International Journal on Emerging Technologies (Special Issue NCETST-2017) **8**(1): 307-309(2017)
(Published by Research Trend, Website: www.researchtrend.net*)*

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

An analysis of Distributed System Over IP Network

Aakansha Adhikari[1], Praney Raghuvanshi[1] and Dr. Sadhana Rana[2]
[1]Student- Computer Science Engineering, Amrapali Institute-AITS, Haldwani, (U.K.), INDIA
[2]Assistant Professor, Computer Science Engineering, Amrapali Institute-AITS, Haldwani, (U.K.), INDIA

ABSTRACT: In this paper, we have made an attempt to analyze the performance of a distributed system over an IP network while keeping in mind the various factors affecting it because while managing networking traffic in distributed system, it becomes necessary for a network administrator to maintain the quality of service (QoS). The main factors affecting network quality are bandwidth, reliability, throughput, delay, jitter, QoS, etc. There are many troubleshooting softwares to overcome the problems regarding network quality. This paper will help to make administration easy in distributed computing environment.

Keywords: ADSOI, DSOI and ADS

I. INTRODUCTION

Distributed computing- Distributed computing is an environment in which computers are connected together in order to achieve a particular set of goals. It has following features-

- ✓ No common or global clock
- ✓ No shared memory
- ✓ Autonomous and heterogeneity

Distributed systems- A distributed system can be called as a piece of software which makes sure that a group of independent computers appears to users as a single coherent system. It works as if it is a single large computer. It helps in sharing the resources.

The key goals of distributed systems are-

- ✓ Performance: It improves the performance.
- ✓ Transparency: It shows the image of a single system without concealing the details of the location, access, migration, concurrency, failure, relocation, persistence and resources to the users.
- ✓ Openness: Configuration and modification of network becomes easier.
- ✓ Reliability: A distributed system is highly secure, consistent and has a high capability of masking errors.
- ✓ Scalability: Distributed systems are expandable with respect to geography, administration or size.

II. MODELS OF DISTRIBUTED SYSTEMS

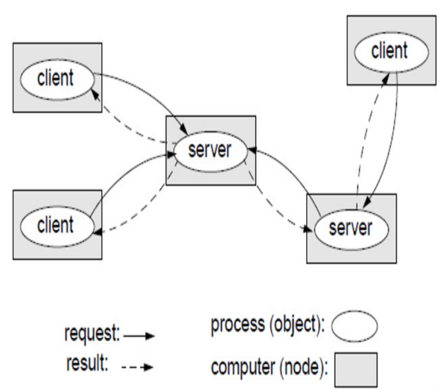
Distributed System Models are as follows: 1.ArchitecturalModels 2. Interaction Models 3. Fault Models

A. Architectural Models

Architectural model describes responsibilities distributed between system components and how are these components placed.

a) Client-server model

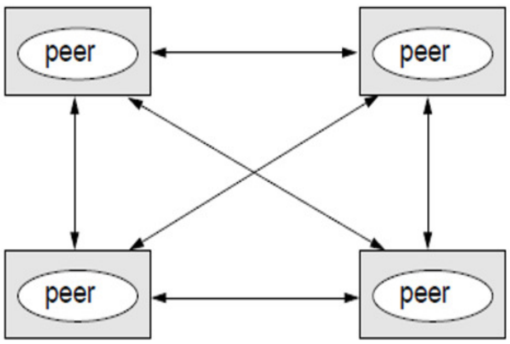
- ➢ The system is structured as a set of processes, called servers, that offer services to the users or clients.
  - The client-server model is based on a request/reply protocol, implemented with send/receive primitives or using remote procedure calls (RPC) or remote method invocation (RMI).
  - The client sends a request (invocation) message to the server and asks for some service.
  - The server fulfils the request and returns a result or an error code if the work could not be performed.



b) Peer-to-peer

- ➢ All processes (objects) play similar role.

- Processes interact without particular distinction between clients and servers.
- The pattern of communication depends on the particular application.
- A large number of data objects are shared; any individual computer holds only a small part of the application database.
- Processing and communication loads for access to objects are distributed across many computers and access links.
- This is the most general and flexible model.



- It distributes shared resources widely.

B. Interaction Model

Interaction model are for handling time i.e. for process execution, message delivery, clock drifts etc.

Main features:

- Lower and upper bounds on execution time of processes can be set.
- Transmitted messages are received within a known bounded time.

