



Dynamic Adaptivity in Support of Extreme Scale

Towards the Incorporation of Dynamic Adaptation into Operating Systems: Adaptive Disk I/O

Patricia J. (Pat) Teller
Professor, Computer Science
The University of Texas at El Paso
pteller@utep.edu

DASES Outline

Dynamic Adaptivity in Support of Extreme Scale

- **Brief Introduction to DASES Research**
- **Adaptive Disk I/O**
 - **How can experiments on small systems help w.r.t. extreme-scale systems?**
 - **What kinds of adaptations are applicable to disk I/O management?**
 - **How can adaptive disk I/O management help in terms of performance and productivity?**
- **Future Work**

Enhanced Performance



Generalized → Customized

Resource Management

Fixed → Dynamically Adaptable

OS/Runtime Services



Dynamic Adaptivity in Support of Extreme Scale

Small-scale MPs  Extreme-scale MPs
Commodity OS  Dynamically
Adaptable OS

- Currently developing prototypes for small-scale MPs running Linux
- Looking forward to applying DAiSES research to compute nodes and I/O nodes of extreme-scale systems

System	Compute Nodes	I/O Nodes	Ratio
SNL Intel Paragon	1,840	32	58:1
ASCI Red	4,510	73	62:1
Cray Red Storm	10,368	256	41:1
BG/L	63,536	1,024	64:1

Dynamic Aadaptivity in Support of Extreme Scale

Small-scale MPs  Extreme-scale MPs
Commodity OS  Dynamically
Adaptable OS

- Currently developing prototypes for small-scale MPs running Linux
- Looking forward to applying DAiSES research to compute nodes and I/O nodes of extreme-scale systems
 - Performance isolation research is applicable to I/O servers
 - I/O stream throttling research is applicable to improving the performance of applications with I/O programmed similarly to MADbench

DAISES Challenges

Dynamic Adaptivity in Support of Extreme Scale

Introduction

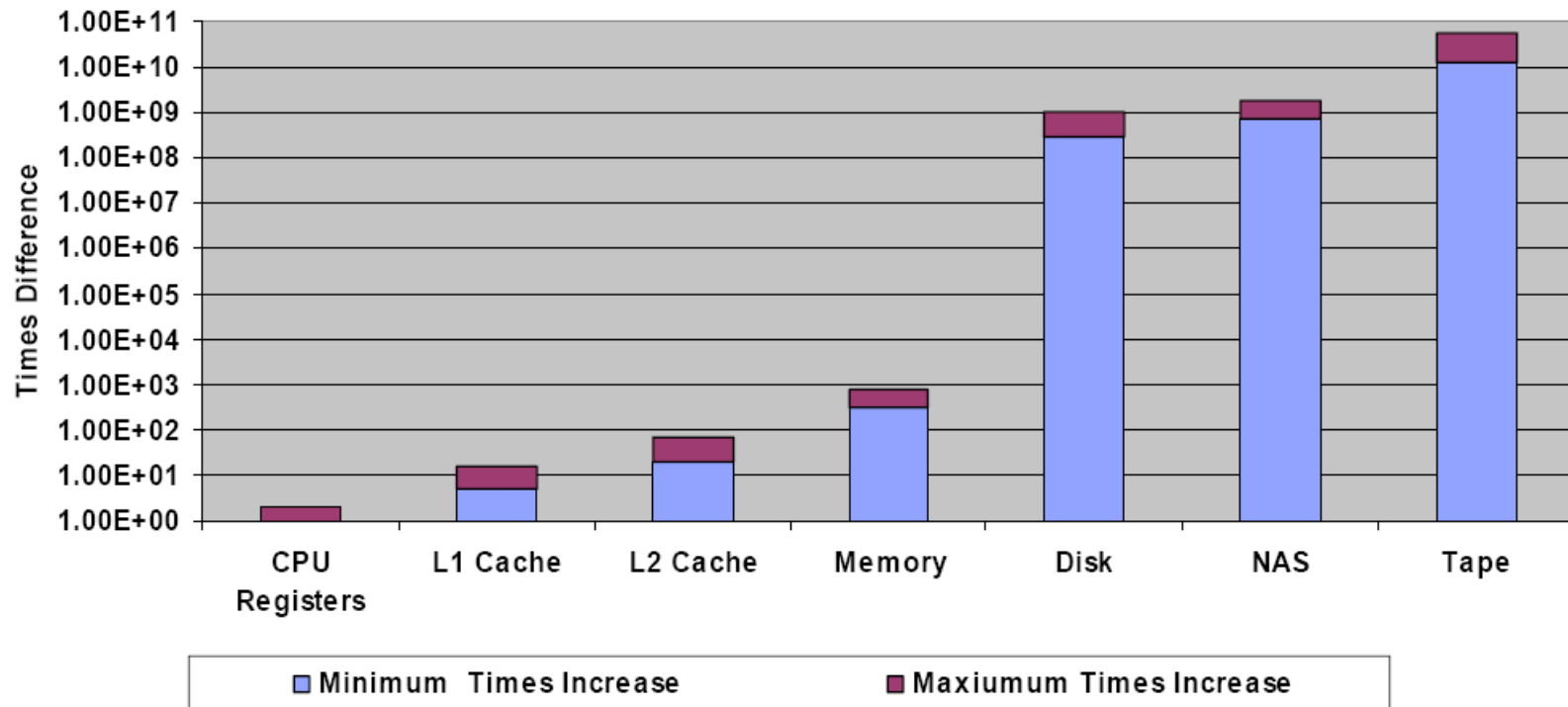
Determining

- **What** to adapt
 - policy
 - parameter value
- **When** to adapt
 - under what circumstances
 - definition of a heuristic, function, or table-driven decision map
- **How** to adapt
 - selection of policy or parameter value
 - mechanism to affect the adaptation
- **How** to measure effectiveness of adaptation
 - metric



Dynamic Adaptivity in Support of Extreme Scale

Disk I/O cause for concern? adaptation target?



From HEC I/O Workshop , August 2005: HPCS I/O and Storage Issues,
David Koester (Mitre) and Henry Newman (Instrumental)

Disk I/O Performance is dependent on ...

- I/O request access times: seeks + rotational latencies
 - I/O request generation pattern
 - Number and type of concurrent I/O-generating processes/threads
 - Disk I/O scheduler: policy and parameters
 - Application I/O requirements, e.g., latency, utilization, fairness
 - I/O controller: order in which requests are serviced by disk
- Data transfer rate
- File system



Dynamic Adaptivity in Support of Extreme Scale

Disk I/O Performance what can be adapted?

- I/O request access times: seeks + rotational latencies
 - I/O request generation pattern
 - Number and type of concurrent I/O-generating processes/threads
 - Disk I/O scheduler: policy and parameters
 - Application data delivery requirements, e.g., latency, utilization, fairness
 - I/O controller: order in which requests are serviced by disk
- Data transfer rate
- File system

Disk I/O Performance

what can be adapted?

