An Effective Congestion Control for TCP using Data-centered Network

A.Vijay Vasanth[1] R.Saranya[2] D.Dhivya Bharathi[3] S.N.Evangilin[4]

[1] Senior Assistant Professor, Christ College of Engineering and Technology , Puducherry
[2]UG Student, Christ College of Engineering and Technology , Puducherry
[3]UG Student, Christ College of Engineering and Technology , Puducherry
[4]UG Student, Christ College of Engineering and Technology , Puducherry

Abstract

Transport Control protocol (TCP) incast congestion happens in high bandwidth and low-latency network when multiple synchronized servers send data to the same receiver in parallel. For many data centered application the many-to-one traffic pattern is common. Hence TCP incast congestion may severely degrade the performance. To avoid incast congestion we focusing on Grid computing, Hashing technique and Fault Tolerance mechanism. The idea is to design an incast congestion on the receiver side. We strongly believe that grid computing is used for process allocation and scheduling the subserver. We using mobile agent to handle failure thus the specific part of failed file will be transmitted rather than whole part to be sent. To improve message integrity hashing technique in which CBU algorithm is used. To improve quality of service fault tolerance mechanism is used.

Index Terms - Data centered network, Incast congestion, TCP, Grid computing,Fault tolerance, hashing.

1.Introduction

Transport Control Protocol is one of the core protocol of the internet suite, and is so common that the entire suite is often called TCP/IP. TCP provide reliable, ordered, error checked delivery of the stream of octet between program running on computer connected to local area network, intranet or the public internet. It resides at the transport layer. TCP provide communication at an internet level between an application program and the Internet protocol. This is when an application program reside to send a large chunk of data across the internet using IP, instead of breaking the data into IP-sized pieces and issuing series of IP request, the software can issue an single request to the TCP and let TCP handles the IP details.

The Transport Control Protocol is widely used on the internet and generally works well. However the recent studies shown that TCP does not work well for many-to-one-traffic pattern on high-bandwidth and low latency network. Congestion occurs when synchronized servers under the same Gigabit Ethernet switch simultaneously send data to one receiver in parallel .Only after all connection have finished the data transmission can the next round be issued. Thus these connections are called barrier-synchronised.

The traffic and network condition in data-centered network creates the three preconditions for incast congestion. First, Data centered networks are well structured and layered to achieve high bandwidth and low latency, and the buffer size of top-of-rack Ethernet switch is usually small. Second, a recent measurement study showed that a barrier synchronized many-to-one traffic pattern is common in data centered network. Third ,the transmission data volume for such traffic pattern is usually small.

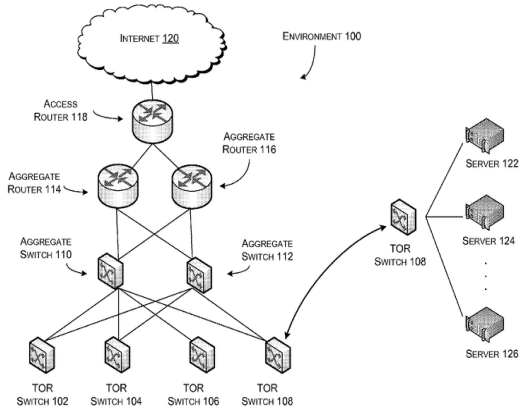

Fig 1: Architecture for the Incast congestion

This paper focuses on avoiding packet loss before incast congestion ,which is appealing than recovery after loss. Of course, recovery schemes can be complementary to congestion avoidance. The smaller the change we make the existing system, the better. The receiver side is a natural choice since it knows the throughput of all TCP connections and the available bandwidth. The receiver side can adjust the receiver window size of each TCP connection. so the aggregate burstiness of all synchronised senders are kept under control.

The root cause of TCP incast collapse is that the highly bursty traffic of multiple TCP connections overflows the Ethernet switch buffer in a short period of time, causing intense packet loss and thus TCP retransmission and timesout. Previous solution focused on either reducing the wait time for packet loss recovery with faster retransmission or controlling switch buffer occupation to avoid overflow.

