**Cloud Computing**

**MidTerm #1**

**Sri Sai Anusha Gandu 16230560**

**1. Explain DCTCP’s unique characteristics compared with traditional TCP.**

**Answer:**

* DCTCP is a TCP-like protocol which is used for data centre networks.
* Unlike TCP, which reduces the window size by half in the case of congestion, DCTCP uses Explicit Congestion Notification (ECN) bits which help in maintaining the window size by reducing only to the required level.
* DCTCP produces high bandwidth, high tolerance and low latency.
* DCTCP helps the applications to handle background traffic without any impact on the current traffic.

**2. Explain the switch/router side algorithm in support of DCTCP.**

**Answer:**

The switch maintains a queue with the maximum value of ‘K’ packets. If the queue length exceeds the threshold value of K, the switch then marks the extra packets with the ECN bits that are the reason for the congestion. This marking of packets constitute the switch side algorithm of DCTCP.

**3. Explain the sender side algorithm in support of DCTCP.**

**Answer:**

The sender side algorithm gives a running average value that is used to cut down the window size as required during congestion. In the case of congestion, when the switch marks the ECN bits, the sender algorithm uses these bits to calculate the running average value.

**a = (1-g)\*a + g\*F**

Here,

a = running average from previous window

g = constant (0<g<1)

F = fraction of the packets that are marked in last window (with ECN bits)

This computation produces a value for variable ‘a’ which lies between 0 and 1

If the ‘a’ value is close to ‘0’, it indicates low congestion and if close to ‘1’, indicates high congestion.

This ‘a’ value is used to calculate the window size.

**w = w\*(1 – (a/2))**

This value of ‘w’ indicates the fraction that has to be reduced unlike TCP which always reduces to half. This helps in increasing the throughput.

**4. What are the benefits of DCTCP (name three)?**

**Answer:**

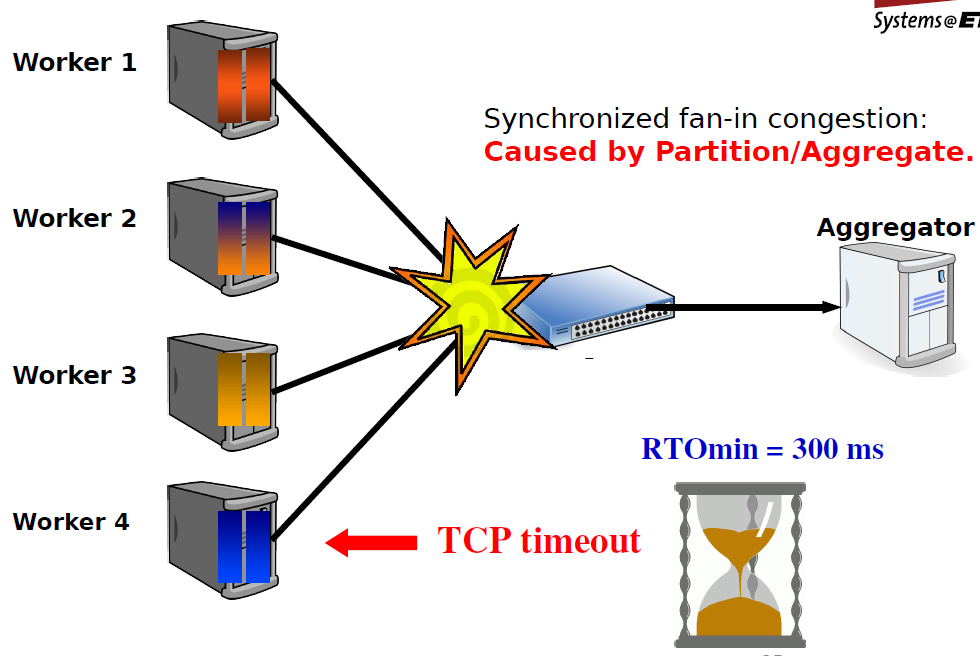
The benefits if DCTCP are listed below:

* **Queue Build-up:** On the sender’s side of DCTCP, a queue of length ‘K’ is present. When the queue overflows, the switch marks the extra packets with ECN bits and the sender forwards the packets in the queue. This process reduces the queuing delay over the switches where congestion is more likely to take place.
* **Buffer Pressure:** DCTCP holds the advantage of buffer pressure as the queue length is fixed and does not increase over time. This protocol has the capacity to handle almost 10 times more traffic than TCP. Hence, a few congested ports or switches would not harm the network.
* **In-cast:** This is a problem where number of small flows hit on the same queue for transmission which results in congestion when an extra packet hits the queue. However, in DCTCP, due to the marking of ECN bits, even when an extra packet is present at the switch, it does not cause buffer overflow or timeout.

**5. What is the TCP in-cast problem?**

**Answer:**

TCP in-cast is a problem that occurs at the switch due to many small flows to the same queue. When a complete task is given to a node, it then partitions the task into multiple sub tasks, assigns each one of them to a separate worker node and the response is received at the switch in a queue from all the worker nodes. This leads to congestion at the switch. When the number of worker nodes is further increased, the congestion at the worker nodes is increased proportionally even with 1 or 2 extra packets. This is called TCP in-cast problem.



**6. What are the causes of TCP in-cast problem?**

**Answer:**

The TCP in-cast problem occurs due to the partition and aggregate method. A complete task is divided into multiple sub tasks which are assigned to separate worker nodes. As the number of worker nodes increases, the congestion during aggregation at the switch increases which is the reason for TCP in-cast.

**7. Explain the proposed solutions for in-cast (name one solution and explain what is the solution about)**

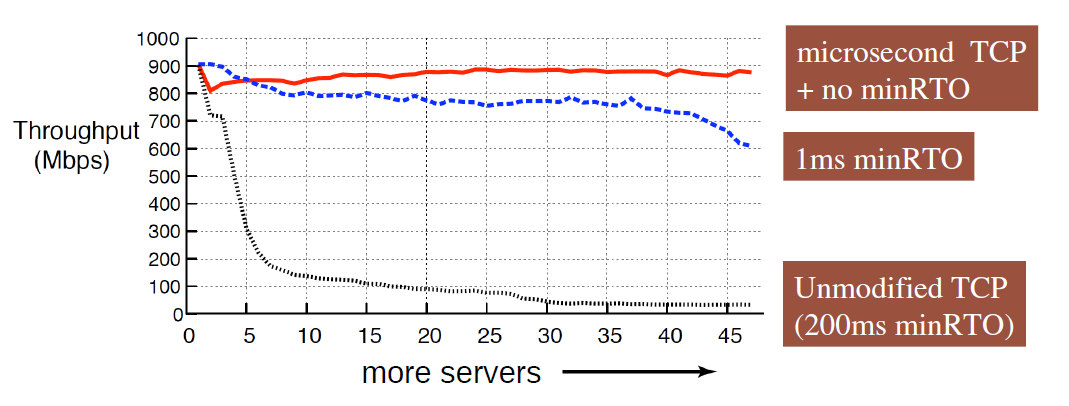
**Answer:**

The possible solutions to reduce the TCP in-cast problem are:

* Reducing RTOmin (Retransmission Time Out)
* Measuring RTT (Round Trip Time) using high resolution timers

**Reducing RTOmin:**

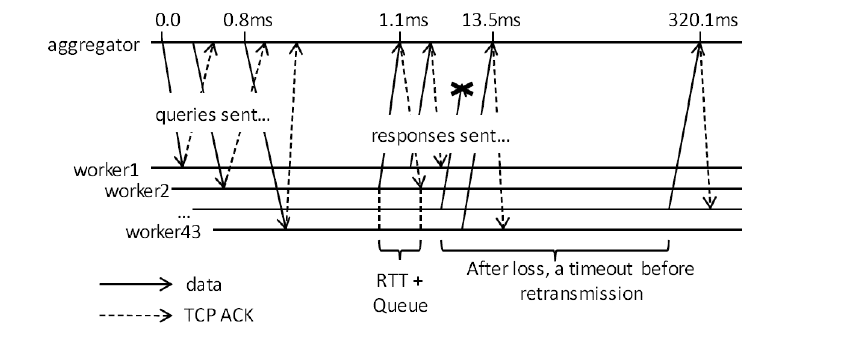
For a fixed number of servers, when RTOmin is comparatively large, throughput of the network is reduced. In order to increase the throughput, we reduce the value of RTOmin which leads to a comparative increase in throughput.