C. Fault Models

- ➢ Failures can occur both in processes and communication channels. The reason can be both software and hardware faults.
- ➢ Fault models are needed in order to build systems with predictable behavior in case of faults (systems which are fault tolerant).
- ➢ Such a system will function according to the predictions, only as long as the real faults behave as defined by the "fault model".

III. PROS AND CONS OF DISTRIBUTED SYSTEM

Advantages-

- ✓ Give more performance than single system
- ✓ If one pc in distributed system malfunction or corrupts then other node or pc will take care of whole system.
- ✓ More resources can be easily expanded in terms of numbers.
- ✓ Resources can be shared on multiple pc's.

Disadvantages-

- ✓ Security problem due to sharing of resources.
- ✓ Messages can be lost in the network
- ✓ Bandwidth is the main problem.
- ✓ Overloading is another problem in distributed operating systems
- ✓ If there is a database connected on local system and many users access that database through remote or distributed way then performance becomes slow.
- ✓ The databases in network is difficult to administrate.

V. APPLICATIONS

- ✓ Telecommunication networks:
  - Telephone networks and cellular networks
  - Computer networks like Internet
  - Wireless sensor networks
  - Routing algorithms
- ✓ Network applications:
  - World wide web and peer-to-peer networks
  - Multiplayer online games and virtual reality communities
  - Network file systems
- ✓ Real-time process control:
  - Aircraft control system
  - Industrial control system
- ✓ Parallel computation:
  - Scientific computing like cluster computing and grid computing

VI. QUALITY OF SERVICE

Everybody needs a reliable Internet connection nowadays because everything depends on the worldwide web. With the overwhelming use of mobile phones, computers and other gadgets, it becomes necessary to check the performance of a system in order to maximize the benefits. Quality of service (QoS) has been receiving wide attention in distributed systems as these systems have a requirement that applications requesting for system resources must satisfy timing, reliability and security constraints as well as maintain application-specific standards. The important properties for a distributed compute network connections are-

1) Bandwidth- It is the rate of data transfer, bit rate or throughput, measured in bits per second (bit/s)

2) Delay- It specifies how long it takes for a bit of data to travel across the network from one node or endpoint to another.

3) Reliability- Reliability is an attribute of any computer-related component of a network that consistently performs according to its specifications.

4) Jitter- Jitter is defined as a variation in the delay of received packets.

VII. PRACTICAL APPROACH

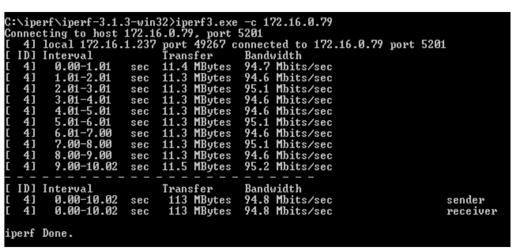
Among many available tools, we have used the **iperf** tool to analyze the performance of a distributed systems. Iperf is a network testing tool that can create TCP and UDP data streams and measure the throughput of a network that is carrying them.
Resource configuration- We used two systems connected in a LAN having access to the Internet to execute the performance analysis process. One acted as a server and another was made the client.

DESCRIPTION	PC1(server)	PC2(client)
Processor model		
Processor Mhz		
Memory	1GB	1GB
Operating System	Windows 7	Windows 7
Network MAC	Ethernet	Ethernet
Network link	100Mbps	100Mbps
IP Address	172.16.0.79	172.16.0.73
ISP	BSNL	BSNL

The following set of commands was used to measure various performance attributes-
For performance analysis on TCP traffic-
On server-
C:\iperf>iperf.exe -s -p 2000
Output-