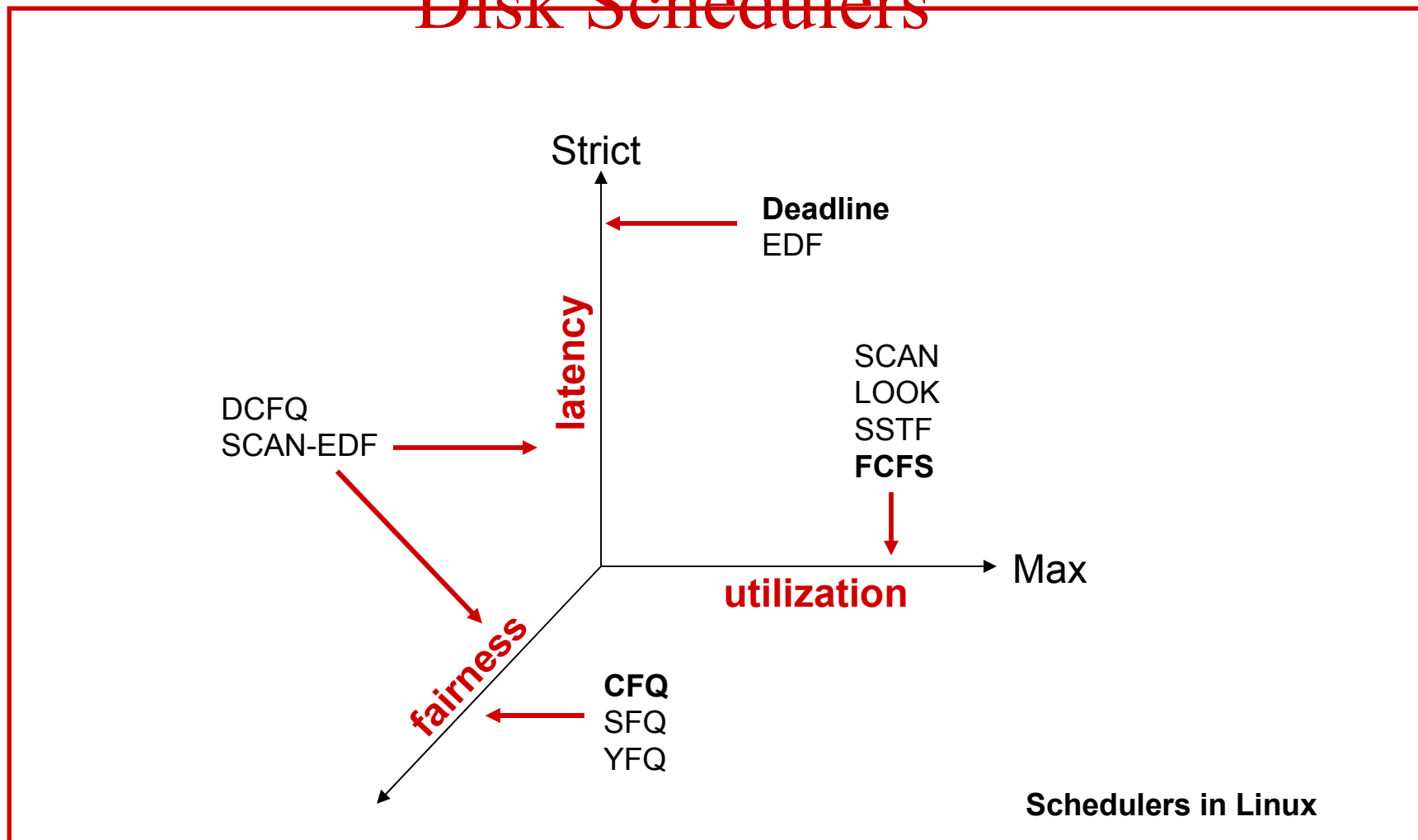
- I/O request access times: seeks + rotational latencies
 - I/O request generation pattern
 - Number and type of concurrent I/O-generating processes/threads
 - **Disk I/O scheduler: policy** and parameters
 - Application data delivery requirements, e.g., latency, utilization, fairness
 - I/O controller: order in which requests are serviced by disk
- Data transfer rate
- File system

