Predictable and Guaranteeable Performance with Throughput, Latency, and Firmness Controls in Buffer-Cache

Roberto Pineiro, Scott Brandt
Computer Science Department University of California, Santa Cruz
{rpineiro, scott}@cs.ucsc.edu

Large storage systems are often shared among a mix of services with different performance needs, some having latency or throughput needs. These services may tolerate different various levels of performance degradation below certain performance goal, while others may not be able to tolerate any degradation. Providing and maintaining performance guarantees requires coordination at each component along the request path. This paper focuses on guaranteeing performance at the buffer cache.

Buffer caches are often seen as unpredictable because the performance is dependent upon previous accesses, and because of asymmetric execution times for read/write operations. This research aims at providing pedictable and guaranteeable, performance for services sharing a storage system. By providing controls over the throughput, latency and the firmness of the performance reservation, our buffer cache can support a mix of services with a wide range of performance needs ranging from hard performance to best-effort.

Several projects have provided service class differentiation [2, 3] by providing isolation, enabling a mix of services with different performance needs to share the system. However, the isolation provided by most of these approaches is soft, most of them lack of latency and throughput controls for each individual service, or they are mainly focused at scheduling requests at the disk, neglecting buffer management. Most of these approaches fail to provide the guarantees needed by hard real-time services that might share the system as well, or the predictability needed by mission critical backups, for example.

Enforcing hard performance guarantees requires more than just assuming worst-case execution times, but also requires careful design and coordination of each component. It is this last characteristic that sets us a part from other research. Contrary to other projects, we use device's time utilization, as opposed to metrics like bandwidth, to manage devices along the request path. Using rate–period schedulers, such as Fahrrad [1], our system enforces hard isolation and performance guarantees needed for hard performance needs by enforcing devices' time utilization and accounting for intra-streams seeks. In our research we revise buffer allocation and buffer management policies to preserve and enhance the performance guarantees provided by these schedulers.

Figure 1 shows the performance of five services sharing the system: two soft reservations for throughput-based streams; two soft reservations for periodic streams (with average execution time of 5 ms and 20 ms); and one hard reservation for a periodic stream. We measured the throughput achieved by each stream and averaged over two periods, then compared the performance of Linux using CFQ (left) and compared against the performance of our system (right). In our system, the performance of each stream is isolated from each other and provided according to its performance reservation and firmness. As expected, the services with soft performance reservations ocassionally miss their performance goal. However, the service with hard performance reservation receives predictable and consistent performance at all times.

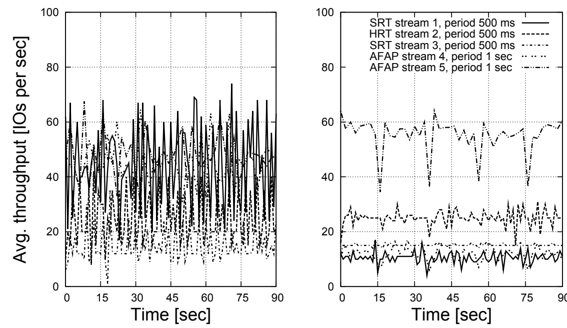

Figure 1: Average throughput achieved by 5 individual streams. One periodic HRT, two periodic SRT streams, assuming 3 ms and 15 ms average execution time and two throughput-based reservations serving greedy (as-fast-as-possible) best-effort I/O streams.

References

[1] A. Povzner, T. Kaldewey, S. Brandt, R. Golding, T. M. Wong, and C. Maltzahn. Efficient guaranteed disk request scheduling with fahrrad. *SIGOPS Oper. Syst. Rev.*, 42(4):13–25, 2008.

[2] M. Wachs, M. Abd-El-Malek, E. Thereska, and G. R. Ganger. Argon: performance insulation for shared storage servers. In *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*, pages 5–5, Berkeley, CA, USA, 2007. USENIX Association.

[3] T. M. Wong, R. A. Golding, C. Lin, and R. A. Becker-Szendy. Zygaria: Storage performance as a managed resource. In *RTAS '06: Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 125–134, Washington, DC, USA, 2006. IEEE Computer Society.