An effective solution of TCP Incast problem

GeJunwei[1, a], ZhaoZhibin[2, b], FangYiqiu[2, c]

[1,2]Chongqing university of posts and telecommunications , Chongqing 400065,china

[a]gejw@cqupt.edu.cn,[b]zhao_zhi_bin@163.com,[c]fangyq@cqupt.edu.cn

Keywords: cloud computing; data center networks; TCP Incast; NS2 simulation;

Abstract. Have sketched the current main mitigation methods for TCP incast problems in cloud computing data center networks,an effective method is put forward simultaneously, and the method is verified by NS2 simulation experiments.Analysis results show that this method can really alleviate the TCP incast problems, but also has some shortcomings and limitations.

1. Introduction

Cloud computing data center need reliable transmission, TCP protocol is applied in a wide area network, NOW TCP protocol is applied to the data center,But due to high bandwidth and low latency data center network environment and the reasons for the original data center network topology,the reasons for the original data center network topology, so the TCP Incast problem happened,For the problems,Began to study the problem at home and abroad.Hope to find a reasonable solution to alleviate even solve this problem completely.There is no good way to solve this problem completely.This article is for the previous scheme of improvement,On the basis of predecessors' improvement effect to further improve the effect of improvement.

2. Background

Cloud computing data center network is widespread the many-to-one communication model,Under such a communication model,The client connected to many servers through the switch.Means that multiple servers Shared a switch.When client request to the server data blocks,Such a block of data is stored on multiple servers respectively.The data on the server called SRU.Respond to the request on the server.Subsequent to the client send the SRU.When the client receives all of the data block SRU.In the next block of data sent on the server.In such a mode of communication,A short time a large amount of data into a switch.Make the switch buffer overflow,In TCP RTOmin is 200 ms,But in the cloud computing data center network, RTT is commonly 100 ms (even lower).In such an environment,A data flow send timeout, TCP will wait for a shortest duration for RTOmin timeout period.Assume that the otherserver has completed the current data transmission,And release the bandwidth, but the client receives all wait RTOmin.At this time, link bandwidth almost free, cause throughput fell sharply on the application layer, TCP incast problems happened.

TCP incst problem was found in distributed storage system. In addition to the distributed storage system, a lot of application of the data center, including web search, MAPRDUCE distributed computing, such as TCP incast problem can occur .

The principal causes of the problem :1. Data of sudden transmission at the same time, the switch light buffer capacity is limited, leading to buffer overflow, after to packet loss;2. A large number of packet loss in TCP congestion control, send window in half, reduce transmission speed;3.TCP timeout retransmission of lost packets, according to the RTO default values, timeout can last 200 ms, and the data center network RTT (10 μs to 100 μs) far less than the RTOmin, lead to more than 90% link in the free state.

3. The main solution

At present, there are many solution to alleviate this problem.These solutions can be divided into three broad categories: application layer solution,transport layer solution, link layer solution.

Application layer solution : increase the size of the SRU and data transmission delay.

Increase the size of the SRU:In TCP incast model, if the data flow timeout, no overtime data flow has completed the data flow of the current data transfer,The release of the bandwidth, the link is in the idle state.Increase the size of SRU, no timeouts in the data flow can be sent within the timeout time, reducing the link of free time, increase the link utilization, and increase the throughput.But, in this way will also as in a distributed system, a bigger SRU will increase kernel memory pressure, will cause the attempt failed.

Data transmission delay:This method is to limit the number of concurrent transmission servers, the server response, to avoid a large amount of data into the switch buffer, avoid TCP incast problem effectively.Time-lapse data transmission can be effectively avoid this problem in theory, but in fact the lack of effective experimental verification and the theory of effective data.

Transport layer solution: improve TCP protocol and modify TCP parameters .

