

## 3rd USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage '11)

Portland, OR  
June 14, 2011

### Panel

#### ***Big Data, No SQL, Big Problems, No Worries***

Moderator: Margo Seltzer, Harvard School of Engineering and Applied Sciences

Panelists: Mark Callaghan, Facebook; Andy Twigg, Acunu and Oxford University; Andy Gross, Basho and Riak; Alex Lloyd, Google

*Summarized by Dutch Meyer (dmeyer@cs.ubc.ca)*

The four panelists each brought lessons and observations drawn from their industrial experience in tackling large-scale data storage and processing.

Mark Callaghan, who leads the MySQL team at Facebook, spoke first about NoSQL, describing how the need for multi-master replication and write-optimized storage was pushing SQL in new directions. Rather than literally providing no SQL, Callaghan would actually prefer what he termed “SomeSQL.” He described a collection of rich features per node that would help him in practice, including secondary indexes, multiple operations per transaction, non-indexed predicates and data types, 10,000 queries per second (at one IOP each query), and high concurrency access to high contention data.

In describing other problems he'd like to see addressed, Callaghan stressed the challenge of reconfiguring storage in a production environment. Often a storage system cannot afford to restart in order to apply changes. How does one coordinate schema changes when applications must specify an access path to the data and the people who wrote the apps are no longer available? In addressing write-optimized storage, Callaghan first explained its many benefits, such as lowering the demands of random writes, simplifying hot backup and compression, and possibly making redundancy cheaper. At the same time, this write optimization introduces problems with increasing random reads and requiring file scans for compaction. The latter problem might be masked by merging backup and compaction into a single scan. In closing Callaghan stated that the world has a surplus of clever ideas but that the challenge, and our focus, should be getting things into production. He advised running a server before writing a new one, and investing heavily in support for monitoring, debugging, and tuning.

Andy Twigg, a research fellow at Oxford and founder of Acunu, has been working to optimize the kernel stack for big storage. He began by questioning the definition of big data. Today, the term might be used incorrectly to mean scale-out, Web-scale, or NoSQL, but some of the biggest data problems actually use SQL and some of the best-known NoSQL databases (such as CouchDB) don't scale out properly. To Twigg, managing big data is the process of finely balancing several huge and unstoppable forces. He drew an analogy to surfing, where harnessing the force of a wave requires knowing how it will behave; trying to work against it is futile, if not dangerous.

There are three fundamental forces involved in big data, and the first is the storage technologies being employed. One hundred dollars today can purchase 60 GB of flash storage which can deliver 40,000 operations per second, or 2 TB of magnetic disk that delivers 100 operations per second. To approach big data problems, one must devise algorithms to exploit these traits. However, this is not always straightforward, as Twigg demonstrated with a graph showing a precipitous drop in SSD performance as the device is filled near capacity. A second fundamental force is the workload, which designers must understand and characterize. Big data workloads often show high rate update and large range queries on data items of varying sizes and value. Naive algorithms and abstractions are probably suboptimal for any particular workload. Twigg's last fundamental force is scale. A social media startup might choose to pay a storage provider for specialized hardware, or to purchase many more smaller machines and scale them out. Twigg pointed to the existing literature on distributed systems as a useful resource.

Andy Gross from Basho next reflected on the state of distributed system research, and open questions for the future. Gross declared that big data is boring; the interesting problems are really distributed systems problems. Web developers today have moved from arguing over frameworks and APIs to discussing Paxos and the CAP (Consistency, Availability and Partition-tolerance) theorem.

Several factors have led to this renaissance of distributed systems. First, the cloud has simplified cost and scalability. Venture capitalists often demand that new companies use EC2 rather than scaling up their own services. Second, customers expect more availability for products, which is changing business requirements. Third, workloads are changing. Log data that would previously have been discarded is now being harnessed as a revenue generator. These advancements are disruptive to the traditional view of storage.

However, Gross pointed to several advances that show progress in the field. The Bloom language from Berkeley can perform consistency analysis and verification of order of independence. Gross's own work on Riak Core provides a generic distsys toolkit that enables experimentation and rapid prototyping. Stasis and Leveldb offer modern storage engines that are permissively licensed. Still, there are important problems left to solve. Global replication is still largely impractical, and the operation of increasingly complex systems is difficult. Formal verification methods could be developed to the point of providing some assurance of system correctness. Finally, the nuances of virtualization and the cloud likely change the assumptions underlying our systems, but we have yet to fully understand them.