Space of Data



Dynamic Adaptivity in Support of Extreme Scale

Delivery Requirements and Disk Schedulers



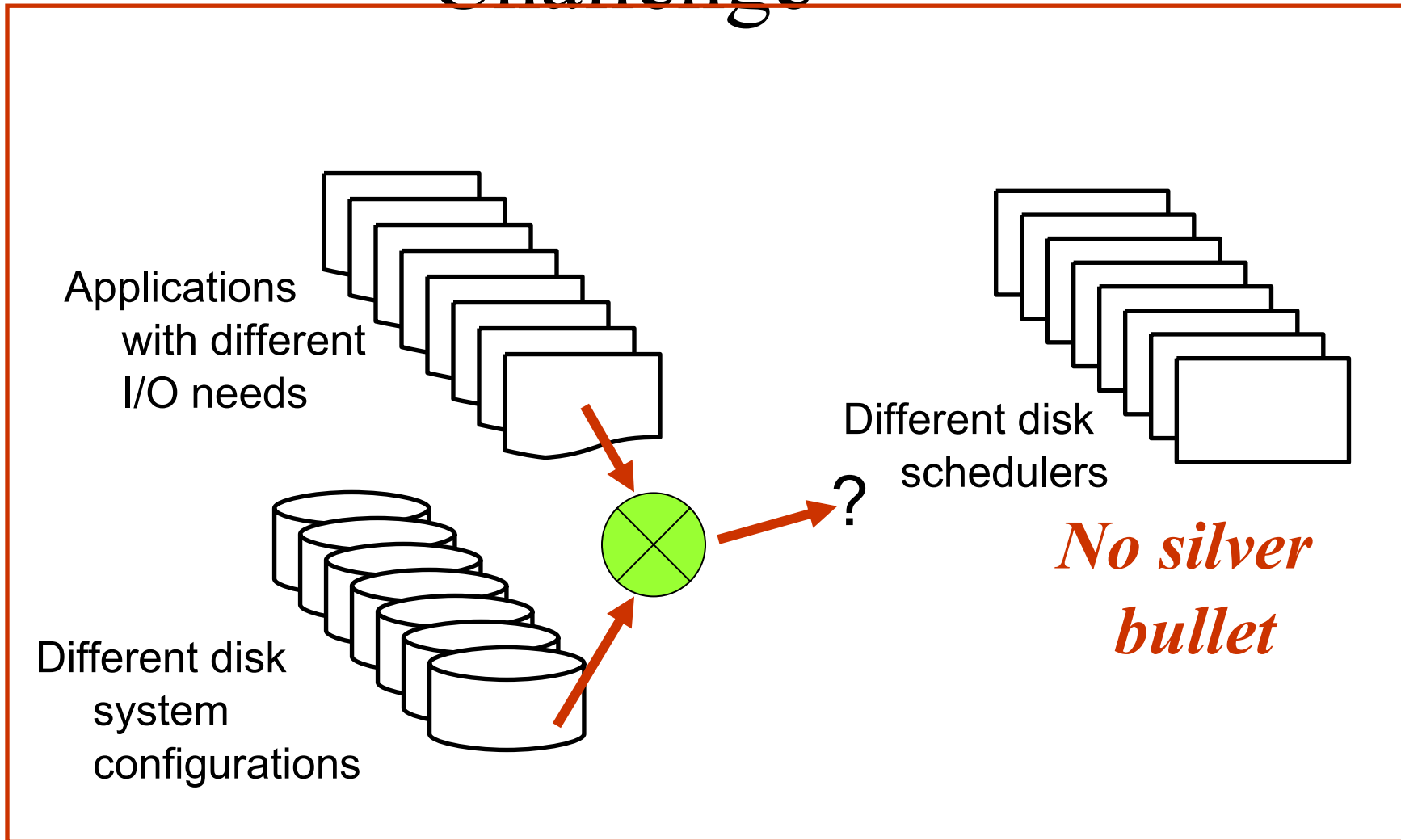


Dynamic Adaptivity in Support of Extreme Scale

Adaptation of Disk Schedulers

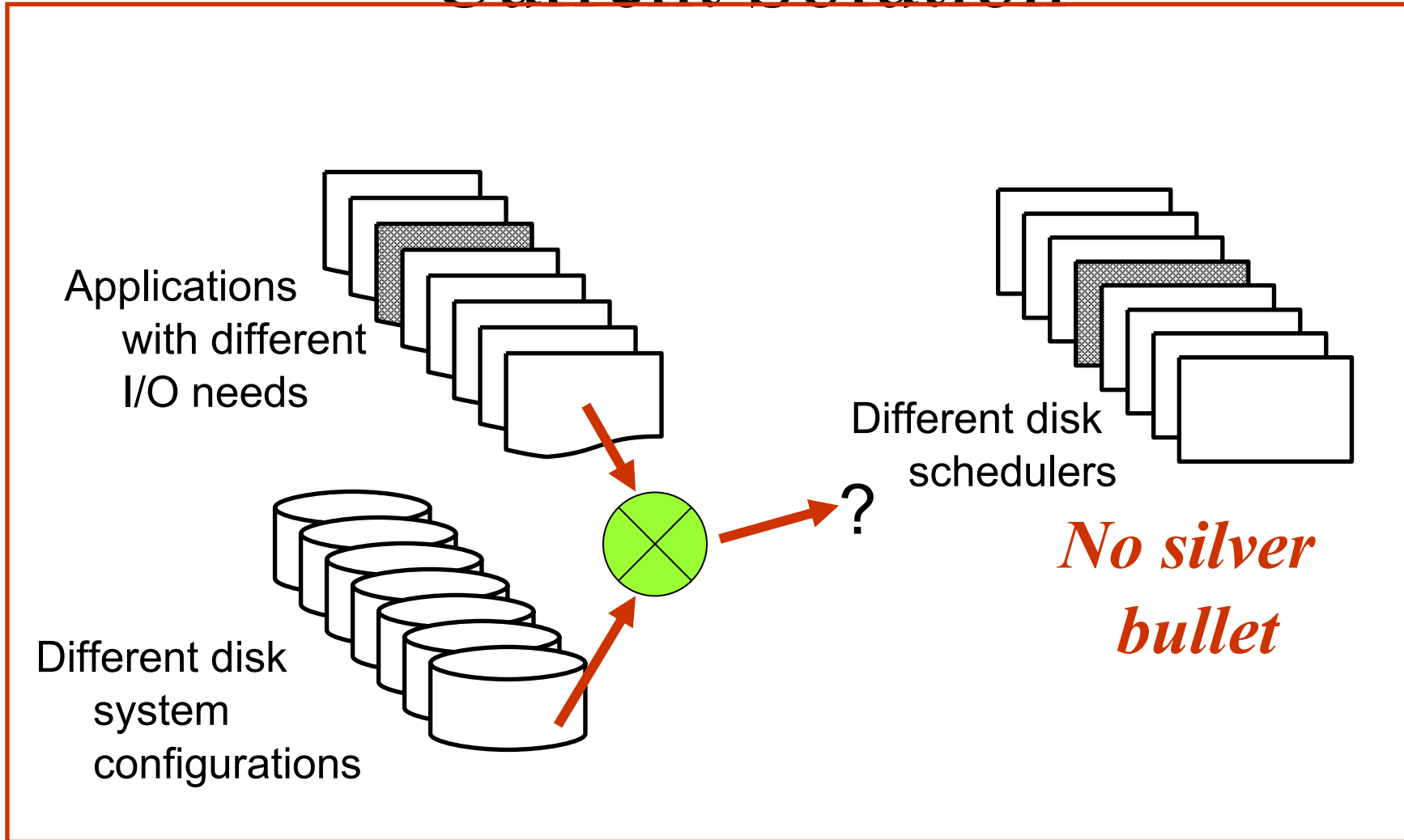
- Linux 2.6 includes four schedulers (Anticipatory, Deadline, CFQ, and noop) with boot-time and run-time selection
- Match the scheduler to application or system data delivery requirements
 - Real-time database → Latency guarantees: Deadline
 - Multimedia database → Fairness: CFQ
 - Throughput/disk utilization: SSF

Disk Scheduling Challenge



Disk Scheduling

Current Solution





Dynamic Adaptivity in Support of Extreme Scale

Linux Solution Shortcomings

- Only one scheduler active at a time
- Queue draining time
- Does not provide
 - fair allocation of disk resource among multiple concurrently executing applications with different data delivery requirements
 - I/O performance isolation / insulation

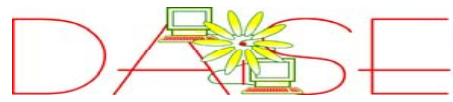


Dynamic Adaptivity in Support of Extreme Scale

A Case for Adaptive Disk Scheduling

- Operating systems and storage systems service concurrently executing applications with different data delivery requirements
- There are various I/O schedulers that satisfy these requirements
- No single scheduler is likely to satisfy all the applications' data delivery requirements
- Server consolidation and virtualization compound this problem

New Solution:

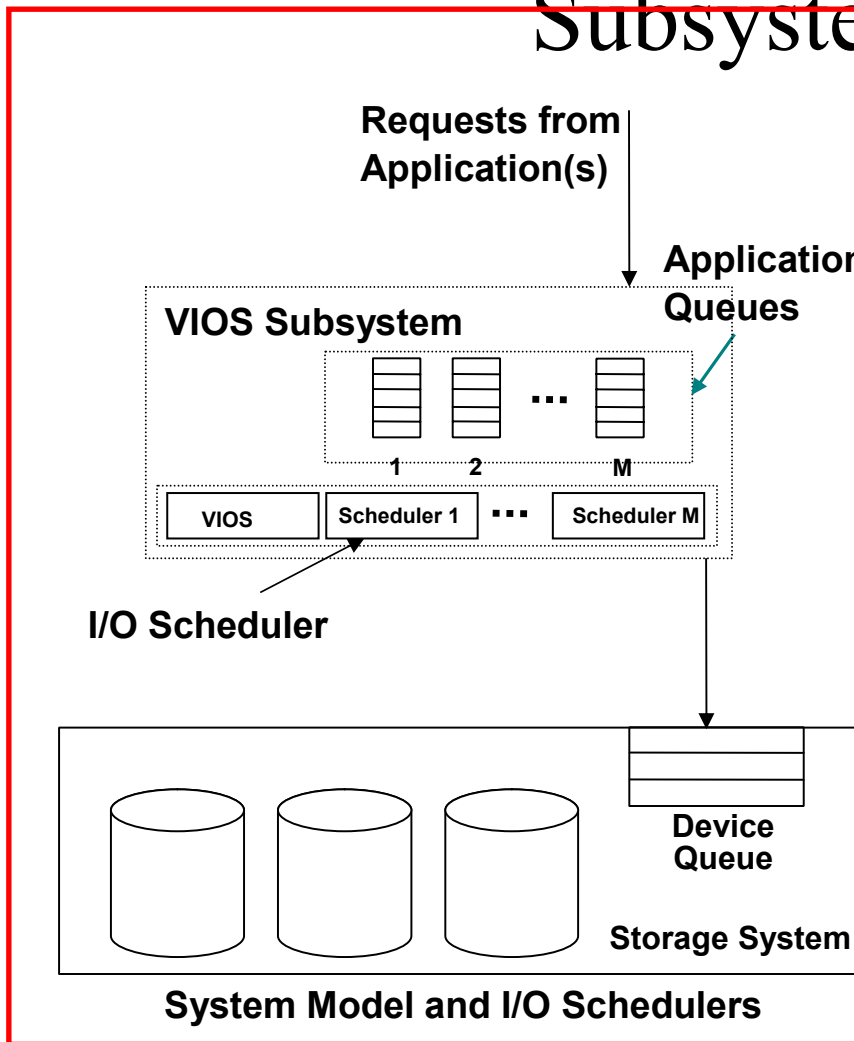


Dynamic Adaptivity in Support of Extreme Scale

Virtual I/O Scheduler (VIOS)

