers.

4.6 Load balancing techniques

In server farms specifically used for distributed computing, most of the fault occur due to unequal, excessive utilization of CPU. To avoid such faults, better load balancing algorithms are used. These algorithms divide the load evenly to make sure that non of the servers is performing more calculations than intended.

These algorithms also locate the computation of the same task in co-located server allowing faster communication.

4.7 Resilience

There has been tremendous growth in the hardware manufacturing processes. This has reduced the number of defects caused due to hardware aspects of the components. To provide better availability and reliability, it is important to build a system that can successfully detect design bugs. The efforts are underway to make the systems more resilient.

Conclusion and Future Aspects

The server farms are the backbone of today's digital world. Their reliability and availability is of paramount important to users as well as business organizations.

Many techniques are used today to provide fault tolerance. As the amount of data grows further, the complexity of applications will also grow exponentially. The techniques used today will not be enough to provide the same level of reliability in that case.

In future, we should expect more complex algorithms for load balancing techniques, better redundancy and replication strategies.

In all these aspects, the reliance of server farms on electrical power and need of cooling mechanisms will remain the biggest constraint.

Acronyms

LAN – Local Area Network

MTA – Mail Transfer Agent

SMTP – Simple Mail Transfer Protocol

FLOPS – Floating Operations Per Second

References

https://en.wikipedia.org/wiki/Server_farm

https://en.wikipedia.org/wiki/Category:Cluster_computing

https://en.wikipedia.org/wiki/List_of_cluster_management_software
https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis

https://en.wikipedia.org/wiki/Fail-safe

https://users.ece.cmu.edu/~koopman/des_s99/sw_fault_tolerance/

http://www.pcmag.com/encyclopedia/term/42984/fallback-system

http://www.pcmag.com/encyclopedia/term/43036/fault-tolerant

http://www.drj.com/articles/online-exclusive/fault-tolerance-in-virtualized-data-centers.html

https://en.wikipedia.org/wiki/Graceful_exit

https://en.wikipedia.org/wiki/Robustness_(computer_science)
https://www.google.com/patents/US6925645

https://www.google.com/patents/US5513314

http://ieeexplore.ieee.org.libproxy.scu.edu/document/7229841/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/7133528/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/7124879/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/920587/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/5502186/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/7124879/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/5307417/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/1015612/

http://ieeexplore.ieee.org.libproxy.scu.edu/document/4698116/