Alex Lloyd from Google described how many aspects of SQL have been discarded with the move to NoSQL simply because they are hard to implement. He argued that it is time to determine what useful features an application team needs and to figure out how to provide them. For example, transactions are important to minimize the time application writers spend reasoning about concurrency concerns. Without this feature, application developers must each reason about concurrency and consistency above the storage layer. Another traditional database feature, joins, is extremely useful, despite being very difficult to scale. Application developers also need to be able to express queries concisely, rather than repeatedly querying the storage server. Other issues he brought up were compaction, conflict resolution, and performance isolation. Lloyd hopes that ultimately "we can have our features and scale them too." However, we need a scalable programming model that gives predictable performance, and unified data repositories. Today each group works with their own island of data, but they need to be able to come together in a tightly coupled system.

Panel Chair Margo Seltzer asked about the relative merits of scaling up (on a single host) versus scaling out (to many hosts). Initially, several panelists saw no difference, but as the discussion progressed some did emerge. For many, scaling up to a single very high-powered machine is an option. Furthermore, there are some differences in how scaling occurs. Paxos, for example, is not appropriate for a single host, but there are reasons to run multiple SQL servers on a single node. Peter Desnoyers (Northeastern University) asked each panelist for the most important reason to scale out. Twigg and Gross agreed on fault tolerance, while Lloyd replied that scaling up could only take a system so far. Twigg reminded the audience that for most people, there are limits to the size of a database, after which more scaling is not necessary.

Erik Riedel (EMC) recalled a comment from Alex Lloyd: “The complexity has to go somewhere [in the storage stack].” Riedel asked if we could use layering to remove complexity at the source and, if not, wondered where the complexity should go. Lloyd believes in finding common operations and integrating them into storage—bringing legal discovery tools into storage, for example, where they may also be useful to other applications. Callaghan and Gross also saw potential to hide asynchronous replication and elements of relational database in storage. Albert Chen (Western Digital) asked the panel to what degree they are concerned about power usage. Gross replied that he didn’t even think about it. Callaghan explained that people who care about power are not usually close to the database servers, and Lloyd said that he was dubious about getting predictable performance from complex power-saving systems.

## A Solid State of Affairs

Summarized by Luis Useche ([luis@cs.fiu.edu](mailto:luis@cs.fiu.edu))

### ***Don’t Thrash: How to Cache Your Hash on Flash***

Michael A. Bender, Stony Brook University and Tokutek; Martin Farach-Colton, Rutgers University and Tokutek; Rob Johnson, Stony Brook University; Bradley C. Kuszmaul, MIT and Tokutek; Dzejla Medjedovic, Pablo Montes, Pradeep Shetty, Richard P. Spillane, and Erez Zadok, Stony Brook University

Rick Spillane introduced a new probabilistic data structure, similar to Bloom filters, especially designed for solid state storage. Their work was prompted by the infeasibility of fitting Bloom filters in memory for large storage systems. With Quotient Filters, a replacement for Bloom filters, the idea is to hash the elements and use part of the key to index in an array and store the rest in the array location. This new structure has the same properties as Bloom filters, with the addition that they can be merged into bigger Quotient Filters. This last property is key to implement Cascade Filters,

a Bloom filter replacement especially designed for flash devices. Cascade Filters maintain a set of Quotient Filters (one in RAM and the rest in disk) organized in a way to allow lookups bounded by  $\log(N)$ . Cascade Filters allow writing sequentially to flash devices while still maintaining fast lookups. With this technique they achieve 40x faster insertions and 3x slower lookup throughput compared to Bloom filters.

How will collisions in the insertion operation affect lookup times? They ensure with their technique that sequential scans to QF clusters will not be more than  $\log(N)$  where  $N$  is the number of elements in the cluster; moreover, this worst case is an unlikely event. James Lentini asked whether the data structure they designed was resilient to power loss or crash. Rick answered that they can have atomicity by using a combination of COW and journaling that is not yet implemented.

### ***Onyx: A Prototype Phase Change Memory Storage Array***

Ameen Akel, Adrian M. Caulfield, Todor I. Mollov, Rajesh K. Gupta, and Steven Swanson, University of California, San Diego

Ameen Akel introduced a new prototype of SSD implemented using Phase-Change Memory (PCM), an emerging byte-addressable persistent memory. This technology takes advantage of the difference in resistance when molecule phases are changed. Current PCM outperforms flash technologies, especially in reads, making it suitable for SSD implementation. Its projected performance is three orders of magnitude faster than current SSDs. Their real data shows different results than simulators, making the case for a prototype PCM-based SSD for better estimation of performance results.

