Incast

(TCP) Incast

A performance issue can be observed with some distributed applications in data center networks. The basic story is that a client sends a request to multiple servers, and these servers send their responses at more or less the same time. If the senders are well synchronized and the responses are large, this leads to bursts of traffic arriving from multiple ports at the switches "upstream" from the receiver. Because data center switches such as ToR switches often have small buffers, this can lead to correlated packet loss. TCP reacts badly to these kinds of loss, and for applications that have to wait for all the responses, the overall times to completion can increase spectacularly.

Suggestions to Mitigate or Avoid Incast Problems

There have been a variety of suggestions to address performance issues due to incast:

Larger Buffers

The effect of incast congestion can be mitigated by having large-enough buffers in network devices where incast can occur. However, this has a direct impact on the cost of the network. Also, larger buffers can lead to increased delays where there is congestion, which harm performance in general.

TCP Configuration Changes

TCP can be reconfigured to use more aggressive retransmission in cases of packet losses. In particular, the original paper recommends to send the RTO (Retransmission Timeout) value to a low value such as one millisecond. (Its default in Linux seems to be 200ms.)

Active Queue Management and ECN

It has been suggested that AQMandECN can improve behavior of TCP in incast situations.

Changes to TCP

Improvements to TCP's congestion control have been a popular research topic for a long time. DC-TCP (Data Center TCP) was developed partly in response to Incast problems.

Pacing

Senders can artificially limit their rate of sending in order to reduce the change of congestion.

Application-Level Changes

Changes at the application level could reduce the amount of incast-induced congestion, for example by de-synchronizing reply traffic in distributed applications.

References

- *Understanding TCP Incast and Its Implications for Big Data Workloads,* Yanpei Chen, Rean Griffit, David Zats and Randy H. Katz, in: Proceedings of the 1st ACM workshop on Research on enterprise networking (WREN'09) (pdf, 2012 Tech Report)
- *ICTCP: incast congestion control for TCP in data-center networks*, Haitao Wu, Zhenqian Feng, Chuanxiong Guo, and Yongguang Zhang, In: ACM CONEXT Proceedings, November 2010 (pdf)
- draft-bensley-tcpm-dctcp-03, *Microsoft's Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters*, Stephen Bensley, Lars Eggert, Dave Thaler, April 2015
- draft-zheng-tcpincast-00, *An Effective Approach to Preventing TCP Incast Throughput Collapse for Data Center Networks*, Hongyun Zheng, Chunming Qiao, Kai Chen, Yongxiang Zhao, June 2016

– %USERSIG{SimonLeinen - 2014-12-31 - 2016-06-25}%