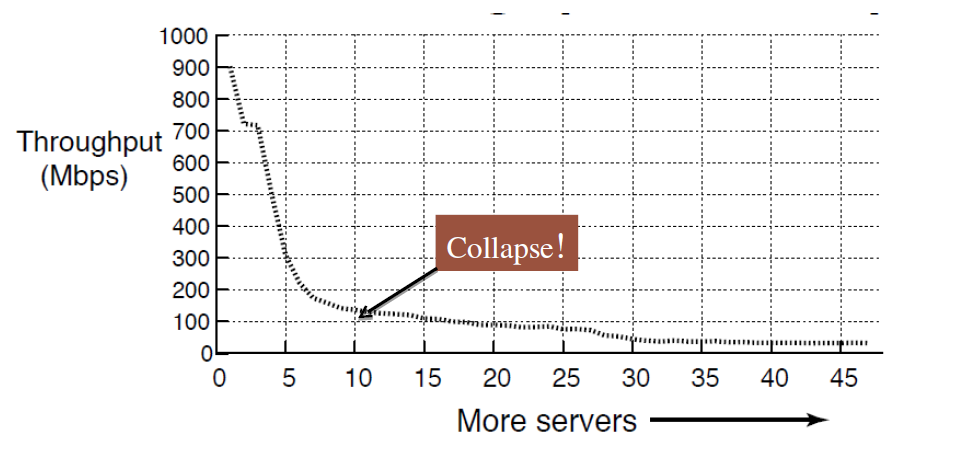


**8. Explain the TCP in-cast problem in the aspect of the long and short RTO (which RTO causes the high chance of TCP in-cast and why)**

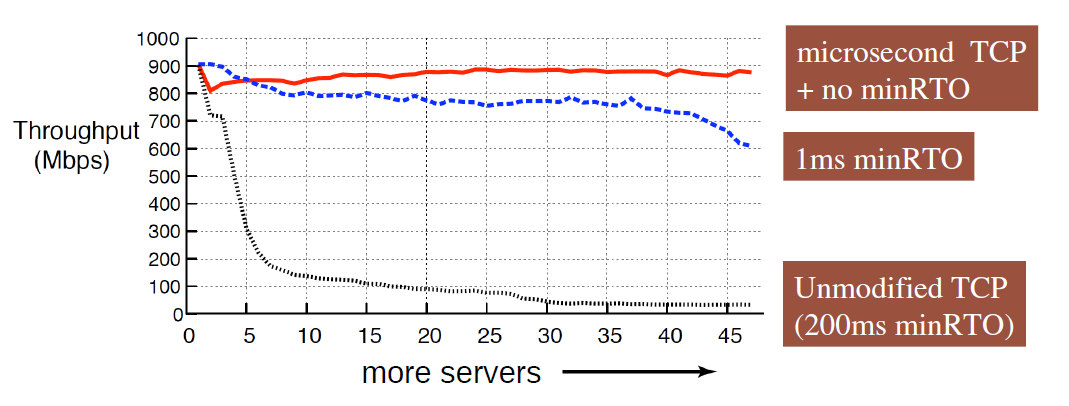
**Answer:**

TCP in-cast problem is occurred when a complete task is partitioned and assigned to separate worker nodes, which then send the response to the single queue at the switch which causes congestion. In the aspect of RTOmin, when the value is large enough, the switch has to wait the whole time until all worker nodes send in the response, which then has to be aggregated and sent to the receiver. This wait time at the switch even with data in hand decreases the throughput. The following figure demonstrates the reduction in throughput with large RTOmin





With reduced RTOmin, the switch can send the data collected from worker nodes as soon as the minimum timeout is completed. This reduces the waiting time and increases the throughput.