Onyx was compared with FusionI/O, a high-end SSD, and showed consistently better read throughput as request sizes increased. Ameen pointed out that PCM does not require complex FTL logics that significantly slow down flash technologies. For writes, Onyx showed better performance only in small request sizes. Ameen attributes this to the more mature flash technology heavily optimized for writes since its conception. When they ran Berkeley DB benchmarks, Onyx did not show exceptional gains. In conclusion, Ameen emphasized the potential of PCMs compared to flash due to its simplicity and because of the absence of FTL.

Andy Twigg asked whether this technology, given that it is byte addressable, will eliminate the problem of creating large requests to obtain better performance. Ameen said that these devices eliminate this problem, making the interaction between the application and the backing stores easier. Peter Desnoyers was curious what lessons they learned from constructing an experimental device like this. Ameen said

that developing Onyx was difficult but worth it, as it renders better results than the simulations used for previous studies. Irfan Ahmad was concerned about what problems PCM technology might have before it can be commercially available. Ameen said that the main concern with PCM technology is whether it will be able to scale in size as flash has been doing in recent years. Irfan asked what interfaces other than ROM are currently available. Ameen suggested that DIMM interfaces will be a big step forward because they will make PCM easier to use. Any ideas for future work? It was important to investigate better interfaces and the impact this technology would have in application performance. Peter wondered whether they felt confident that PCM will replace flash as the choice of SSD. Ameen expressed high confidence in PCM's future.

### ***SSD Characterization: From Energy Consumption's Perspective***

Balgeun Yoo, Youjip Won, Seokhei Cho, and Sooyong Kang, Hanyang University, Korea; Jongmoo Choi, Dankook University, Korea; Sungroh Yoon, Korea University, Korea

Youjip Won stressed the importance of understanding the internals of SSDs. He mentioned that disk characterization has been done for decades and has allowed the design and implementation of many of the important improvements available today. Now the question is, what measurements can be used to characterize the SSD? Given the electronic nature of SSDs, they decided to characterize based on energy consumption. SSDs have multiple channels to communicate with the NAND chips. SSD logic usually maximizes parallelism by using as many channels as possible to increase performance. In this paper they focused on how the channels in the SSD are used to service every request.

They started the characterization by measuring the power consumption of the SSD while increasing the request size. They found peaks that indicate how many channels are used. Moreover, the duration of the peaks give an estimate of how long the channels are activated to service the request. Just for comparison, 16 KB and 32 KB request sizes showed same duration but different peak sizes. This indicates an increase in the number of channels involved when the request size is doubled from 16 KB to 32 KB. On a different example, 256 KB and 512 KB request size showed the same power consumption but with 2x difference in the duration of the peak. Youjip also showed the tradeoffs between parallelism and the peak power consumption of the device: with higher parallelism comes a higher peak power consumption. High peaks cause problems such as supply voltage drop, signal noise, and blackout. For this reason, they propose a technique called Power Budget that will maximize the parallelism as long as the peak power is held below the specified maximum. Youjip

ended by highlighting the usefulness of power consumption in characterizing SSDs.

How willing are manufacturers to disclose internals of SSDs? Youjip said that more important than the number of channels, which is usually available, companies should standardize the way the power is reported to upper layers. Theodore Ts'o asked why the peak power consumption was more important in the paper when it is not as representative as the area under the curve. Youjip replied that peaks were relevant because they can accidentally turn off the machine if not controlled. Irfan Ahmad asked whether additional features could be extracted with this technique. Youjip said that unfortunately there are some SSDs that do not show clear behavior, limiting the scope of the power consumption characterization. Irfan wondered whether information from the FTL could also be extracted. Youjip replied that FTL is complex and they do not know how it works. However, they can use comparisons to find which type of FTL is more energy efficient. Finally, Peter Desnoyers wondered whether this technique could leak information that was not intended to be public. Youjip thought there was no relation between the power consumption and the information in the SSD.