Subsystem

Disk Scheduling



Scheduler of Disk Schedulers

- A queue is associated with each application
- Each application queue is associated with an instance of an I/O scheduler that can meet its data delivery requirements
- The VIOS distributes I/O service quanta among the scheduler instances (applications) via a round-robin algorithm

DASES Fairness and VIOS

Dynamic Adaptivity in Support of Extreme Scale

Disk Scheduling

- How can fairness be achieved?
- What is fairness in disk scheduling?
- What does VIOS achieve by providing fairness?

DASES Fairness and VIOS

Dynamic Adaptivity in Support of Extreme Scale

Disk Scheduling

- How can fairness be achieved?
 - Using quanta as in round-robin process scheduling
- What is fairness in disk scheduling?
 - Depends on definition of quantum
 - Number of requests dispatched
 - Number of bytes transferred
 - Amount of allocated disk time
- What does VIOS achieve by providing fairness?

DASES Fairness and VIOS

Dynamic Adaptivity in Support of Extreme Scale

Disk Scheduling

- How can fairness be achieved?
 - Using quanta as in round-robin process scheduling
- What is fairness in disk scheduling?
 - Depends on definition of quantum
 - Number of requests dispatched
 - Number of bytes transferred
 - **Amount of allocated disk time**
- What does VIOS achieve by providing fairness?

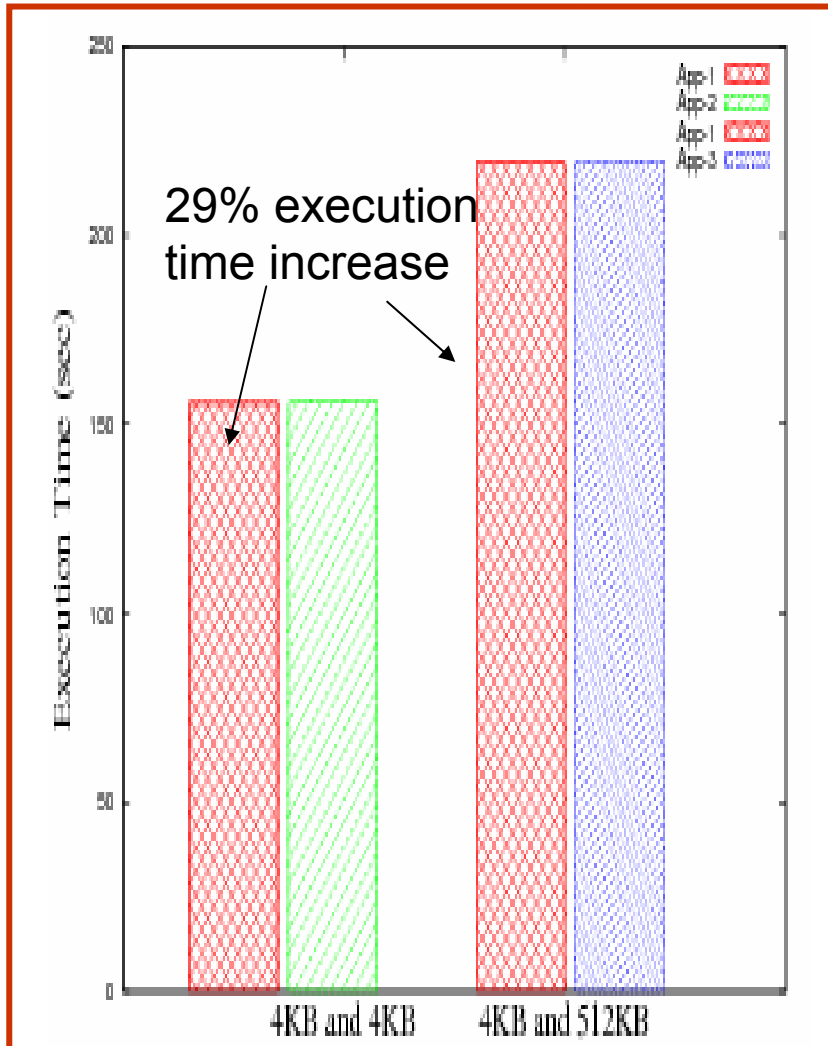


Disk Service Quantum

Dynamic Adaptivity in Support of Extreme Scale

Number of Requests - Example

Disk Scheduling



- Applications essentially only generate I/O read requests
- App 1 and App2 generate a total of n 4KB requests each time they run
- App 3 generates a total of n 512KB requests
- **Service is fair in terms of number of requests but it is not fair in terms of disk time allocated**
- **Unpredictable execution time** – it depends on the I/O characteristics (size of requests and seek characteristics) of the application with which it runs
- **No performance insulation**

Our Solution:

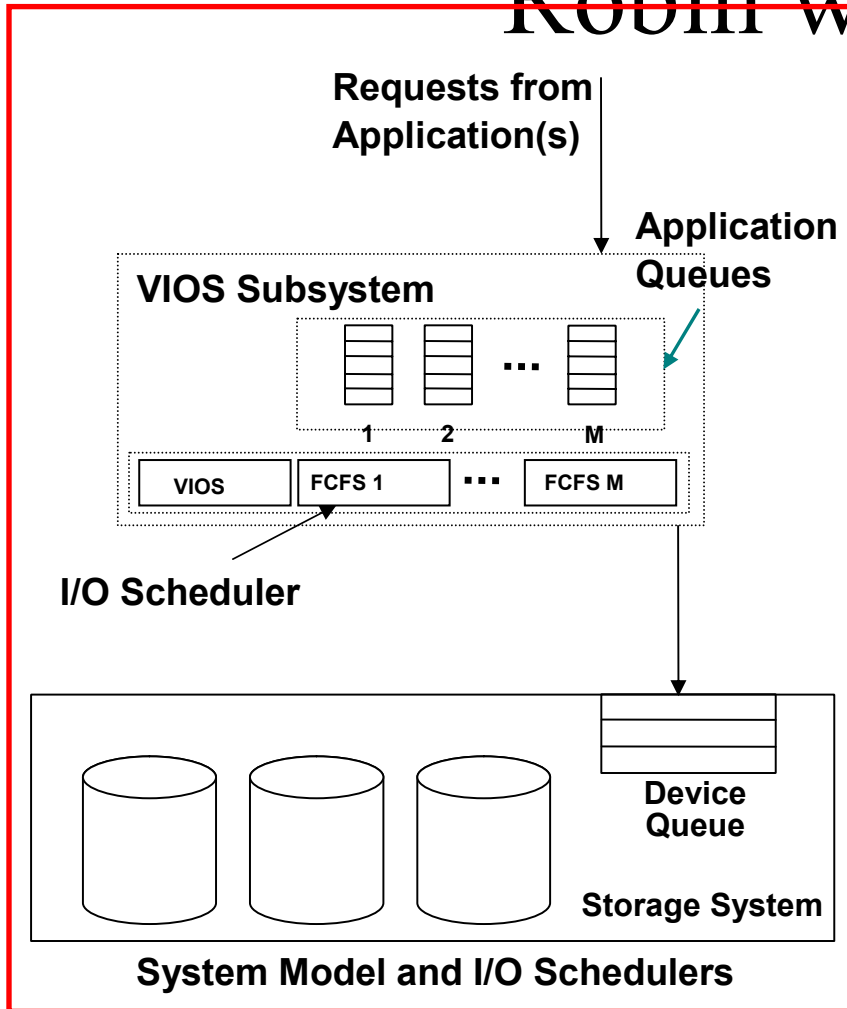


Dynamic Adaptivity in Support of Extreme Scale

Compensating Round

Robin w.r.t. Disk Time

Disk Scheduling



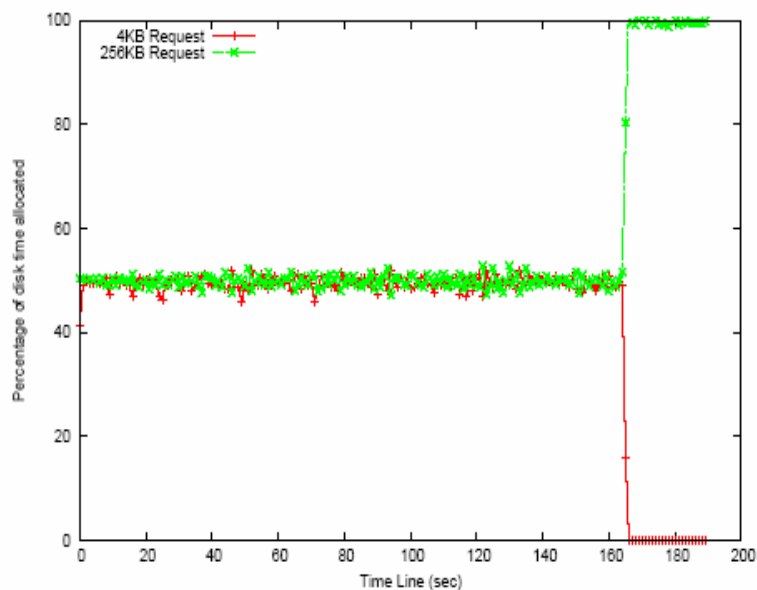
Provides disk-time fairness

- As requests are dispatched and serviced, request service time is subtracted from the quantum
- If remaining quantum ≤ 0 , then
 - no more requests are dispatched from the queue and
 - the quantum of the next round is shortchanged by the over-compensation of this round
- Unused quantum in a round is not carried forward

VIOS Subsystem

Fairness given Different

Request Sizes



VIOS Disk Access Time Partitioning among Applications with Different Request Sizes

Fairness in terms of disk-time allocation

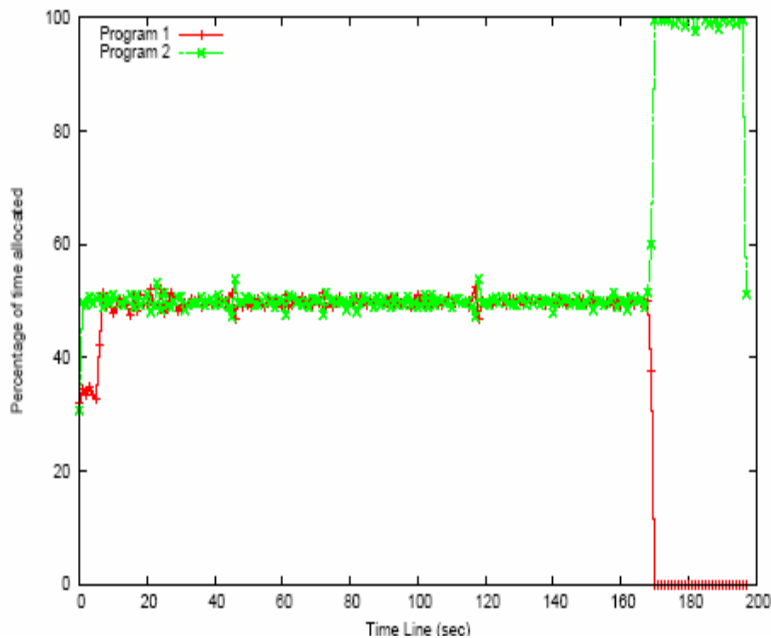
- Each application consists of eight threads to ensure queues with enough requests to consume quantum
- One application performs 4KB reads and the other performs the same number of 256KB reads
- Since each queue has the same quantum, irrespective of the request sizes, approximately 50% of disk time is allocated to each application



Dynamic Adaptivity in Support of Extreme Scale

VIOS Subsystem

Fairness given Different Seek Characteristics



VIOS Disk Access Time Partitioning among Applications with Different Seek Characteristics

Fairness in terms of disk-time allocation

- Each application consists of three threads to ensure queues with enough requests to consume quantum
- One application performs a 1M sector seek for each read while the other performs a 64M sector seek for each read
- Since each queue has the same quantum, irrespective of the seek characteristics, approximately 50% of disk time allocated to each application

DASES Fairness and VIOS

Dynamic Adaptivity in Support of Extreme Scale

Disk Scheduling

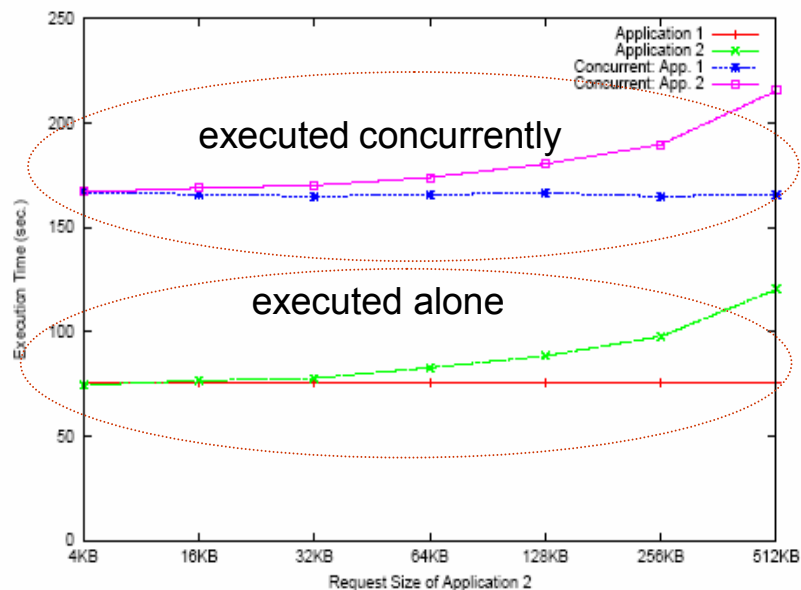
- What is fairness in disk scheduling?
 - Round-robin scheduling
 - Quantum: amount of allocated disk time
 - CFQ-CRR
- **What does VIOS achieve by providing fairness, i.e., what are the benefits of providing fairness and are there penalties as well?**