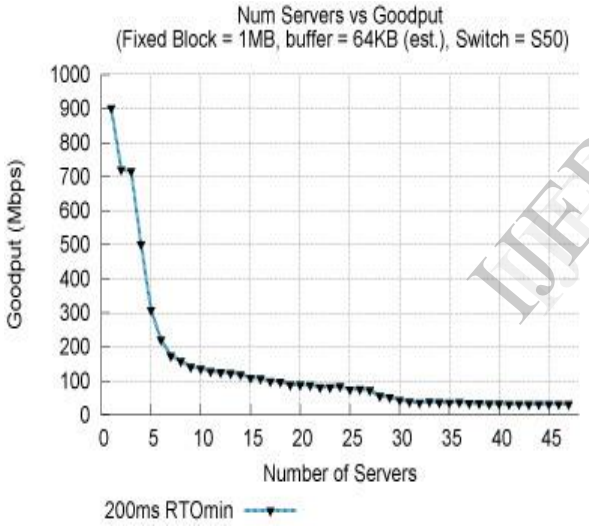

Fig 2: graph for number of servers and goodput

Our paper addresses the above challenges with a systematic approaches such as Grid Computing, Fault Tolerance, Hashing Techniques to avoid the congestion. In our proposed these approaches avoid the congestion. The grid computing is mainly for process allocation and scheduling the subservers .We use mobile agent to handle failures thus the specific part of failed will be transmitted rather than whole part to be sent. To ensures message integrity hashing technique is used. To improve quality of service we use fault tolerance system.

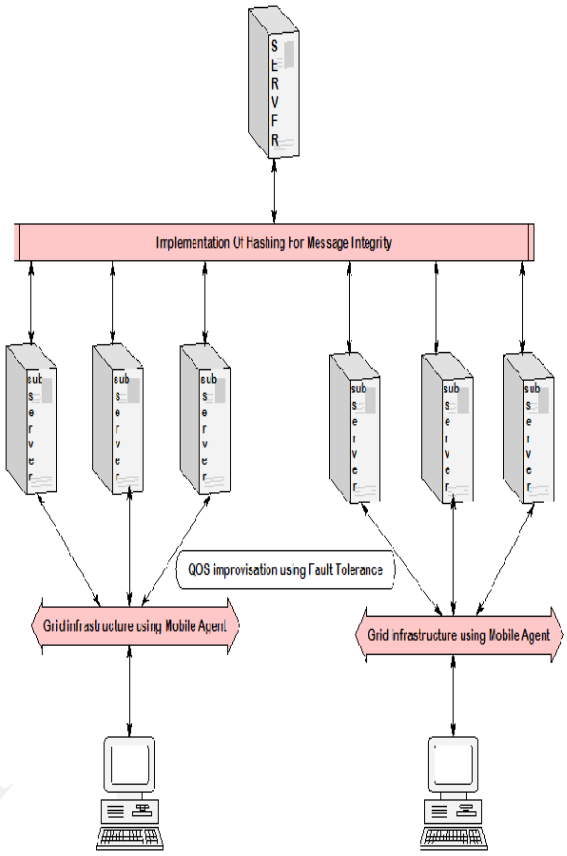

Fig 3: Entire proposed system architecture

2.Related Work

In [1] Map Reduce is an programming model and associated implementation for processing and generating large data set. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world task are expressible in this model. Program written in this functional style are automatically parallelized and executed on a large cluster of commodity machine. The run time system take care of this details of partitioning the input data scheduling the program's execution across the set of machines ,handling machine failure and managing the required inter-machine communication .This allows programmers without any experience with parallel distributed system to easily utilized of a large distributed system. Our implementation of Map

Reduce runs on a large cluster of commodity machines and is highly scalable.

In [2] Socket buffer is often claimed that TCP is not suitable transport protocol for data intensive Grid application s in high –performance networks..We argue that this is not necessarily the case. Without changing the TCP protocol, congestion control, or implementation, we showed that approximately tuned TCP bulk transfer can saturate the available bandwidth of a network path. The proposed techniques called SOBAS, is based on automatic socket buffer sizing at the application layer .In non congested path , SOBAS limits the socket buffer size based on direct measurement of the received throughput and of the round trip time.

In [3] TCP throughput collapse, also known as Incast is a pathological behavior of TCP that results in gross under utilization of link capacity in certain many-to-one communication patterns. This phenomenon has been observed by others in distributed storage, Map Reduce and web-search workloads. In this paper we focus on understanding the dynamics system of simultaneously communicating TCP entities. We propose an analytical model of account for the observed Incast symptoms ,identify contributory factors ,and explore the efficacy of solutions proposed by us and by others.

3. System model

Transport Control Protocol widely used on the internet and generally works well. TCP does not work well for many many-to-one traffic pattern on high-bandwidth and low latency network. We use Grid computing mainly for process allocation and scheduling subservers. We use mobile agent to handle failures thus the specific part of failed file will be transmitted rather than whole part to be sent. To ensures message integrity hashing technique is used. To improve quality of service we use fault tolerance system.