The value of long RTOmin causes high chance of TCP in-cast when compared to short RTOmin due to high waiting time at the switch for transmission.

**9. Briefly explain about Cloud Computing.**

**Answer:**

Cloud Computing is a convenient, on demand network access to a shared pool of computing resources that can be rapidly provisioned and released with minimal management effort and service provider interaction.

It refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services here are referred as Software-as-a-Service (SaaS) and the datacenter hardware and software is what we call Cloud. The service that is being sold is called Utility Computing.

**10. Briefly explain about virtualization and VM/Hypervisor.**

**Answer:**

The ability to run multiple operating systems (OS) on a single physical system and share the underlying hardware resources is called Virtualization. It is the construction of an interface that maps the virtual guest OS to the physical host OS.

Hypervisor, also called Virtual Machine Monitor (VMM), is a layer of software that is designed to support virtualization (server virtualization). As it sits between guest OS and host OS, it controls CPU, memory and storage of the guest user.

**11. Explain about types of (server) virtualizations, pros/cons**

**Answer:**

There are two types of virtualization:

* **Type-1:** Native/Bare Metal

This is a type of virtualization where the hypervisor directly sits over the hardware of the host system and the guest OS resides over the hypervisor.

* **Type-2:** Hosted

In this type of virtualization, the hypervisor sits over the OS of the host system and the guest OS resides over the hypervisor.

These are also classified as:

* **Full Virtualization:** This allows unmodified guest OS to execute on VM. This results in performance degradation.
* **Hosted Virtualization:** The guest OS gets installed and runs as an application. Virtualization capability is a part of host OS.
* **Hardware Virtualization:** In this type of virtualization, the guest OS takes help of the external hardware such as Intel VT and AMD-V processors to execute on VM.
* **Para Virtualization:** this type of virtualization allows the modification of guest OS to make it compatible with the hypervisor. This leads to better performance.

Pros of Virtual Server:

* Resource pooling
* Availability
* Migration
* Easy to deploy
* Redundancy
* Isolation
* Ability to reconfigure
* Optimization of physical resources

Cons of Virtual Server:

* Harder to conceptualize
* Indirect hardware access
* Security

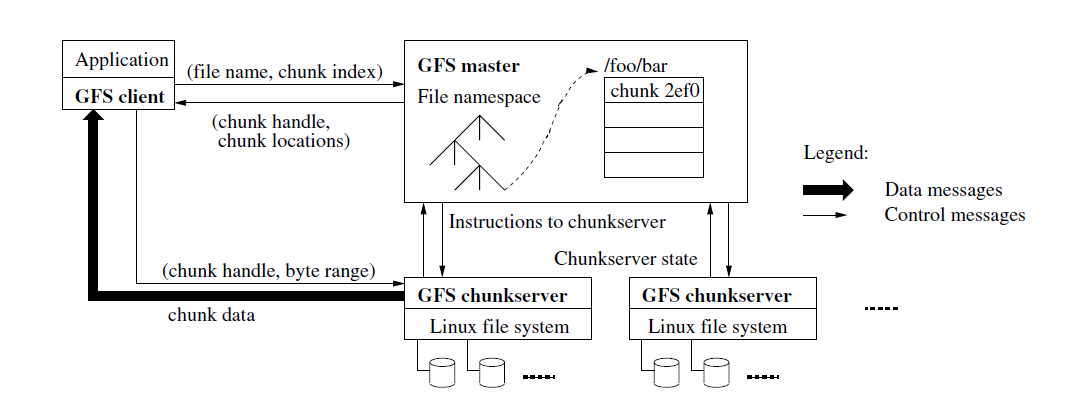
**12. Explain about the unique workload of GFS.**

**Answer:**

The workload in GFS primarily consists of 2 types of reads: large streaming reads and small random reads. In the case of large streaming reads, any individual operation reads hundreds of KB (almost 1 MB) of data. Some may also include a contiguous region of file. In small random reads, only a few KB of data is read. To increase the performance, some applications batch and sort the small reads to avoid back and forth movement. The workload also consists of large, sequential writes that gets appended to the file. Once any file has been written, it is modified very less often.