Dynamic Adaptivity in Support of Extreme Scale

Performance Isolation given Different

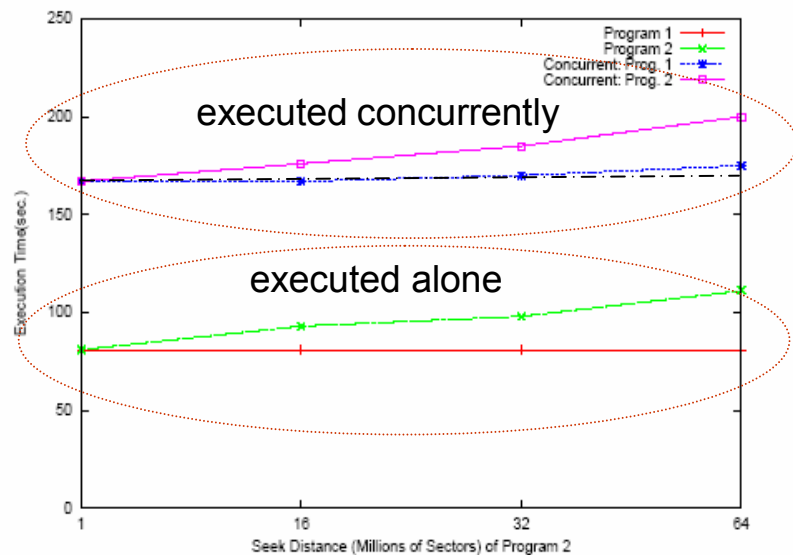
Request Sizes



VIOS Execution Times of Different Instances of Applications 1 and 2. Application 1 always has a Fixed Request Size, while the Request Size of Application 2 Varies.

- Each instance of Application 1 has a fixed 4KB request size
- Each instance of Application 2 generates a different fixed request size in the range of 4KB to 512KB
- The execution times of both applications increase due to concurrent disk usage
- Note that the execution time of Application 1 is not affected by the I/O characteristics (request size) of Application 2

Predictable Performance given Varying Seek Characteristics



VIOS Execution Times of Different Instances of Programs 1 and 2. Program 1 always has a Fixed Inter-request Seek Distance, while that of Program 2 Varies.

- Each instance of Program 1
 - generates only 4KB requests
 - has a fixed inter-request seek distance of 1M sectors
- Each instance of Program 2,
 - generates only 4KB requests
 - has a fixed inter-request seek distance in the range of 1M to 64M sectors
- The execution times of both applications increase due to concurrent disk usage
- Note that the execution time of Program 1 is not affected by the I/O characteristics (seek characteristics) of Program 2



Dynamic Adaptivity in Support of Extreme Scale

Implicit Benefits of Providing Disk-Time Fairness

- Different applications can be given different quanta, i.e., disk-time allocations \Rightarrow Performance Differentiation
- Disk-time fairness results in
 - **Predictable disk-time allocation**
 - Given the number of non-empty queues, the inter-service time interval for any queue is known within a specified bound
 - Unpredictability hinders disk performance guarantees
 - Can support QoS
 - **Disk performance insulation / isolation**
 - The I/O characteristics of one application cannot impact the disk time allocated to another application
 - An application cannot monopolize the disk system
 - Not provided by contemporary disk schedulers



Dynamic Adaptivity in Support of Extreme Scale

Adaptive Disk I/O opportunities - 1

- Performance improvements: but what does performance mean?
 - Improved utilization / decreased execution time / throughput
 - Checkpoints: schedule sync and asynch requests differently: 80% of I/O usage associated with checkpoints (asynch)
 - Match disk I/O scheduler to application data delivery requirements
 - Throttle number of concurrent I/O-generating processes / threads



Dynamic Adaptivity in Support of Extreme Scale

Adaptive Disk I/O opportunities - 2

- Performance improvements: but what does performance mean?
 - Provide fairness, performance isolation, performance predictability
 - Virtualized I/O
 - Service guarantees / contracts
 - Repeatable performance
 - Load balance in terms of I/O

- Performance improvements: but what does performance mean?
 - Provide fairness, performance isolation, performance predictability
 - Virtualized I/O Can increase execution time
 - Service guarantees / contracts
 - Repeatable performance
 - Load balance in terms of I/O

- Performance improvements: but what does performance mean?
 - Provide fairness, performance isolation, performance predictability
 - Virtualized I/O
 - Service guarantees / contracts
 - Repeatable performance
 - Load balance in terms of I/O
- Facilitates performance debugging / tuning => PRODUCTIVITY

Virtualized I/O Performance vs. Productivity

WITHOUT



PRODUCTIVITY: application development time (including performance debugging / tuning)



PERFORMANCE: application execution time

*productivity improvements are not always
performance improvements*

WITH



PRODUCTIVITY: application development time (including performance debugging / tuning)



