Latency Management and Quality of Service in Storage Systems

Rodney Van Meter
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
rdv@ISI.Edu
(310) 822-1511

March 21, 1997

1 Motivation

Latency in digital data storage systems varies by twelve orders of magnitude, from nanoseconds through thousands of seconds. Current storage systems of all types (local file systems, distributed file systems, hierarchical storage management systems) attempt to provide transparent service, greatly simplifying programming. However, it is inevitable that not all data can be retrieved equally quickly, so that "programming transparency" is not equivalent to "performance transparency".

The federal government runs many information services in which latency management is or will be crucial. Very large databases of satellite images, for example, are often stored on tape in robotically controlled autochangers (called tertiary storage), because the volume of data makes keeping the data online prohibitively expensive. Hierarchical storage management (HSM) systems provide a convenient, transparent interface to this data, but without information on how long retrieval will take, choosing which images to view is a blind process. Video-on-demand services, useful for soldier training, historical and political archives, and stored scientific visualizations, are difficult to adapt to tertiary storage due to problems managing latency. Many of the government's problematic legacy databases are stored on tape and accessed with custom programs due to lack of a consistent approach to managing high-latency devices. Looking towards the future, we expect that federal agencies will rely increasingly on network-accessible information repositories, which provide service to both agency employees and clients.

An analogy can be made with the road system. All roads, whether side streets or freeways, operate in a similar fashion: cars drive in lanes, following a set of familiar, well-understood rules. This "transparent interface" greatly simplifies all aspects of managing your transportation needs. However, without maps, traffic reports and a knowledge of speed limits, it is impossible to predict how long any given trip will take, or to plan the order of sites to visit.

What would the world be like if 97% of the time you got in the car, your trip took three minutes, but, unpredictably, the other 3% of the time it took between one and six hours? Many cars would never actually reach their destination because uninformed drivers, tired of waiting for the trip to end, abort. The road system would be clogged with these pointless trips. Productivity would decline and many drivers would abandon driving altogether out of frustration.

Storage systems, especially networked and hierarchical storage management systems, are in this state today. This white paper proposes an innovation dubbed SLEDs (Storage Latency Estimation Descriptors), which will provide the road maps and traffic reports that allow intelligent utilization of both computer system resources and human beings' time.

2 Problem Statement

Quality of Service (QoS) in computer systems refers to techniques being developed to provide performance suitable to the application. In networking, control of bandwidth utilized and promised, in conjunction with parameters such as jitter, delay and buffering, is used to provide the substrate over which real-time applications such as video and audio conferencing can run effectively. In operating systems, real-time CPU scheduling techniques and multimedia file systems provide equivalent services.

One feature all QoS systems have is a reservation system that allows the application to inform the system what type of service it would like to have. The system usually responds with a simple yes/no indicating its ability and willingness to provide the service.

In hierarchical storage management systems the access time for a given page can vary from under a microsecond for RAM-cached data to milliseconds for data on hard disk, tens of seconds for data on magneto-optical disks in an autochanger or hundreds to thousands of seconds for offline material stored in manually mounted tapes. This is a span of roughly twelve orders of magnitude.

While networks are generally stateless in the sense that past activity has little impact on the ability of the network to provide future service, storage systems depend a great deal on the state of caches at many levels and the physical state of complex devices – tape position, the location of the picker in an autochanger, even the head position of a disk drive.

No standard interface exists for exchanging information about the state of such systems. Currently, it is impossible for applications, Network File System (NFS) clients and HSM systems to cooperate on the performance management of storage systems.

3 Unique Approach

In this position paper, we propose exposing certain aspects of storage state to allow applications to make their own determinations about storage access timing and ordering based on the current state of the data. This appears to be especially promising for use in hierarchical storage management (HSM) systems, where the cost to retrieve data can be extremely high. Our approach provides storage system state information via an abstraction known as Storage Latency Estimation Descriptors (SLEDs).

SLEDs are the substrate that allows different parts of a large system to exchange information about storage state. Applications, NFS servers and HSM servers will all be able to communicate utilizing a standard notation.

SLEDs will enable applications cooperating with storage systems both to predict their performance and to improve that performance by reducing the I/O load they impose on the system. The potential reduction in I/O load comes primarily from applications electing not to perform certain I/O operations, and secondarily from reduction in thrashing due to improved cache utilization by I/O reordering.

Hints in file systems have been the topic of much recent research. Hints send information from the application to the storage system to improve prefetching and caching. SLEDs invert this process, sending information from the storage system to the application to allow applications to make intelligent, informed decisions.