### ***Exploiting Heat-Accelerated Flash Memory Wear-Out Recovery to Enable Self-Healing SSDs***

Qi Wu, Guiqiang Dong, and Tong Zhang, Rensselaer Polytechnic Institute (RPI)

Qi Wu predicted a bright future for SSDs, which are lowering their price while continuing to grow in size. Moreover, they are the perfect candidates for high-performance applications. Unfortunately, NAND flash chips, the most popular technology behind SSDs, suffer from a limited number of program/erase (P/E) operations, and that limits their lifespan. Interface traps are one of the causes for NAND flash failures. In previous work, researchers found that NAND flash can recover from interface traps, because they heal faster when heat is applied.

In this presentation, they proposed a new self-healing SSD architecture that recovers flash chips from interface trap failure. The basic idea is to include a small heater on every chip that will be used once the number of P/E cycles are close to the limit. Qi mentioned that while the heating process is in progress the data in the chip under recovery cannot be accessed. For this reason, they add one additional chip on every SSD channel, to be able to migrate the data before applying heat. While the backup operation occurs, the chips attached to the channel in use cannot be accessed. Qi said that they found a tradeoff between latency, backup time, and copying granularity: Faster and higher copying granularity leads to higher latencies.

They set up a multi-component simulator to evaluate their new architecture. In their simulation they found that their architecture can result in a fivefold increase in SSD lifespan. On the downside, this architecture could increase the latency of I/Os up to 15% compared to commodity SSDs.

Peter Desnoyers asked whether they have implemented a real prototype of this architecture. Qi said that they are relying on real implementations in previous works. Peter then asked what type of market will embrace this technology. Qi replied that any type of write-intensive application can benefit from this technology, since it will increase the life of their backing stores. Somebody asked why they did not try the experiment of heating a commodity SSD and checking its lifespan. Qi replied once again that they are relying on previous work. Moreover, Peter added that such an experiment will also heat the controller, which can ultimately damage the SSD.

## A Coterie Collection

Summarized by Sejin Park ([baksejin@postech.ac.kr](mailto:baksejin@postech.ac.kr))

### **Italian for Beginners: The Next Steps for SLO-Based Management**

Lakshmi N. Bairavasundaram, Gokul Soundararajan, Vipul Mathur, Kaladhar Voruganti, and Steven Kleiman, NetApp, Inc.

Datacenters' increased system complexity arising from service automation needs causes low operational and management efficiency. Gokul Soundararajan laid out current datacenter trends: the move from a siloed to a shared world to improve resource utilization; increased configuration complexity (e.g., RAID level, dedup) in which the impact of combining these technologies is very hard even for the system administrator to understand; huge scale, which requires a large number of administrators to manage the datacenter; and dynamic applications, requiring administrators to understand dynamic resource requirements. To handle all this, the datacenter industry provisions for peak demand and hires a lot of administrators.

Gokul said the solution is automated management with service level objectives (SLOs). SLOs are specifications of applications' requirements in technology-independent terms, and their attributes are performance, capacity, reliability and availability, and security and compliance. The MAPE (Manage, Analyze, Plan, Execute) loop is used to achieve automated management. However, SLOs are slow to be adapted because, among other reasons, it is difficult to specify SLO requirements. He suggested focusing on process, not product, through the use of pre-defined SLOs (Qualified SLOs) as a way to manage various systems simply and reliably.

Someone asked how Qualified SLOs provide support if someone's system isn't working properly. Gokul emphasized that customer support is one of the benefits of Qualified SLOs.

### **In Search of I/O-Optimal Recovery from Disk Failures**

Osama Khan and Randal Burns, Johns Hopkins University; James Plank, University of Tennessee; Cheng Huang, Microsoft Research

Traditionally, systems are made reliable through replication (easy but inefficient) and erasure coding (complex but efficient). Because storage space was a relatively expensive resource, MDS codes were used to achieve optimal storage efficiency with fault tolerance. However, time and workload have changed and the traditional  $k$ -of- $n$  MDS code would require  $k$  I/Os to recover from a single failure. Osama Khan addressed this problem and suggested a new way to recover lost data, with minimal I/O cost, that is applicable to any matrix-based erasure code.

Osama claimed that enumerating all decoding equations is not an easy job and finding a decoding equation set with minimal I/O is the challenge. He transformed this into a graph problem and used Dijkstra's shortest path algorithm. He also explained that GRID code is an I/O-efficient recovery code. GRID code allows two (or more) erasure codes to be applied to the same data, each in its own dimension. With the GRID code, the author could combine the STAR (for high redundancy) and Weaver (for low I/O) codes and make an I/O-efficient code.

