End-to-End Performance Management for Large Distributed Storage

Scott A. Brandt
Associate Professor
Computer Science Department
University of California, Santa Cruz

Background and Motivation

- Storage is traditionally purely best-effort
  - Trade-off between fairness and performance
- High-performance computing requires performance guarantees from the storage subsystem (and elsewhere)
  - Real-time data capture
  - High-performance simulation
    - I/O should not be the bottleneck
    - Isolation is required
      - Transfer, rebuild, …
    - Synchronization matters
  - Visualization
- Goal: Develop flexible end-to-end mechanisms to provide QoS guarantees in the storage subsystem

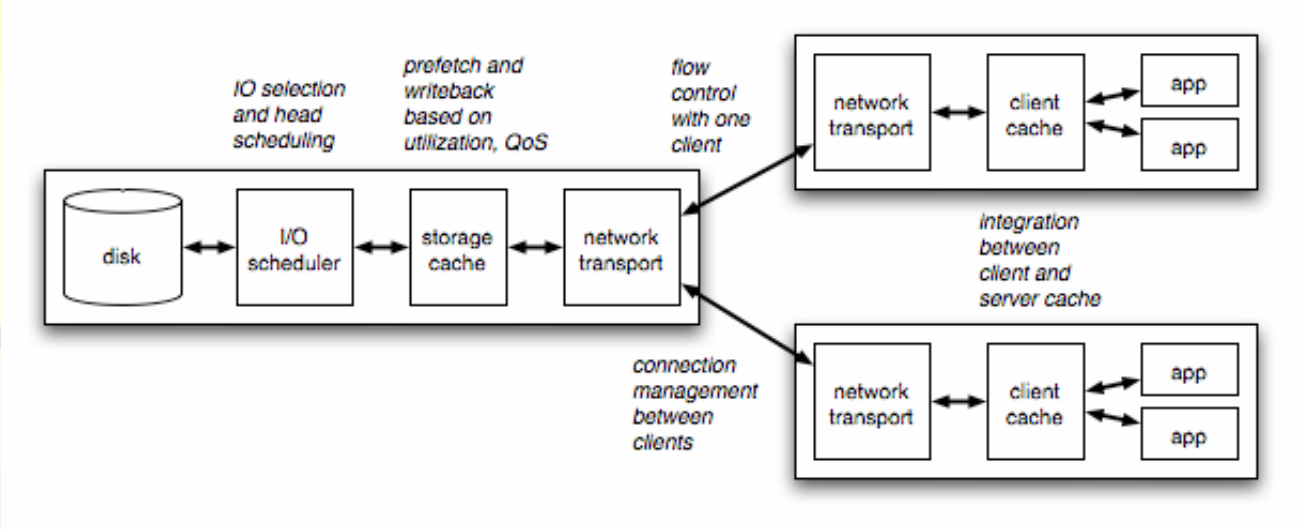
Programmatic Details

- Collaboration between UC Santa Cruz and IBM Almaden Research Center
  - UCSC: Scott Brandt, Darrell Long, Carlos Maltzahn
  - IBM: Richard Golding, Theodore Wong
- ~$1 million/3 years
  - 3–4 students + faculty/staff + equipment
- Integrated with the Ceph object-based storage system
  - Linux-based
  - Results will be publicly available

Real-Time Computing Primer

- Many applications require performance guarantees
  - Flight control, defense systems, medical devices, multimedia, device drivers, *etc*.
- Real-time research
  - Hard real-time: critical timing requirements
  - Soft real-time: non-critical timing requirements
  - Firm real-time: some deadlines can be missed
  - Quality of Service: average performance guarantees
  - Lots of point solutions, mostly CPU and network scheduling
- Mixed-mode processing with real-time and non-real-time processing is becoming common
  - Desktop multimedia, device drivers, automotive systems, …
  - Hierarchical: RT-Linux [Yodaiken], HLS [Regehr]
  - Integrated: RAD/RBED [Brandt], VRE [Goddard]
- Some research on real-time/QoS storage
  - DAS [Reuther], AQuA [Wu, Brandt], Zygaria [Wong, Golding]