PERFORMANCE: application execution time

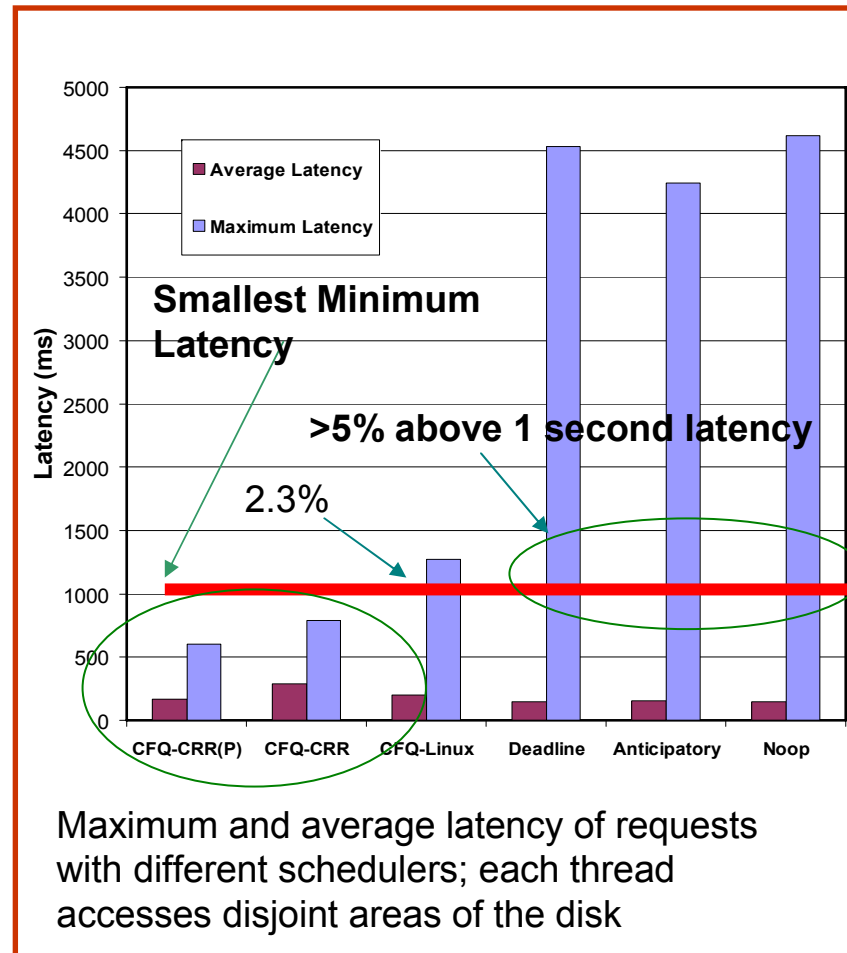
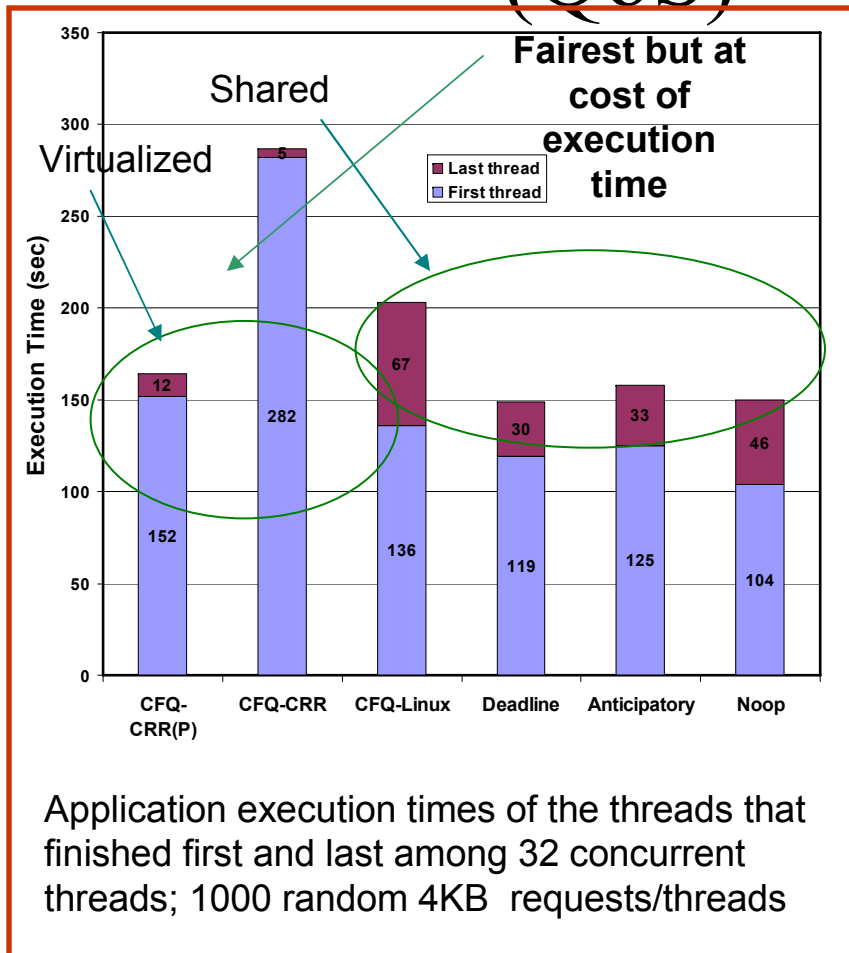


Dynamic Adaptivity in Support of Extreme Scale

Virtualized I/O

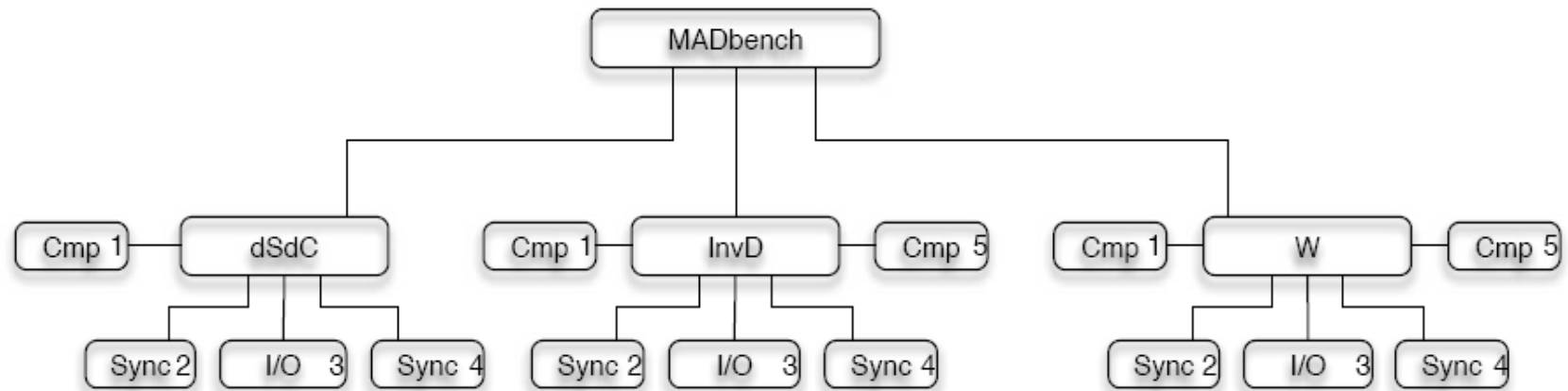
Execution Time vs. Fairness

(QoS)