Someone asked about their approach to the variety of other erasure codes. Osama recognized the presence of alternative erasure codes besides the traditional Reed Solomon code and said the technique they used is applicable to all types of erasure codes that can be represented in matrix form. Someone asked whether CPU utilization had been considered, since erasure coding is CPU-intensive. Osama replied that CPU utilization was not part of their study; they focused, instead, on measuring I/O for recovery.

### **ViDeDup: An Application-Aware Framework for Video De-duplication**

Atul Katiyar, Windows Live, Microsoft Corporation, Redmond; Jon Weissman, University of Minnesota Twin Cities

Atul Katiyar talked about the kinds of redundancy in large-scale storage systems and said that the redundancy is managed if the storage system is aware of it and replication is performed for specific goals such as fault tolerance or improved QoS. The redundancy is unmanaged when the storage system is unaware of it and it merely acts as an overhead on the storage system. The system views redundancy as

an identical sequence of bits. However, the application-level view of redundancy is a little different, defined as a metric that gauges redundancy at the content level with the flexibility to define and hence tolerate noise in replica detection as dictated by the application. Atul gave examples of videos encoded in different formats, frame resolution, etc.

Atul said that large-scale centralized Web storage is an emerging trend and there is a significant degree of unmanaged redundancy in such storage systems. Application-level redundancy can significantly reduce the storage space by its deduplication. The ViDeDup system is an application-aware framework for video deduplication, and it detects similarity among contents. The framework provides application-level knobs for defining acceptable noise during replica detection. He enumerated various aspects in which near-duplicate videos differ. They leveraged the research of the multimedia community in adapting, modifying, and integrating existing approaches for video similarity detection into the framework. In contrast to system-level deduplication, in ViDeDup the choice of which of the two duplicate replicas to store is not trivial. They propose a centroid-based video deduplication approach, where the centroid video is the representative video of good quality in the cluster, against which remaining videos of the cluster are deduped. Atul presented an algorithm for centroid selection which balances the tradeoff between compression and quality within the cluster.

Someone asked whether this is lossy compression and how this technique compares with other video compression techniques. Atul said it uses lossy compression; standard mpeg compression looks intra-file, while this compression seems more inter-file. In some interesting key contexts, such as in-cloud service wherein uploading video is for dissemination, it might make sense to tolerate loss. Peter Desnoyers expressed doubt about whether this compression can really work. Atul proved it with his demo video of two videos having the same basic nature, compressed to result in a final video. Someone asked how long it took to process the data set. Atul said compression of 1017 videos took two hours, but the system-level deduplication processed in 15–20 minutes.

## A River of Data, More or Less

### *Truly Non-Blocking Writes*

Luis Useche, Ricardo Koller, and Raju Rangaswami, Florida International University; Akshat Verma, IBM Research—India

### *Exposing File System Mappings with MapFS*

Jake Wires, Mark Spear, and Andrew Warfield, University of British Columbia

### *Stratified B-trees and Versioned Dictionaries*

Andy Twigg, Andrew Bye, Grzegorz Miłoś, and Tim Moreton, Acunu; John Wilkes, Google and Acunu; Tom Wilkie, Acunu

No reports are available for this session.

## Invited Short Talks and Wild Ideas

Summarized by Dutch Meyer ([dmeyer@cs.ubc.ca](mailto:dmeyer@cs.ubc.ca))

### *Using Storage Class Memory for Archives with DAWN, a Durable Array of Wimpy Nodes*

Ian F. Adams and Ethan L. Miller, University of California, Santa Cruz; David S.H. Rosenthal, Stanford University

Adams made the argument that the research community should consider Storage Class Memory (SCM) as an archival storage medium. SCM refers to a class of technologies, including Flash, PCM, Memristor, and others, that have a high cost-to-capacity ratio, but offer much higher storage performance than magnetic disk, especially for random reads. These characteristics are not usually associated with archival storage, but Adams pointed out several ways in which the total cost of ownership for SCM may actually be lower than magnetic disk.