**13. Explain how file read works in GFS**

**Answer:**

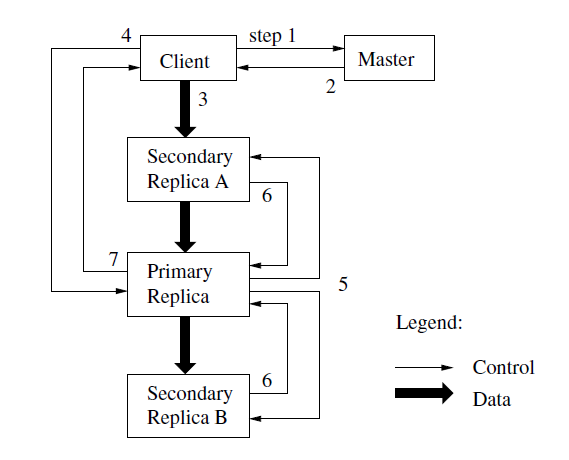


When a file needs to be read by the client, the following steps are followed.

1. The client first sends a request to the master with the file name and chunk index of the file to know about the location of the file.
2. The master then sends the chunk handle and chunk location of the requested file and all its replicas. The client caches this information.
3. The client then locates the closest chunkserver that holds the data and requests using chunk handle and byte range. No more client-master interaction is needed to read the file until the presence of cached information.
4. The chunkserver then sends the requested data to the client.

**14. Explain how file write works in GFS**

**Answer:**



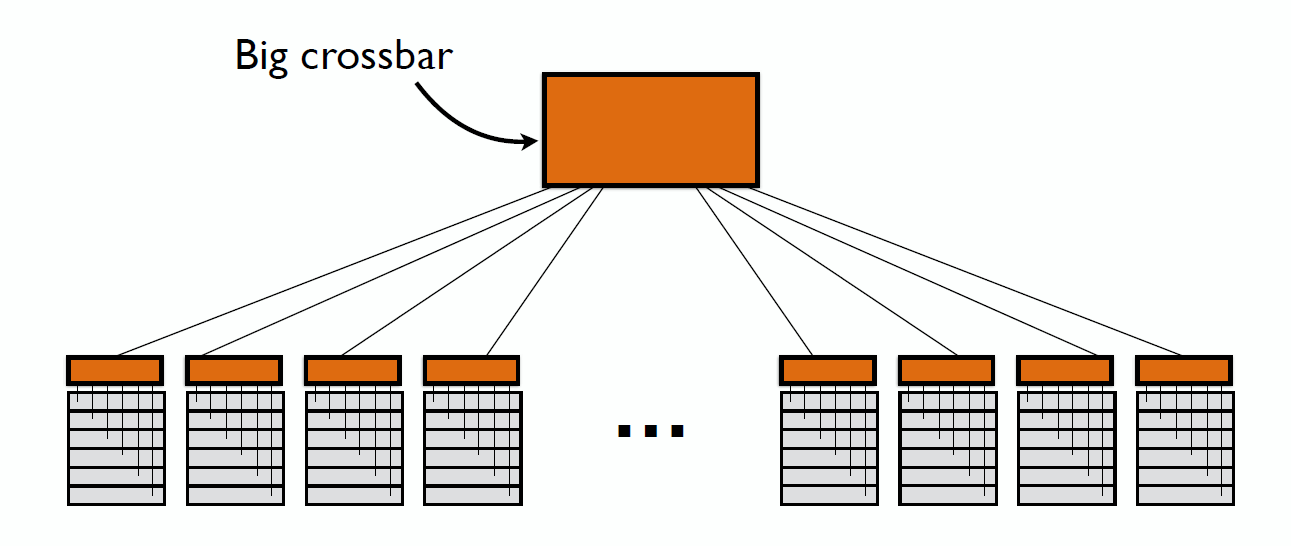
In order to perform write operation on a file, the client follows the following steps.

