Scalable Shared-Nothing QoS (SNQoS) for High-Performance Storage Systems

Yan Li, Xiaoyuan Lu, Will Kong, Ethan Miller, Darrell Long
(yanli, xyuanlu, wkong1, elm, darrell)@soe.ucsc.edu
Storage Systems Research Center
University of California, Santa Cruz

Motivation
Modern high-performance file systems distribute I/O requests to many servers to maximize performance. This parallelism may cause congestion in many places in the system.

Comparing a random write workload when running without QoS and with SNQoS:

Without QoS, the throughput varies greatly between clients, and shows significant fluctuation.

Congestion harms the efficiency of the system
- Decreases bandwidth utilization and fairness
- Storage system tuning is hard
- Modern storage systems can have thousands of parameters
- Manual tuning and optimizing are slow
- Need to run lots of benchmarks and analyses
- Need to handle a wide range of workloads
  - Sequential versus random, read versus write

QoS Rule (Congestion State (CS) statistics) → <action>
- CS statistics: ack_ema: exponentially weighted moving average (EWMA) of gaps between RPC acks send_ema: EWMA of gaps between sender timestamp embedded in RPC acks
- Actions: m, b: control the congestion window (cwnd)
  - cwnd = m × cwnd + b
  - t: minimum interval between two outgoing RPC requests

A Sample QoS Rule Set
This is a sample rule set that maps a 2-D congestion state space, consisting of ack_ema and pt_ratio, to different actions.

Evaluation & Results
We have evaluated seven workloads, and SNQoS increased the throughputs of all of them.

Two objective functions are used: TP only seeks max throughput, TP and Var seeks a balance between throughput and the fairness of bandwidth allocation.

The sequential read and write workloads got the greatest boost in throughput.

If the users need to share the bandwidth fairly, they can choose the red rules, which show both increased throughputs and lowered speed variances.