He began by reviewing current technologies for archival storage. Hard drives have a lower initial purchase cost, but magnetic disk is mechanically complicated. Large racks of disks are heavy to the point that they may require reinforced flooring. They also are vibration and shock sensitive and require a great deal of power to operate. Tape is even denser, requires more maintenance and cleaning, and has poor random access performance. It also doesn't scale well, in that a cost-benefit analysis may show tape-based storage to be a poor value in surprising ranges of storage capacity. Alternatively, Adams imagines an array of simple low-power SCM-based system boards he calls DAWN. Such an architecture could be scalable, power-efficient, and largely self-managing, as each unit would be responsible for its own integrity checks, and be replaceable. Adams suggested that this approach may provide a lower total cost of ownership, but he said that more investi-

gation is needed to characterize current and future SCM and to do the necessary cost analysis.

Erik Riedel (EMC) asked whether backups could be left on the shelf without any maintenance. Adams replied that they see a wide variety of customer workloads, ranging from write-once, read-maybe to high-frequency scrubbing and error checking. Riedel suggesting looking closely at disk drive technology, as the failure rate for a disconnected drive likely approaches that of disconnected SCM. Peter Desnoyers from Northeastern joined the presenter in calling for more research into the effects of temperature on archival storage and into whether even the best device would survive for very long lifetimes. Several attendees asked what range of devices should be considered, and the answer covered a broad range of archival options, including S3 and paper printouts.

### ***Principles of Operation for Shingled Disk Devices***

Garth Gibson and Greg Ganger, Carnegie Mellon University

Less than half of the audience for Ganger's talk had heard of shingled disks. This new technology will lead to higher capacities in magnetic disks, but not without introducing some new performance effects. Instead of writing each track with gaps in between, in a few generations disks will write tracks that overlap. One consequence is 1.5 to 2.5 times more density, but it also means that one cannot overwrite old data without erasing newer data. Firmware could theoretically hide this behavior, as is done in flash devices, but the resulting read-modify-write cycle is 1,000 to 10,000 times longer than that of Flash. Ganger believes that this would be impractical, and instead proposes to explore an interface to let the software above the device manage its peculiarities. Such software could minimize in-place modification through a log structure, very large block sizes, or a hierarchy of performant hardware, such as flash or PCM. Ganger closed by stressing the new questions that the introduction of this technology will raise. Researchers must determine the right interface for this storage, how best to exploit features and dodge costs, and what role firmware will play in managing the device.

Session chair Anna Povzner (IBM Research) asked if shingling is an appropriate capacity/performance tradeoff. Ganger replied that this is an inevitability. While we may have the choice of not using shingled regions of the disk, that would be abandoning the order-of-magnitude capacity increase. Geoff Kuenning (Harvey Mudd) asked if there was any hope of getting to a place where we don't have to keep rewriting systems for each new technology. Ganger argued that there is enough difference between devices that it's not clear that we'd want to generalize our storage software.

Albert Chen (Western Digital) said that they had considered many of the options discussed in the session, and that in the short term, drives will not need much support from the software above them. He and the presenter agreed that hints may be a method of striking the right balance between compatibility and specialization.

## **Panel**

### ***Storage QoS: Gap Between Theory and Practice***

Moderator: Carl Waldspurger

Participants: Greg Ganger, Carnegie Mellon University; Kaladhar Voruganti, NetApp; Ajay Gulati, VMware; Ed Lee, Tintri

*Summarized by Dutch Meyer (dmeyer@cs.ubc.ca)*

Moderator Carl Waldspurger posed questions for both the panelists and the audience. First, do users and administrators want to specify QoS in terms of predictable performance, or service level objectives (SLOs)? Even if we had an answer, it's not clear we'd know how to quantify QoS; we might measure latency, IOPS, transactions per second, or other metrics. Enforcing these metrics is also challenging, because performance isolation is challenging and the storage stack is complex. There's also a tension between delivering QoS and performance.

Ajay Gulati (VMware) took the position that storage QoS will be pervasive in the next five years. His vision is driven by virtualized environments giving rise to the need for per-VM controls and will be made possible through the deployment of SSD. Current systems use deadline-based scheduling or CFQ, or one of the virtualization-specific schedulers such as SFQ or mClock. Mostly, these schedulers are based on proportional allocation, which doesn't give applications much of a guarantee. The lack of guarantees is because of the efficiency versus fairness tradeoff, because the metric isn't clear, and because customers are scared off by worst-case performance numbers. To solve these issues, he proposes that QoS be defined in terms of latency, perhaps up to a specified number of IOPS. To meet this latency bound, arrays of SSDs large enough to fit the working set of applications could offer reliable performance.