1. The client requests the master about the chunkserver which holds the current lease for the chunk and the locations of all the replicas. If it already exists, the master considers that lease. If not, it creates a new lease.
2. The masters responds with the locations of the primary and secondary replicas to the client. It caches the data to avoid any unnecessary interactions with the master.
3. The client forwards the data to all the replicas. There is no specific order in which it has to send the data. It does so to the replica that is closest to the client.
4. After the acknowledgement of data from all the replicas, the client sends a write request to the primary replica. It then identifies the data previously sent and assigns an order in which the data has to be modified.
5. The primary replica then forwards the write request along with the order of mutation.
6. Once the operation is completed by the secondary replicas, they reply to the primary replica.
7. The primary replica then acknowledges the client regarding the completion of the write operation.

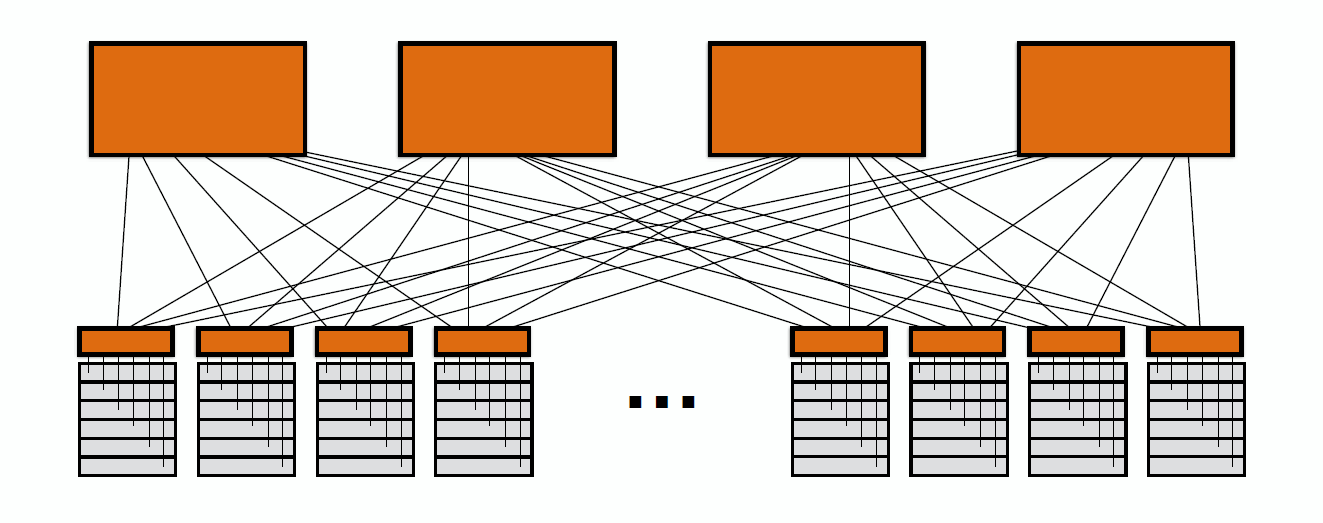
**15. Explain about DC architecture and network (BW, delay) and traffic characteristics**

**Answer:**

The DC Architecture is depicted below.



The aggregator/router is located at the top and is connected to each and every server rack located at the bottom. In the case of single router, it may be acceptable but with a number of users, it leads to congestion.



Each of the server rack consists of a ToR (Top of Rack) switch at the top. This is the source that connects all the servers on the particular server rack.

The DC characteristics constitute load balancing, congestion control and reconfigurable topologies.

The network characteristics constitute high bandwidth and low latency/delay.

The traffic patterns are highly volatile and unpredictable. The characteristics include small traffic for large flows and large traffic for much number of small flows.

**16. Motivation of DCTCP (or other TCP variants for DC)? What is/are the problems of traditional TCP being used in DC?**

**Answer:**

TCP is primarily designed for Internet. The use of TCP for other networks like wireless or datacenter (high bandwidth/delay product) is not so efficient. When used, it results in very high congestion and lowered quality of the response (data). To avoid this problem, TCP has to be modified for each of the network when separately used which is a hectic task. Hence, separate protocols for each of the other networks is created including SnoopTCP for wireless and DCTCP for datacenter.