Improve TCP protocol: Using TCP protocol data center (DCTCP), DCTCP simple tag at switch mechanism, when the switch buffer exceeds a specified threshold, with congestion through code point (CE) mark the package, DCTCP source side according to the marked packets size to adjust the window size, the proportion of the marked the ratio,The greater the proportion of tag, reduce the amplitude of the greater.The goal of DCTCP that is in data center network switch buffer capacity is very small for high burst tolerance ability, low latency and high throughput.DCTCP through in the network using the display congestion notification (ECN) to provide terminal host according to the feedback information.Based on literature[4] the establishment of a mathematical analysis model analysis DCTCP throughput and delay performance, their results showed DCTCP throughput can maintain more than 94%, their analysis is based on only a bottleneck link, in the actual network circumstances, the actual network also need to study, DCTCP did well in the experimental environment, pointed out that ECN switches, less not widely deployed in the center of the data.

Incast using TCP congestion control (ICTCP), mainly is the receiver design a ICTCP mechanism, implement Incast congestion avoidance.ICTCP according to the connection of throughput and bandwidth available, before the packet loss occurs, take the initiative to adjust each TCP connection receive window size, thereby avoiding the TCP incast problem.Literature[5] the experimental results show that ICTCP is close to zero overtime, to avoid the congestion effectively, provides a high performance data flow between competition and fairness.However, in actual network, the accurate estimate available bandwidth is very difficult, in the transfer volume is very small, ICTCP cannot make full use of bandwidth.

Modify TCP parameters:The main method is to reduce the RTOmin, the traditional TCP protocol RTOmin value is 200 ms, this value is higher than data center network RTT, the data center RTT is approximately 100 microseconds.Due to the data center network link bandwidth high, the transmission time of each block is far less than RTOmin, there have been great waste of bandwidth.reduced RTOmin can greatly increase the throughput.IN reality that can't provide the efficient software regularly.

The link layer solution:

On Literature[6] The researchers use Ethernet TCP incast flow control to solve the problem, when the switch buffer overload, support to all flow control switch to the switch interface to send data sends a suspension frame, notification to the switch to send data server is stopped or wait for a period of time in sending data to the switch.The experimental results showed that when a switch is can effectively alleviate the TCP incast, but when multiple switches, occurring adversary block make serious decline in performance. At the same time, the exchange of different vendors have different problems.

4 our solution and verification

Our solution is based on reducing packet length to alleviate the problem .First of all, introduce the plan reducing packet to alleviate the problem.

This scheme is given theoretical analysis:Assume that the system has N sending server and a receiving server, all the server to the switch link capacity is C, transmission experiment set is T, L is the packet length, B is switch cache size. M is set to switch caching and number of grouping can carry on a link. i.e. M = (B + 2 C*T)/L (1).

At the same time, under the assumption that each between sending and receiving server only a stream, and flow between fair enough. i.e. W = M/N (2).

Assume the TCP flow congestion avoidance stage, namely the wheel RTT packet loss does not occur, the congestion window is added 1, each flow assumes that flow is in a state of complete synchronization at the same time, at the same time flow is fair, each TCP flow up to throw a grouping within the RTT. If the TCP throughput collapse did not occur, the TCP flow can be recovered from a packet loss event, not detected by timeout.If an ACK packet loss can be repeated three times fast detection, then the congestion window for at least five, because the TCP window is greater than 5 can detect through repetitive ACK packet loss. W> 5. i.e.N< (B + 2 *C*T) / 5 L (3).

Explain the work server type on N in accordance with the number of TCP incast throughput is not going to happen, namely Nmax is the maximum number of work server, increase with the decrease of number does make work server packet length. In the actual network, TCP flow not absolutely fair and synchronization. But this rule is established.This shows is that under the same conditions, the decrease of the packet length, for the server in a fixed number of cases, the link will effectively increase the effective throughput, which proves that the TCP incast effectively alleviate the problem.

To reduce the packet length to reduce the TCP incast problem. Before the simulation gives the concept of two kinds of load, a son is a fixed block size load and the load of fixed block size.Stator block size load is to point to in a request, the Worker is sent to the Aggregator the amount of data is a fixed size, have nothing to do with the number of Worker, a fixed block size load refers to all the Worker of the total amount of data that are sent to the Aggregator is fixed, each Worker to send the amount of data is determined by the number of the Worker. In this paper, the simulation are involved is a fixed block size loads.

In this scheme is put forward on the basis of their own improvement plan, my improvement program is to reduce the packet length, and in combination with shortened the retransmission timer value.