SLEDs will improve the utilization of HSM systems, with potential performance gains of several orders of magnitude while reducing overall system load. It is here that SLEDs offer the greatest potential gains in user convenience and system throughput.

SLEDs will be useful when browsing datasets kept in HSM systems – for example, large satellite image databases. Although web browsers can predict completion time as data arrives by knowing the size and transfer rate once data begins arriving, the latency to the first byte of data is usually unknown when the data is actually stored in an HSM system. SLEDs provide the interface that allows the storage system, web server and browser to cooperate to provide the end user the information required for informed, time-efficient database browsing.

SLEDs are expected to have a broad impact on data-driven information services, including networked databases, digital libraries and video-on-demand services. Without the performance improvements and predictions SLEDs can provide, successfully deploying such services will require ad hoc solutions to these problems.

3.1 SLEDs

SLEDs are an exposure of the file's current storage state. Using a system call, an application can retrieve information about the locations of a file's data blocks. Note that if the metadata is not memory-resident, this call may itself result in I/O being performed.

SLEDs provide:

- An open, consistent interface for storage access optimization, regardless of storage technology or location.
- Information flow about storage state from the system to applications; when combined with hints and reservations, information flows in both

directions across the system/application boundary.

- The necessary "next step" enabling technology to bring predictable performance to local, network and wide-area storage systems, supporting real-time applications and quality of service (QoS).
- A "future proof" substrate for I/O programming, because they are technology independent.
- Resource utilization improvements by allowing applications to participate actively in I/O scheduling and pruning.

The metadata is returned as an extent map with two key pieces of timing information, the expected latency and throughput. In addition, SLEDs may include an indication of the reliability level of the latency and bandwidth numbers, which should be high for local disk and low for distributed file systems, and a system state change function, which will be discussed later.

The latency includes rough estimates of the time to retrieve data from tape, when necessary. This must include some estimate of the wait time for a tape drive, robot handler, tape load and seek. This information is clearly both very dynamic and difficult to estimate accurately, and the representation of this data is an open area of research. Key difficulties in representation include characterizing the effects of a given operation on the system state.

SLEDs describe extents, so an entire file or dataset can consist of disjoint segments stored in various places. SLEDs are independent of whether the data is stored locally in cache, on hard disk or tape, or remotely in distributed file systems. SLEDs are "future proof" in that they describe time-to-data in an abstract fashion, not tied to the concepts of sequential or rotating media (tape or disk) or networks. Thus, code written once to work within the SLED paradigm will never become burdensome legacy code.

3.2 Using SLEDs

The interaction between SLEDs and applications falls into four categories: (a) applications with flexible I/O ordering that can use SLEDs to schedule I/O in arbitrary order, (b) applications that will use SLEDs to make decisions about which I/Os to perform, "pruning" the set of I/Os actually executed in the interest of completion time or cost (for those systems that charge for I/O), (c) applications that use SLEDs just to predict performance, and (d) fixed-order algorithms with no need for prediction, which will be unable to use them effectively.

SLEDs support application-controlled access patterns. They require recoding of applications in order to realize the benefits. Applications must be willing to be flexible about the order of file requests. Many database-like programs, where the order of record execution often is inconsequential, are expected to make good use of SLEDs. This will allow better use of data currently in cache, reducing the thrashing that may otherwise occur.

The Unix `find` utility is an excellent example of a utility which will benefit from being adapted to use SLEDs by being able to "prune" its I/O request tree. If SLED-aware `find` is instructed not to access any file with a latency of more than, for example, 100 milliseconds, the find will complete more quickly and with less overall system load.

For applications unable to reorder or reduce their I/O requests, SLEDs will provide only a means for estimating the performance that can be achieved, a useful feature for admissions control in real-time environments and evaluation of potential execution in multimedia file systems.

In addition to predicting data retrieval time for web browsers, real-time applications will find SLEDs' performance prediction useful. Servers can utilize performance predictions to manage movement of data among levels of a hierarchy for HSM-based video-on-demand environments. When combined with the Internet's ReSerVation Protocol (RSVP) at the application layer, true end-to-end quality of service guarantees can be achieved.

3.3 SLEDs in Different Storage Environments

SLEDs are expected to be used in several disparate types of storage environments: local file systems, HSM systems and distributed file systems. Although all of these provide similar programming interfaces, their performance characteristics are very different. The goal of SLEDs is to effectively manage this heterogeneity.