DASES MADbench

Dynamic Adaptivity in Support of Extreme Scale

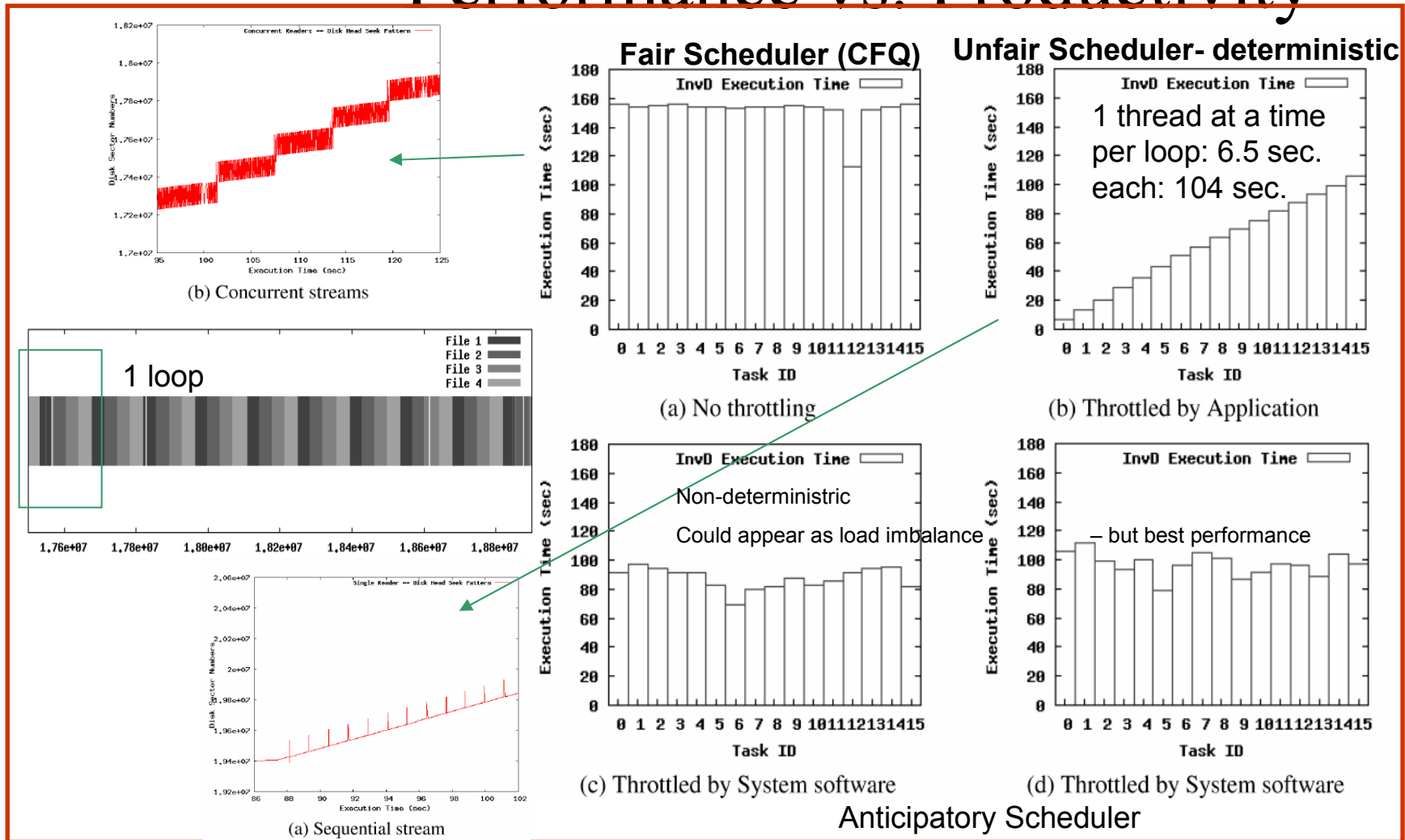


- dSdC – calculates signal correlation derivatives; writes to files
- invD – Calculates pixel-pixel data correlation matrix D; reads data
- W – Calculate dense matrix-matrix multiplications; reads data
- Each phases goes through several loops

MADbench-invD

Dynamic Adaptivity in Support of Extreme Scale

Performance vs. Productivity





Dynamic Adaptivity in Support of Extreme Scale

Performance vs. Productivity Sometimes in Conflict

- Algorithms/schedulers are built to improve the performance of the resource
- Designed in isolation
 - Disk schedulers
- On extreme scale systems
 - We need schedulers that provide performance isolation
 - Separating workloads
 - Performance analysis – productivity
 - But we need schedulers that provide performance
 - Unfair I/O scheduler on BG/L I/O node makes performance analysis of compute tasks that much difficult, e.g., load imbalance



Dynamic Adaptivity in Support of Extreme Scale

Software on Extreme Scale

- Predictable systems improve productivity
- Opportunistic / unpredictable systems may improve performance
- Adaptation is not just for improved performance but it must be for improved productivity as well!
- Systems must be virtualized
 - Fault tolerance
 - Performance analysis
 - Quality of service

DASES Future Related Work - 1

Dynamic Adaptivity in Support of Extreme Scale

- Using VIOS framework, provide both fairness and latency guarantees
- Apply fairness, performance isolation, and performance predictability to parallel I/O systems
- Apply the framework to storage systems that support parallel access
- For quality of service, weights can be defined. Consider how these weights can be used in large systems with thousands of processors and multiple applications.

DASES Future Related Work - 2

Dynamic Adaptivity in Support of Extreme Scale

- Further consider adaptation w.r.t. stateful vs. stateless resources
- Identify other ways that resource management adaptivity can be applied successfully
 - Virtual memory management
 - Small/large page allocation
 - Multi-core scheduling
- Consider how fairness and performance insulation / isolation can be guaranteed in other parts of the system
 - Lightweight operating systems - CPU performance

Acknowledgments

Dynamic Adaptivity in Support of Extreme Scale

- DOE, Office of Science (Grant # DE-FG02-04ER25622)
- IBM Corporation for IBM Shared University Research (SUR) Grants and IBM Faculty Awards
- UTEP and UT System for STAR (Science and Technology Acquisition and Retention) Program Award
- Seetharami Seelam, Ph.D.

DASES Refereed Publications - 1

Dynamic Adaptivity in Support of Extreme Scale

- Seetharami Seelam and Patricia Teller, “Fairness and Performance Isolation: an Analysis of Disk Scheduling Algorithms,” submitted to Workshop on High Performance I/O Techniques and Deployment of Very Large Scale I/O Systems (HiperIO’06), in conjunction with the IEEE International Conference on Cluster Computing, September 25-27, 2006.
- Seetharami Seelam, Jayaraman Suresh Babu, and Patricia Teller, “Performance Analysis of Disk Scheduling Algorithms for Asynchronous Requests,” to appear in Proceedings of the 3rd International Conference on Quantitative Evaluation of SysTems (Qest ’06), Riverside, CA, September 11-14, 2006.
- Patricia Teller and Seetharami, Seelam, “Insights into Providing Dynamic Adaptation of Operating System Policies,” ACM Operating Systems Review, 40:2: 83-89, April 2006.

DAISE Refereed Publications - 2

Dynamic Adaptivity in Support of Extreme Scale

- Seetharami Seelam, Jayaraman Suresh Babu, and Patricia Teller, “Automatic I/O Scheduler Selection for Latency and Bandwidth Optimization,” *Proceedings of the Workshop on Operating System Interference in High Performance Applications – OSIHPA*, Saint Louis, Missouri, 17 September 2005.
- Seetharami Seelam, Rodrigo Romero, Patricia Teller, and William Buros, “Enhancements to Linux I/O Scheduling,” *Proceedings of the 2005 Linux Symposium*, Ottawa, Canada, 20-23 July 2005.



Dynamic Adaptivity in Support of Extreme Scale

Theses, Dissertations, Technical Reports

- Ricardo Portillo, *Fine-grain Dynamic Adaptation of the Linux 2.6 Virtual Memory Manager: a First Step*, Master's Thesis, University of Texas-El Paso, Computer Science, July 2006.
- Seetharami Seelam, *Towards Dynamic I/O Scheduling in Commodity Operating Systems*, Ph.D. Dissertation, University of Texas-El Paso, Computer Engineering, May 2006.
- Jayaraman Suresh Babu, *Coarse-Grain Dynamic Adaptation for Asynchronous I/O Scheduling: Is it needed?*, Master's Thesis, University of Texas-El Paso, Computer Science, May 2006.
- Seetharami Seelam and Patricia Teller, "Disk Scheduling Using Fair Queuing and Round-Robin: Fairness Analysis," Technical Report, University of Texas-El Paso, Computer Science, December 2005.
- Rodrigo Romero, Seetharami Seelam, and Patricia Teller, "Workload Dependent Performance Analysis of Process Schedulers: A Case Study," Technical Report, University of Texas-El Paso, Computer Science, November 2005.



Dynamic Adaptivity in Support of Extreme Scale

Publications/Patents in Review or Preparation

- Seetharami Seelam and Patricia Teller, “CFQ-CRR: an I/O Scheduler that Guarantees Fairness, Performance Isolation, and Performance Predictability.”
- Seetharami Seelam, Andre Kerstens, and Patricia Teller, “ I/O Throttling to Improve the Performance of a Cosmology Application”.
- Seetharami Seelam, Sarala Arunagiri, and Patricia Teller, “VIOS2: a Scheduler of I/O Schedulers that Provides Explicit Latency Guarantees.”
- Seetharami Seelam and Patricia Teller, “Disk Scheduling with Performance Objectives.”
- Seetharami Seelam and Patricia Teller, “Disk Scheduling for Predictable Performance Behavior: Completely Fair Queuing and Compensating Round-Robin.”
- Seetharami Seelam and Patricia Teller, “CFQ-CRR (P): an I/O Scheduler for Meeting Performance Objectives.”

DAISES Questions?

Dynamic Adaptivity in Support of Extreme Scale



<http://research.utep.edu/daises>

pteller@utep.edu