3.1 Grid Computing

Grid Computing acts as a layer of middleware to communicate with and manipulate heterogeneous and datasets. Grid computing is a significant approached forehancing the performance of process.It uses mobile agent for implanting grid computing. It increases performance by scheduling the process and allocation them to each subserver process so that

- It makes no computer to be idle.
- We can identify the status of each process.
- According to FCFS algorithm scheduling is done so avoid congestion in transmission.

Grid computing is the collection of computer resources from multiple location to reach a goal. The grid can be thought of as a distributed system with interactive workloads that involve large number of files. Grids are a form of distributed computing whereas a super virtual computer is composed of many networked loosely coupled computer acting together to perform to perform a large task. For certain application distributed or grid computing ,that relies on complete computers connected to a network by a conventional network interface such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processor connected by a local high speed computer bus.

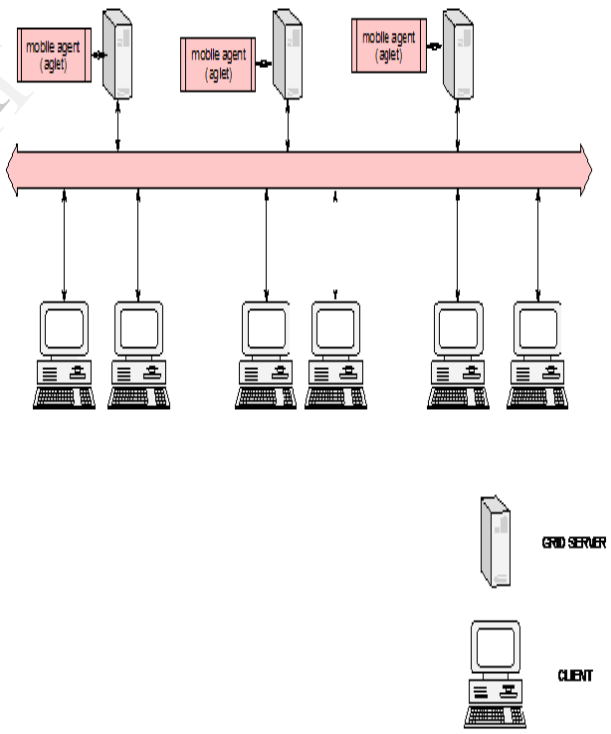

Fig 4: Grid Architecture

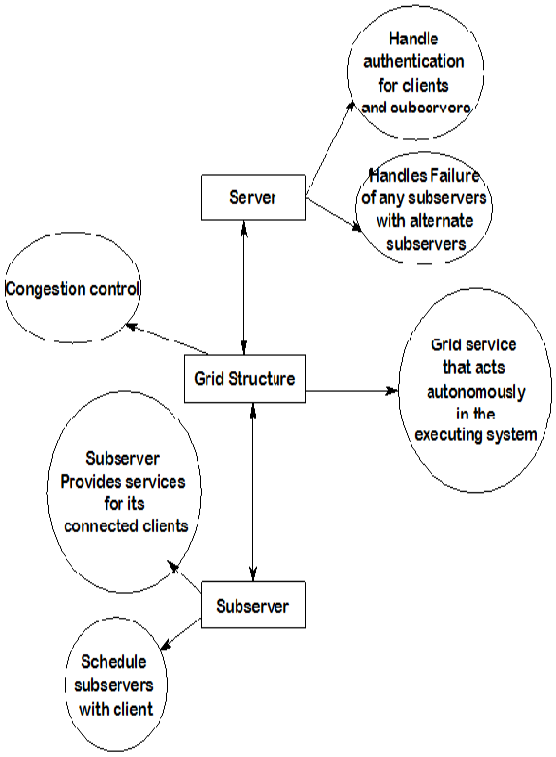

Fig 5: Grid Implementation

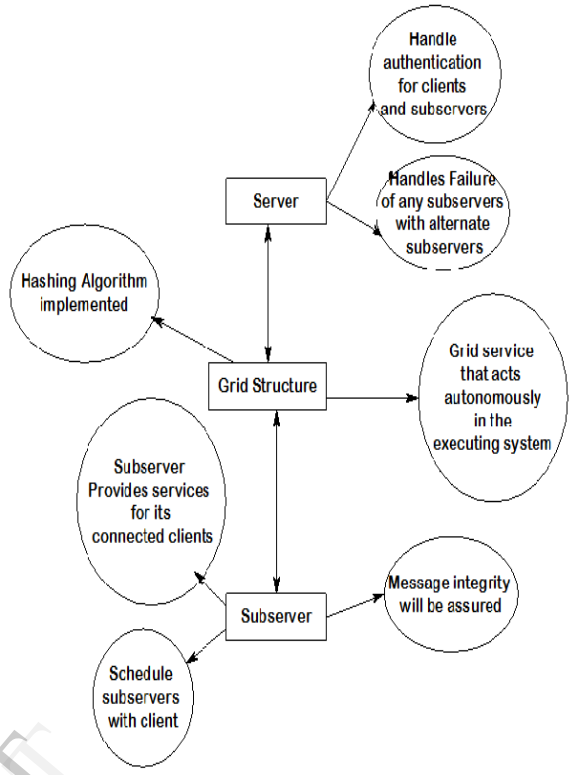

Fig 6: Hashing Architecture

3.2 Hashing Technique

Hashing is the very efficient method in the searching to the exact data item in a very short time. Hashing is the process in which we place the each and every data item at the index of the memory location for the purpose of ease usability.

Message integrity in sending and receiving packets in communication is ensured using hashing algorithm. It checks the integrity for message sends from source to destination .Attackers can change the message when message travelling in the network or delete the network or even delete the packets. To ensure integrity towards such message we implement hashing algorithm which checks each and every line even with the white spaces and sends the hash bucket to destination to assure integrity of messages .It efficiently manages message transmission in network congestion traffic.

