RADoN: QoS in storage Networks

Tim Kaldewey, Andrew Shewmaker, Richard Golding†, Carlos Maltzahn, Theodore Wong†, Scott Brandt
Computer Science Department, University of California, Santa Cruz
†IBM Almaden Research Center
{kalt, shewa, golding, carlosm, tmwong, scott}@cs.ucsc.edu

Specific performance requirements in large scale storage systems are commonly achieved by physically or temporally partitioning (*e.g.* isolating) workloads, or by over-provisioning the system. Despite constantly falling hardware prices, facility and power expenses make physical partitioning costly and inefficient. Over-provisioning of shared storage systems does not isolate workloads, hence irregular workload behavior—especially peak loads—will have significant impact on concurrent workloads. With constantly growing storage demands and data centers reaching their physical limits, more intelligent solutions are required.

Recent disk schedulers can achieve nearly perfect temporal isolation, allowing reservations close to the maximum physical disk performance [2, 3]. Many network schedulers have been developed, allowing QoS guarantees. However, an integrated mechanism providing end-to-end QoS in large scale networked storage systems is still missing. We are currently building a framework called RADI/O[1] to manage reservable end-to-end storage performance including absolute performance guarantees. It comprises a real-time disk scheduler [3], QoS aware Caching, and the RADoN[2] networking component.

RADI/O is intended to cater to a wide spectrum of applications including those with real-time I/O requirements. Hence RADoN must both tightly control network traffic and keep the server cache occupied so that the disk scheduler has the opportunity to optimize for sequential accesses within and across reservations. We are currently evaluating different flow control mechanisms via extensive simulations based on the queuing model shown in Figure 1. Clients are allowed to submit a storage request to the system when tokens are available. Tokens are doled out by the server, which is constantly monitoring the cache occupancy of each client. Network and Disk are currently modeled as fixed delays.

In the most promising implementation, clients replenish tokens required to achieve the reserved performance themselves based on server-assigned rates and periods, while the server directly manages tokens for unused resources. Figure 2 shows that even when certain applications violate service level agreements by generating requests beyond their reservation, RADoN manages cache occupancy such that after a reasonable startup time all reservations are fulfilled.

[1] storage is a form of **I/O** and all of the framework's components use the **RAD** model [1] to manage resources

[2] **RAD o**n the **N**etwork

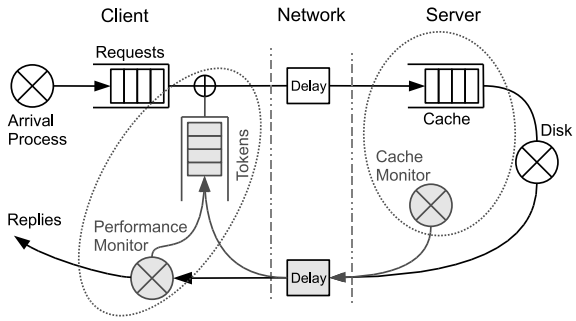

Figure 1: Queuing-theoretic model of RADoN

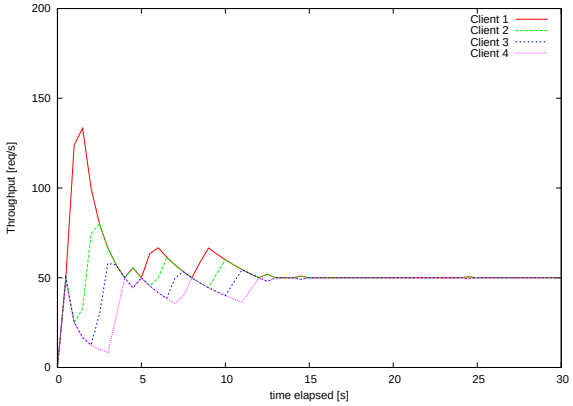

Figure 2: Time series for throughput of 4 clients, each reserving 12.5% of disk performance, but producing enough requests to saturate the disk itself.

The current model of the disk scheduler and cache are simple FIFO queues. This is being replaced by more accurate models, reflecting the status of the current implementation. Based on the results of the simulations we plan to implement the most promising flow control mechanisms as part of our overall RADI/O framework.

References

[1] S. A. Brandt, S. Banachowski, C. Lin, and T. Bisson. Dynamic integrated scheduling of hard real-time, soft real-time and non-real-time processes. pages 396–407, Dec. 2003.

[2] T. Kaldewey, T. Wong, R. Golding, A. Povzner, C. Maltzahn, and S. Brandt. Virtualizing disk performance. In *To appear in Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2008)*, 2008.

[3] A. Povzner, T. Kaldewey, S. Brandt, R. Golding, T. Wong, and C. Maltzahn. Efficient guaranteed disk request scheduling with fahrrad. In *To appear in Proceedings of the 2008 Eurosys conference (Eurosys 2008*, 2008.