Problem Definition



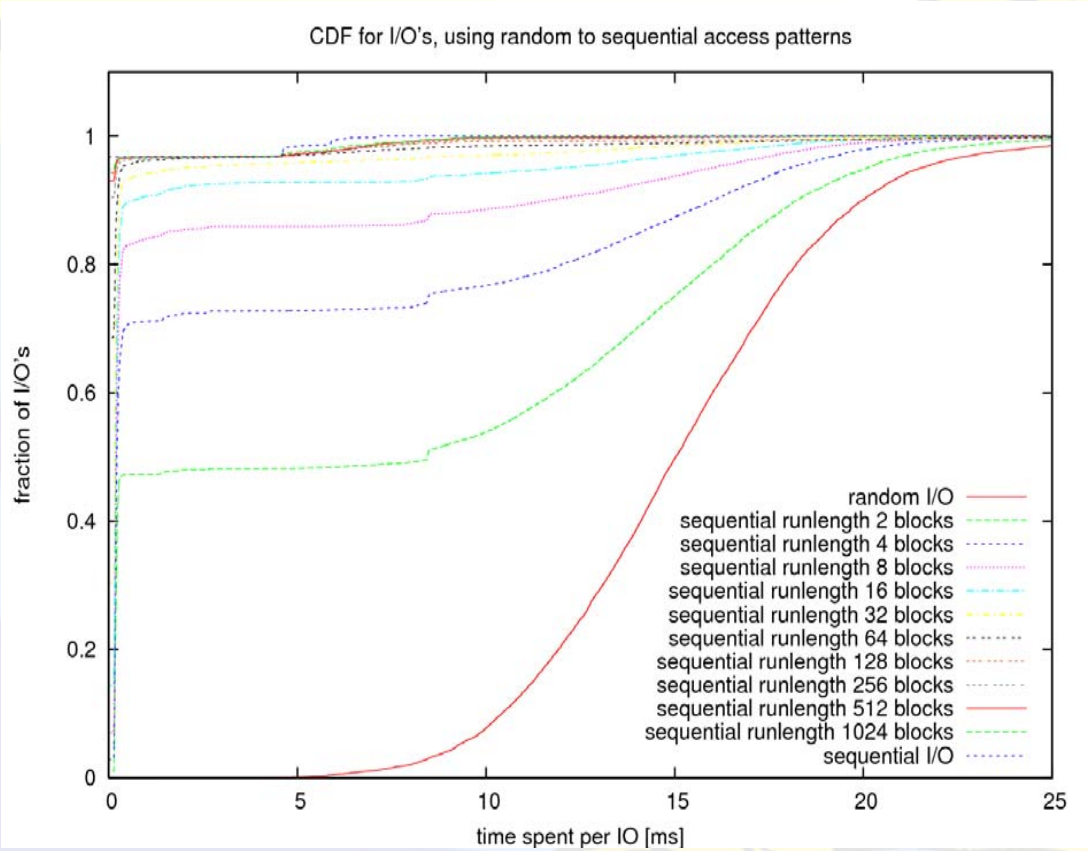
- The I/O path has many components
  - Shared among many clients
  - All must support the performance guarantees
- Need
  - Performance guarantees (hard, soft, firm, QoS, best-effort)
  - Isolation
  - Performance
  - Fairness?

Approach

- Disk scheduling (*In progress*)
  - Apply mixed-mode real-time and QoS principles to disk scheduling
  - Goals: Performance guarantees, isolation, overall performance
  - Challenges: Stateful, non-preemptable, variance between best/average/worst case, incomplete *a priori* knowledge
- Cache management (*Year 2*)
  - Use cache to improve performance guarantees
  - Modify caching/prefetching based on performance requirements, system behavior
- Flow control (*Year 3*)
  - Incorporate network QoS to provide end-to-end guarantees

Preliminary Results: Estimating Time Required for Disk I/Os

- Building new disk scheduler based on RBED CPU scheduler
- Investigating appropriate metrics: I/Os, bytes, utilization



- CDF of request times for random 20 GB disk