3.3 Fault Tolerance

A Fault tolerance is a setup or configuration that prevent a computer or network device from failing in the event of an unexpected problem or error. To make a computer or network fault tolerant requires that the user or company to think how a computer or network device may fail and take steps that help prevent that type of failure.

Queries send to main server will be scheduled accordingly and allocates to be processed by its corresponding subservers. It will also provide fault tolerance in Incast congestion in Transport Control protocol which checks whether every subserver sends different part of message to the same client system. This kind of fault tolerance mechanisms improvises quality of service from servers.

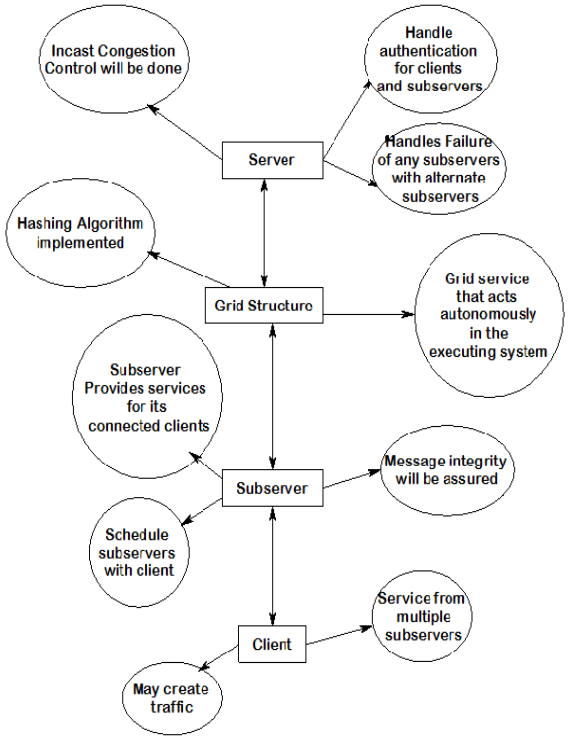

Fig 7: Fault tolerance Architecture

5. References

[1]A. Phanishayee, E. Krevat, V. Vasudevan, D. Andersen, G. Ganger, G.Gibson, and S. Seshan, "Measurement and analysis of TCP throughput collapse in cluster-based storage systems," in *Proc. USENIX FAST*,2008, Article no. 12.

[2] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. Andersen, G.Ganger, G. Gibson, and B. Mueller, "Safe and effective fine-grainedTCP retransmissions for datacenter communication," in *Proc. ACMSIGCOMM*, 2009, pp. 303–314.

[3] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "Thenature of data center traffic: Measurements & analysis," in *Proc. IMC*,2009, pp. 202–208.

[4] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. OSDI*, 2004, p. 10.

[5] M. Alizadeh, A. Greenberg, D.Maltz, J. Padhye, P. Patel, B.Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in *Proc. SIGCOMM*, 2010, pp. 63–74.

[6] D. Nagle, D. Serenyi, and A. Matthews, "The Panasas ActiveScale storage cluster: Delivering scalable high bandwidth storage," in *Proc. SC*, 2004, p. 53.

4.Conclusion

In this TCP incast congestion control we are using a technique of grid computing ,fault tolerance and hashing technique in order to reduce the problems in the related work and also to improve our proposed work. Using this reduces the traffic , congestion , data loss, buffer overflow and also improves the performance. So this adds a high level of advantage to our system.