In the simulation experiments to verify their own solutions. In this paper, the simulation experiment of using open source NS2 simulation platform (NS2.35), under the Linux system (ubutun14.04) simulation experiment. For the simulation experiments, because of my plan is to reduce packet length and shorten the retransmission timer is used at the same time, first of all, to shorten the retransmission timer, namely reduce RTOmin.

The experiment simulating network topology is given:

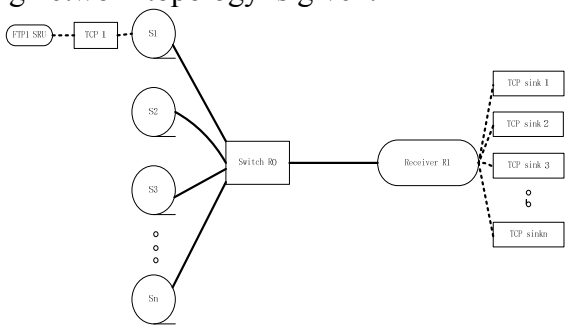


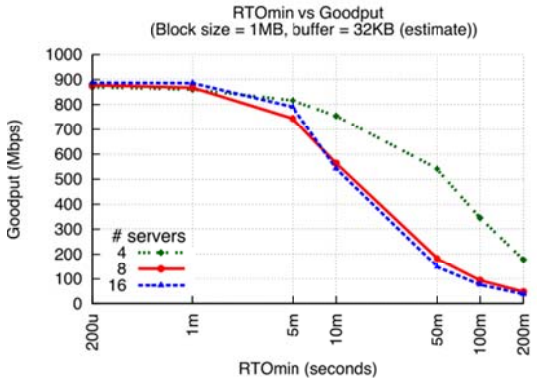
Figure 2 experiment simulating network topology.

The simulation parameters:

The specific simulation parameters	
Number of servers	variable
Switch buffer	variable
Link bandwidth	1000Mbps
Link delay	25 μ s
RTT	100 μ s
RTOmin	variable
The server request unit SRU	variable

NS2 is an event-based simulator, so on the granularity of the timer is unlimited, so can be used to simulate the NS2 to intuitive understanding of shortening the retransmission timer to the problem of TCP incast relief, the main measure is the throughput.

The literature[1] is introduced in detail to solve the problem of TCP incast by reducing RTOmin and the restrictive conditions, then borrow the literature [1] for reducing RTOmin in real concrete throughput as shown in figure 3.



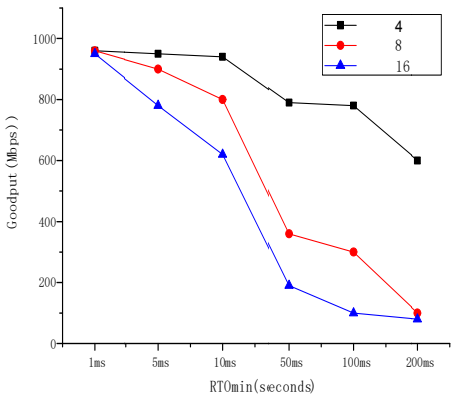
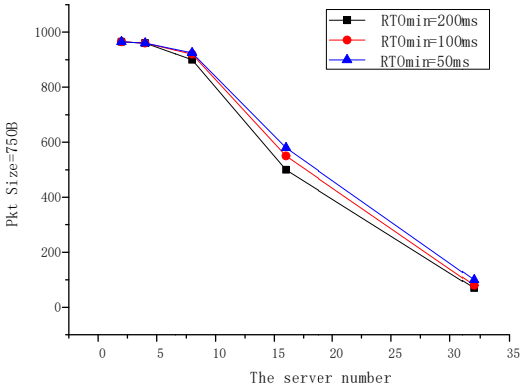


Figure 3 reducing RTOmin in reality Figure 4 reducing RTOmin in simulation

In real cases, reducing RTOmin can significantly increase the throughput.

