Paper Evaluation, Queues Don't Matter When You Can Jump Them!

Jiani Jiang <jianij@kth.se>

Paper summary

Many datacenter applications are sensitive to tail latencies, one source of latency tails is network interference. Network interference occurs when congestion from throughput-intensive applications causes queuing that delays traffic from latency-sensitive applications.

To address this problem, they introduce QJUMP, to mitigate network interference, QJUMP applies internet QoS-inspired techniques to datacenter applications. The basic idea of QJUMP is that packets from higher levels are rate-limited in the end host, the higher levels packets can jump the queue over packets from lower levels.

In QJUMP, they would like to use multiple values of f (throughput factor), so that different applications can benefit from the latency variance and throughput tradeoff that suits them best. QJUMP couples priority values and rate-limits: for each priority, they assign a distinct value of throughput factor, with higher priorities receiving smaller values. QJUMP users must choose between low latency variance at low throughput (high priority) and high latency variance at high throughput (low priority).

They evaluate QJUMP both on a small deployment and in simulation. There are some key findings.

1) QJUMP resolves network interference for a collection of real-world datacenter applications. When they run the QJUMP with Hadoop, Memcached and PTP, they find that it is can mitigates the network interference, resulting in near ideal performance. And in a realistic multi-application setting, DJUMP both resolves network interference and outperforms other readily available systems.
2) They compare outperforms of Ethernet Flow Control, ECN and DCTCP in their deployment. In Ethernet Flow Control, it has a limited positive influence on memcacehed, but increases the RMS for PTPd. ECN very effectively resolves the interference experienced by PTPd and Memcached. However, this comes at the expense of increased Hadoop runtimes. DCTCP increases in Hadoop job runtimes. Only QJUMP can achieve nearly ideal stage.

Top 3 contributions

1) QJUMP resolves network interference for latency-sensitive applications without sacrificing utilization for throughput-intensive applications.
2) QJUMP offers bounded latency to applications requiring low-rate, latency-sensitive messaging.
3) Comparing to competing systems, including ECN, 802.3x, DCTCP and pFabric, QJUMP performs close to or better, but is considerably less complex to understand, develop and deploy. QJUMP is simple and immediately deployable, requiring no changes to hardware or application code.

Problems

1) When they calculate service delay, they assume packet size P= 1500B, but in practice, we may face with a mixed size packet, I think maybe this will affect the system.
2) In some datacenter, maybe they can not control all the host packets, so they can not give priority to these packets, which will make the higher priority packets can not jump over these queues.