The first level of information is knowing what is in the file system cache, and what must be fetched from disk or tape, e.g. what level of the storage hierarchy data contains the data. A second-generation SLEDs implementation will understand the page replacement algorithm so that varying request sequences can be explored. Third-generation models will incorporate physical characteristics such as head position, seek times, load times, etc. SLEDs can ultimately support such functionality without a change in approach.

It is in HSM systems that SLEDs have the potential to be most effective. Reducing the I/O load on HSM-managed devices has the potential to improve the performance of not only a given application, but the entire system. Optimizing the access patterns for tape drives is closely tied to the ability to ability to predict their performance. SLEDs will incorporate such information, allowing applications to explore different request sequences.

In distributed environments, SLEDs will provide the mechanism that various components can use to exchange information about storage state. A web server running on a host with an HSM system mounted via NFS can obtain estimates of data retrieval time. The web server process requests information from its local file system, which in turn requests the information from the HSM system.

SLEDs for distributed file systems will require real-time network protocols and techniques such as RSVP to create a complete end-to-end quality of service. This will create, effectively, NFS with the guaranteed I/O rates now provided by multimedia file systems.

4 Qualifications

The Information Sciences Institute of the University of Southern California is one of the nation's leading university-based information-processing research centers. ISI is involved in a broad spectrum of information-processing research and in the development of advanced computer and communication technologies. ISI's staff is working in the areas of software engineering, intelligent systems, VLSI, high performance computing and communications, and systems integration and packaging. The Computer Networks Division played a major role in the development of the Internet, and continues to conduct research on topics such as gigabit networking and RSVP.

Rodney Van Meter received his B.S. in Engineering and Applied Science from the California Institute of Technology in 1986, and his M.S. in Computer Engineering from the University of Southern California in 1991. He worked for USC/ISI from 1986 until 1992, holding positions in the Information Processing Center and on the MOSIS project. He rejoined USC/ISI's Computer Networks division in 1995, after three years in Japan developing high-performance mass storage technologies. Rodney has also done file systems performance characterization and systems integration for HSM systems.

He is currently developing the OS abstraction for file systems and safe sharing of network attached peripherals for the NAAAN Netstation task. Netstation allows direct transfer of data from peripherals to clients on the network, and scales aggregate bandwidth available for devices by attaching them to a switched network.

5 Implementation and Deployment Plans

In order to successfully deploy SLEDs, several technical problems must be solved, and a sufficient base of applications and knowledgable applications programmers must be built. The key problem is representing changes in system state, both those scheduled by the SLEDs-aware application and those initiated by other, unrelated clients of the storage system. In addition, for more effective use in networked environments, wider use of technologies such as RSVP is required.

SLEDs are expected to be deployed first as a performance prediction tool for use in multimedia and web servers. As experience is gained and the application base builds, SLEDs will gradually affect many data-intensive environments, such as databases.

SLEDs, although they represent a significant shift in I/O programming, can be deployed incrementally. The sophistication of applications, kernel services and device SLEDs are expected to go through several generations, all capable of coexisting.

During the course of this research, we expect to resolve a number of technical issues:

- Defining the correct set of parameters for characterizing storage latency.
- Representing the dynamic effects of I/O operations on SLEDs.
- Defining the applications programming interface that make SLEDs easy to use.
- Quantifying the performance gains SLEDs provide.

The proof of concept and maturation of SLEDs as a research endeavor is expected to require three to five years: two to three years of basic research and development, one to two years of testbed evaluation, including expansion of the supported hardware and software platform base, and one year to complete the deployment, including increasing the base of adapted applications. The core development team will be small, but cooperation with industry, university researchers and federal agencies will be necessary to develop critical applications and acceptance of the SLEDs programming paradigm among the community.

6 Conclusion

In this white paper, we have proposed Storage Latency Estimation Descriptors (SLEDs) as a means to more directly involve applications in the management of data movement in heterogeneous storage environments. SLEDs represent the latency and bandwidth to any given segment of storage, representing latency across twelve orders of magnitude. When utilized with hierarchical storage management systems, SLEDs have the potential to improve performance of applications by orders of magnitude by allowing them to intelligently control their own access patterns. SLEDs exploit the increasing imbalance between CPU and I/O device speeds by utilizing the former to improve utilization of the latter, in a device- and technology-independent fashion, so that SLEDs will remain a viable paradigm as storage technology evolves in unforeseen ways.

References

Related white papers can be found at `http://www.isi.edu/~rdv/sleds/`.