In our own experiment environment using the same data for simulation, data summary, figure 4.Comparison figures 3 and 4, we get that reducing RTOmin in our experiment environment, experimental results show that under the current experimental environment, the number of servers is 4, have a certain gap between the reality ,but for servers of 8 and 16, basically can be close to the reality, as a result of the main consideration is the number of servers more cases, the effective throughput evaluation, all we think that our environment simulation is reasonable.

On this basis, we combine to reduce the length of the grouping method for simulation experiment, figure 5 and figure 6.



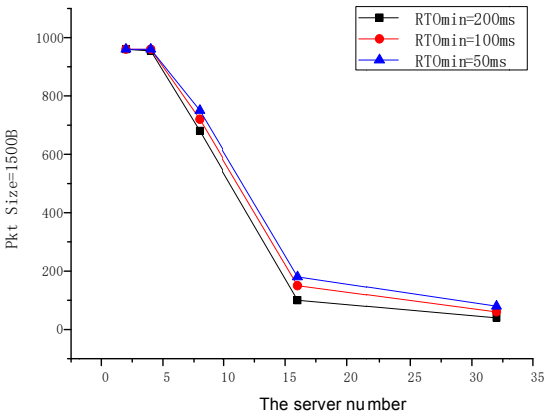


Figure 5 grouping of 750 B Figure 6 grouping of 1500B

Comparative analysis, we can concluded that obviously reduce the packet can significantly improve the throughput of a link, combining can shorten RTOmin a certain degree of increase throughput (increased the effective throughput about 5%), so the method of combining, in the same

condition, can effectively improve the actual throughput, despite the promotion effect is not very prominent, but in reality, if reduce the grouping method is used to alleviate the problem of TCP incast, then in combination with low RTOmin really can better alleviate the problem.Because this method by the limitation of two methods, this method also encountered the same situation, main is to reduce the packet length will bring additional group head overhead, reduce the utilization of the link, when reduced to 400B from 1500 B, the link utilization rate reduced to 94.9% from 82.2%, when less than 400 B link utilization is also fell sharply, so the minimum group limited to 400 b. At the same time because the Linux system default delay ACK is 40 ms, when less than 40 ms, will produce an ACK haven't reached timeout retransmission redundancy phenomenon. So we set a lower RTOmin limit is 40 ms.

5.Conclusion

This article simply describes the TCP incast problem, at the same time, this paper expounds the main reasons for this problem, and then introduces the some mainstream approach to this problem, to solve this problem, the effect of and some problems.And then put forward to solve the problem by ourselves, is to reduce the combination of reducing the packct and shorten the retransmission timer scheme, through the simulation experiments verify the plan, do to alleviate this problem can produce effect, although the effect is not very obvious, and also puts forward the solution of some of the problems and constraints.At the same time, to the specific effect of this method in real cases need further research. Through the study of many literatures, and validation of their scheme analysis, to fundamentally solve this problem, need according to the characteristics of cloud computing data center network TCP congestion control algorithm and conduct the thorough research to the transport protocol.

References:

[1] Vasudevan V, Phanishayee A, Shah H, et al. Safe and effective fine-grained TCP retransmissions for datacenter communication[C]//ACM SIGCOMM computer communication review. ACM, 2009, 39(4): 303-314.

[2] Phanishayee A, Krevat E, Vasudevan V, et al. Measurement and Analysis of TCP Throughput Collapse in Cluster-based Storage Systems (CMU-PDL-07-105)[J]. 2007.

[3] Chen Y, Griffith R, Liu J, et al. Understanding TCP incast throughput collapse in datacenter networks[C]//Proceedings of the 1st ACM workshop on Research on enterprise networking. ACM, 2009: 73-82.

[4] Ke D, Yongmao R, Jun L. Avoiding TCP incast through scheduling data requests[C]//Multimedia Information Networking and Security (MINES), 2012 Fourth International Conference on. IEEE, 2012: 453-457.

[5] Wu H, Feng Z, Guo C, et al. ICTCP: incast congestion control for TCP in data-center networks[J]. IEEE/ACM Transactions on Networking (TON), 2013, 21(2): 345-358.

[6] Brakmo L S, Peterson L L. TCP Vegas: End to end congestion avoidance on a global Internet[J]. Selected Areas in Communications, IEEE Journal on, 1995, 13(8): 1465-1480.