Server listening on TCP port 2000
TCP window size: 64.0 KByte

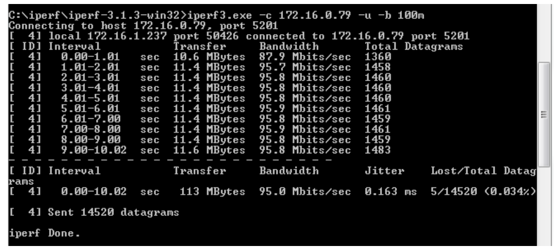
On client-



```
C:\iperf\iperf-3.1.3-win32>iperf3.exe -c 172.16.0.79
Connecting to host 172.16.0.79, port 5201
[  4] local 172.16.1.237 port 49267 connected to 172.16.0.79 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.01   sec  11.4 MBytes  94.7 Mbits/sec
[  4]   1.01-2.01   sec  11.3 MBytes  94.6 Mbits/sec
[  4]   2.01-3.01   sec  11.3 MBytes  95.1 Mbits/sec
[  4]   3.01-4.01   sec  11.3 MBytes  94.6 Mbits/sec
[  4]   4.01-5.01   sec  11.3 MBytes  94.6 Mbits/sec
[  4]   5.01-6.01   sec  11.3 MBytes  95.1 Mbits/sec
[  4]   6.01-7.00   sec  11.3 MBytes  94.6 Mbits/sec
[  4]   7.00-8.00   sec  11.3 MBytes  95.1 Mbits/sec
[  4]   8.00-9.00   sec  11.3 MBytes  94.6 Mbits/sec
[  4]   9.00-10.02  sec  11.5 MBytes  95.2 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-10.02  sec   113 MBytes  94.8 Mbits/sec                  sender
[  4]   0.00-10.02  sec   113 MBytes  94.8 Mbits/sec                  receiver

iperf Done.
```

For performance analysis on UDP traffic-
Server process is same as TCP traffic analysis. **On client-**
C:\iperf\iperf-3.1.3-win32>iperf3.exe –c 172.160.79 –u –b 100m



```
C:\iperf\iperf-3.1.3-win32>iperf3.exe -c 172.16.0.79 -u -b 100m
Connecting to host 172.16.0.79, port 5201
[  4] local 172.16.1.237 port 50426 connected to 172.16.0.79 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.01   sec  10.6 MBytes  87.9 Mbits/sec  1360
[  4]   1.01-2.01   sec  11.4 MBytes  95.7 Mbits/sec  1458
[  4]   2.01-3.01   sec  11.4 MBytes  95.8 Mbits/sec  1460
[  4]   3.01-4.01   sec  11.4 MBytes  95.8 Mbits/sec  1460
[  4]   4.01-5.01   sec  11.4 MBytes  95.8 Mbits/sec  1460
[  4]   5.01-6.01   sec  11.4 MBytes  95.9 Mbits/sec  1461
[  4]   6.01-7.00   sec  11.4 MBytes  95.8 Mbits/sec  1459
[  4]   7.00-8.00   sec  11.4 MBytes  95.9 Mbits/sec  1461
[  4]   8.00-9.00   sec  11.4 MBytes  95.8 Mbits/sec  1459
[  4]   9.00-10.02  sec  11.6 MBytes  95.8 Mbits/sec  1483
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  4]   0.00-10.02  sec   113 MBytes  95.0 Mbits/sec  0.163 ms  5/14520 (0.034%)

[  4] Sent 14520 datagrams

iperf Done.
```

VIII. CONCLUSION

We used client-server architecture for analyzing the TCP and UDP traffic control. we worked on throughput, delay, reliability and jitter for providing good quality of services to the administrator for handling the performance of distributed system network.
This paper focused on distributed system is beneficial for practical reasons. It may be more cost efficient to obtain the desired level of performance by using a cluster of several low-end computers. A distributed system can provide more liability than a non-distributed system, as there is a no single point of failure.

REFERENCES

[1]. Andrew S. Tanenbaum. Computer Networks, Fourth Edition. Prentice Hall, 2003.
[2]. Annabel Z. Dodd. The Essential Guide to Telecommunications, Second Edition. Prentice Hall PTR, 1999.
[3]. William Stallings. Local networks. ACM, 1984.
[4]. Mahbub Hassan and Raj Jain. High Performance TCP/IP Networking. Pearson Prentice Hall, 2004.
[5]. Classification of Distributed Systems Properties of Distributed Systems http://vega.cs.kent.edu/~mikhail/classes/aos.s02/l01DStaxonomy.PDF
[6]. what are distributed systems?? https://www.slideshare.net/Rupsee/distributed-systems-1701403
[7]. Distributed System Definition - What does Distributed System mean? https://www.techopedia.com/definition/18909/distributed-system
[8]. https://users.ece.cmu.edu/~koopman/des_s99/quality_of_service/
[9]. Bandwidth From Wikipedia, the free encyclopedia https://en.wikipedia.org/wiki/Bandwidth
[10]. Network delay From Wikipedia, the free encyclopedia https://en.wikipedia.org/wiki/Network_delay
[11]. Iperf From Wikipedia, the free encyclopedia https://en.wikipedia.org/wiki/Iperf