QoS support for Intelligent Storage Devices

Joel Wu          Scott Brandt
Department of Computer Science
University of California Santa Cruz
ISW 04
Mixed-Workload Requirement

- General purpose systems today expected to handle heterogeneous workloads
  - best-effort
  - soft real-time (multimedia)

- Existing systems employ best-effort resource management

- Requirements are met by over-provisioning

- Lots of research on mixed-workload CPU scheduling
  - Processor Capacity Reserve
  - Hierarchical Scheduling
  - SMART
  - Rialto
  - RBED

- Having QoS-aware CPU scheduler alone is not sufficient
Many soft real-time applications are **storage-bound**
- Large data needs
- Must access storage continuously and timely
- Toleration of some missed deadlines

Storage may become bottleneck

Storage can dictate progress of soft real-time applications

Question: How to support storage-bound SRT applications in a mixed-workload environment?

Lots of research on real-time storage
- **Disk scheduling**
- Data Placement
- Admission Control
- File System
Most work on QoS for direct-attached storage focused on the use of QoS-aware disk schedulers.

Disk scheduling: ordering of disk requests
- balance between response time and throughput
- access time = seek time + rotational latency + transfer time
- Exploit geometric property of disk

More detailed knowledge allows more aggressive utilization

Disk scheduling is NP-Complete, stateful, and non-preemptable.

Three types of disk schedulers:
- Best-Effort: SCAN, LOOK, C-SCAN/LOOK, V-SCAN
- Real-Time: EDF, SCAN-EDF
- Mixed-Workloads: Cello
Issues with disk scheduling

- Effective scheduling of disk requests with deadlines require fine-grained knowledge of disk drive internals
  - Disk model needed for accurate prediction of service time.
  - Accuracy of model determines effectiveness of scheduler.

- Disk profiling: Required parameters must be extracted from the disk drive

- Trends in drive design
  - Increasing intelligence and autonomy
  - Encapsulation of internal complexities
  - Evolving interface
**Problems**

- Challenges faced by external disk scheduler [Lumb02]:
  - Coarse observation
  - Onboard caching
  - Autonomous internal disk activities
  - Rotational offset
  - Drive internal scheduling

- Complexity of external disk scheduler increases

- Disk drives are becoming intelligent and autonomous units, but fine-grained external disk schedulers still try to retain control over every step of its operation

- Fine-grained external disk scheduling possible now, but may become infeasible in future as drives become ever more intelligent

- Alternative way to provide QoS for storage?
Traffic Shaping: Concept

- Our Proposal: Traffic shaping above external disk scheduler
- By adjusting resource usage, best-effort scheduler can provide reasonable soft real-time performance if resource usage is less than 100% [Brandt98]
Traffic Shaping: Detail

- Bandwidth management above external disk scheduler
  - Coarse-grained view.
  - Treats disk drive as a black box
  - Can work with any underlying best-effort disk scheduler
  - Simple. Clean.
  - Independent of disk properties

- Components
  - Bandwidth Control Mechanism – traffic shaping
  - Policy: reservation/feedback based
Traffic Shaping: Implementation

- Use token bucket filter (TBF), a technique for traffic shaping in networking, to control disk bandwidth.

- Differentiate disk requests:
  - Best-effort (BE)
  - Soft real-time (SRT)

- Associate disk request with type of issuing process
  - Explicit: API
  - Implicit: BEST heuristic

- BE pipe: aggregate disk traffic from all BE processes
- SRT pipe: aggregate disk traffic from all SRT processes
Missed Deadline Notification

- Give SRT requests preferential handling by throttling the rate BE processes can issue requests

- When to throttle BE request rate?

- Missed Deadline Notification (MDN)
  - Signify the inability to access data on disk in time.
  - Reduce BE pipe size in order to boost SRT pipe size.
Missed Deadline Notification

- Soft real-time processes
- Best-effort processes
- Missed Deadline Notification
- External disk scheduler
- Disk

Background
Motivation
Traffic Shaping
Results
Increasing token rate

- MDN decreases BE pipe size and increases SRT pipe size.
- Also need a way to decrease SRT pipe size and increase BE pipe size

- Two options considered:
  - Optimistic: Continuously increase token rate over time.
  - Pessimistic: Increase only when the aggregate SRT bandwidth changes.
Implementation

- Implemented on Linux 2.6
- One TBF and wait queue for each block device request queue
- Kernel thread to handle token replenishment
- API for SRT declaration and MDN
- Associate request with issuing process.
- Synthetic application for testing
Normal Linux behavior

Four 8 MB/s streams – no boosting
Fixed token rate

Four 8 MB/s streams - stream 1 boosted, cap BE token rate (90 req/s)
Fixed token rate

Four 8 MB/s streams - stream 1 boosted, cap BE token rate (50 req/s)
Feedback based - optimistic

Four 8 MB/s streams – stream 1 boosted

Background
Motivation
Traffic Shaping
Results
Feedback based - pessimistic

Four 8 MB/s streams – stream 1 boosted

Throughput (MB/s) vs Time (s) graph showing SRT boosted CR stream 1 (8 MB/s), CR stream 2 (8 MB/s), CR stream 3 (8 MB/s), and CR stream 4 (8 MB/s).
Result (1) – Total Throughput

Max throughput of disk ~ 27.59 MB/s for sequential read

<table>
<thead>
<tr>
<th>Method</th>
<th>Total Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>No boosting – normal Linux behavior</td>
<td>16.80 MB/s</td>
</tr>
<tr>
<td>Fixed token rate – 90 tokens per second</td>
<td>17.25 MB/s</td>
</tr>
<tr>
<td>Fixed token rate – 50 tokens per second</td>
<td>14.71 MB/s</td>
</tr>
<tr>
<td>Feedback-based, optimistic</td>
<td>15.85 MB/s</td>
</tr>
<tr>
<td>Feedback-based, pessimistic</td>
<td>14.48 MB/s</td>
</tr>
</tbody>
</table>

- Traffic shaping can actually increase total throughput in some situations
- Pessimistic method is more aggressive at boosting SRT stream, resulting in less total throughput
Conclusion

- Future intelligent disks invalidate the use of external disk schedulers for QoS

- Traffic shaping is a feasible alternative
  - Avoids complexity of fine-grained disk scheduling
  - Feedback scheme requires no *a-priori* knowledge on resource requirements
  - Small loss of throughput