Kaladhar Voruganti (NetApp) reframed the QoS problem in terms of SLO. His model consists of three parties: a storage vendor who delivers features, a storage provider who creates a service catalog with quality tiers, and a storage subscriber who orders a particular level of service. Managing an SLO is preferable for a subscriber because they understand application requirements, not the effects of storage system latency. Still, there are problems with current approaches. SLO specification remains difficult: the complexity of stor-

age makes management models hard to create and must be made to cross management layers. Kaladhar sees promise in combining proactive approaches to SLO management, such as application-specific templates for performance, with reactive approaches like hybrid flash and disk systems that can minimize the impact of an incorrectly specified SLO.

Ed Lee (Tintri) offered another definition of QoS: it avoids user complaints without spending a lot of money or time. He pointed out that users don't actually notice fairness, but they do notice performance inconsistency, and will complain about slow-downs. Fairness, consequently, is less useful than performance consistency. Lee's presentation proceeded to point to a number of current problems in QoS. First, the technologies are all built by different vendors, and each will develop their own notions of QoS. It's appropriate to build QoS at each level, but mechanisms need to be able to work together. Storage systems are large and complex, with many components that can affect performance. Furthermore, constraints must be specified in aggregate. Finally, Lee made the case for building rational systems that have no performance cliffs, are consistent over time, and provide a simple, transparent model of their behavior.

Greg Ganger's presentation explored the gap, which he referred to as a "chasm," between QoS theory and practice. First, he explained how the theoretical assumption of starting with a clear SLO specified by a customer is flawed. Humans, even experts, are bad at expressing their needs in terms of performance. In practice, one usually guesses by choosing from broad tiers of quality, and reacts to performance problems as they arise. Theory might also dictate that workloads should be admitted based on demand, but this assumes that demand on the system is predictable. In practice, this kind of stability can only be seen in a very large window of time. Since workload varies both in intensity and in characteristics like locality, it is very difficult to predict what the actual system demands will be for any workload.

To make matters worse, one might assume that device load could be determined by the workload. However, in practice, the observed load given a device and a workload is often not repeatable. There are many sources of variability, including interference between workloads, internal maintenance, and device retries, where the disk initially fails to complete a request. Even differences between two devices of the same apparent make and model can skew workload results. The good news for QoS advocates is that no one is expecting perfect results. If the current approach is to start with one of a small number of service tiers and then respond to complaints, perhaps we can make that process faster and easier.

The first question addressed the different views on what metric QoS should use. Lee said that the chief difficulty is that different applications need different metrics. Youjip Won (Hanyang University) noted that the more general QoS problem has been around for a decade, but that storage has a much larger range of variability. He asked whether turning to SSD to solve this problem was just another form of over-provisioning. Several members of the panel acknowledged that flash brings new problems, but it does make some issues (such as random read latencies) easier to solve. The overall customer experience can be improved in embracing SSD.

This conversation sparked a discussion about SSD latencies. According to Lee, SSD can have latencies much higher than disk under some conditions, although Gulati suspects that only lower-cost SSDs would exhibit this behavior. Peter Desnoyers (Northeastern University) confirmed that he had seen an iSCSI interface time out while connected to an SSD under a stress test. He wondered what level of performance isolation is ultimately possible. Lee and Ganger explained that nearly perfect performance isolation is possible, even under multi-tenancy, by giving different applications different time quantum, but in practice we want more than isolation. Goals like cost-efficiency, high performance, or performance reliability interfere with isolation.

Desnoyers also asked if SLO specifications could be made for each application or at each level of the storage stack. Lee responded no. Even a very well designed system would specify aggregate SLOs, ideally automatically so that more time could be spent managing user expectations. Alex Lloyd (Google) thought that it seemed it would be years until we could hope to describe SLO for a single user/single tenant environment, so perhaps it would be better to push for wider interfaces with more control. Lee agreed that vendors should expose more of their system state and suggested they could provide hooks with reasonable defaults that naive customers could ignore. He hoped that this would lead to consensus around a standard model. Ganger was more pessimistic about companies agreeing on standards.

Irfan Ahmad (VMware) noted that someone shipping through FedEx gets several service options, and you get an SLO that you can characterize, manage, and buy insurance around. He wondered if we should stop focusing on the tails of the performance curve and instead focus on the 80th or 90th percentile. Ganger, Lee, and Ahmad discussed some of the potential benefits of keeping requests in-buffer to smooth the variability in performance. It would make performance predictable and also give administrators the ability to easily increase performance in response to customer complaint.