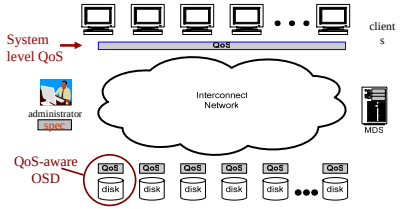
Providing Quality of Service Support in Object-Based File System

Department of Computer Science, University of California, Santa Cruz
Joel Wu, Scott Brandt

Overview

Class-based performance isolation
Two parts:

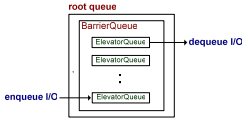
1. QoS-aware OSD: Enable individual OSD to be QoS-aware
2. System-level QoS: How individual QoS-aware OSD can work together to provide system-level QoS



Q-EBOFS

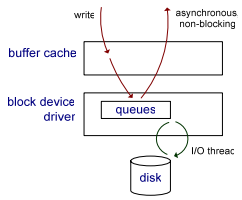
- Based on EBOFS (Extent and B-Tree Based Object File System) by Sage Weil
- Performance isolation achieved through
  - Queueing
  - Buffer management

Original EBOFS queues

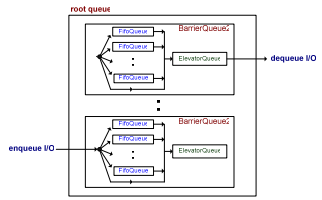


Buffer Cache Management

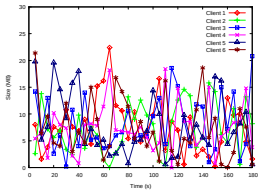
- Writes are asynchronous in EBOFS
- Will only block when buffer cache is approaching full
- Throttle writes through selective blocking

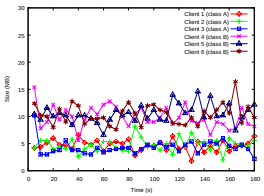


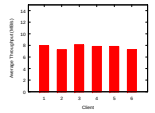
Q-EBOFS queues



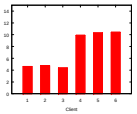
Q-EBOFS Results



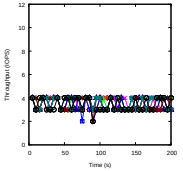




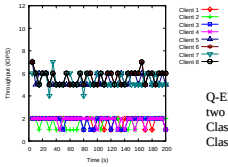
6 clients writing
original EBOFS
No performance isolation



Q-EBOFS with two classes
performance isolation
Class A - 30%: Client 1-3:
Class B - 70%: Client 4-6:



8 clients, mixed read/write
68% read percentage
original EBOFS behavior



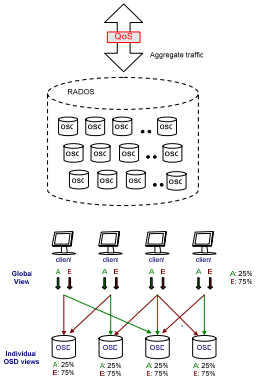
Q-EBOFS behavior
two classes:
Class A - 25%: client 1-4
Class B - 75%: client 5-8

System-Level QoS

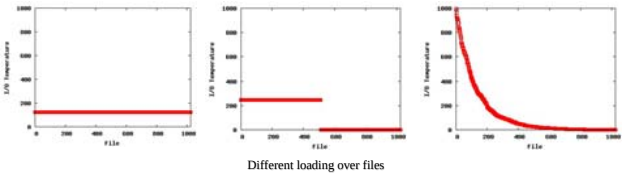
- Data is striped across OSDs in Ceph
- Existing approaches for system-level QoS over distributed storage require centralized components and/or propagation of global state information
- Ceph – designed with extreme scalability in mind
- QoS framework for Ceph should preserve scalability
  - Avoid introduction of potential bottlenecking components
  - Avoid introduction of additional complexities

Approach: Leverage on randomized data distribution

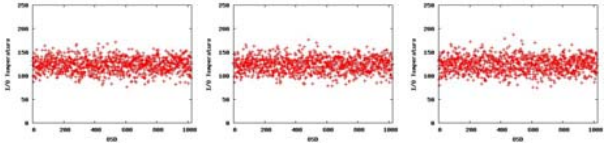
Hypothesis is that CRUSH can distribute data/load well enough such that independent per-OSD sharing can combine to approximate the same global-level sharing during overload.



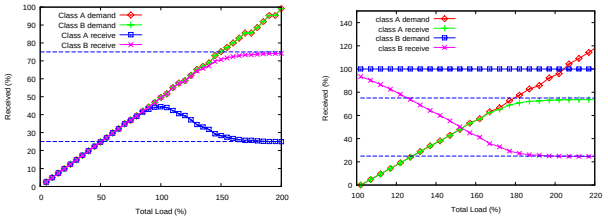
System-Level QoS Results



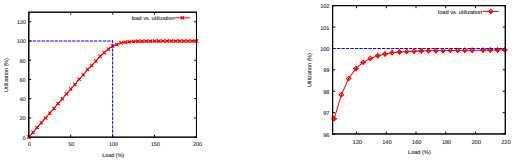
Different loading over files



Resulting load distribution over OSDs. All OSDs similarly loaded.



Demand vs. receive at system-level for two classes, 25% and 75%



Load vs. utilization – 100% load close to 100% utilization

Summary

1. Q-EBOFS can provide performance isolation at OSD
2. Leveraging on randomized data distribution, a collection of OSDs working together can satisfy QoS goals at system-level without global state information