Quality of Service Guarantee for Scalable Parallel Storage Systems

Professor Tzi-cker Chiueh

Experimental Computer Systems Laboratory

Computer Science Department

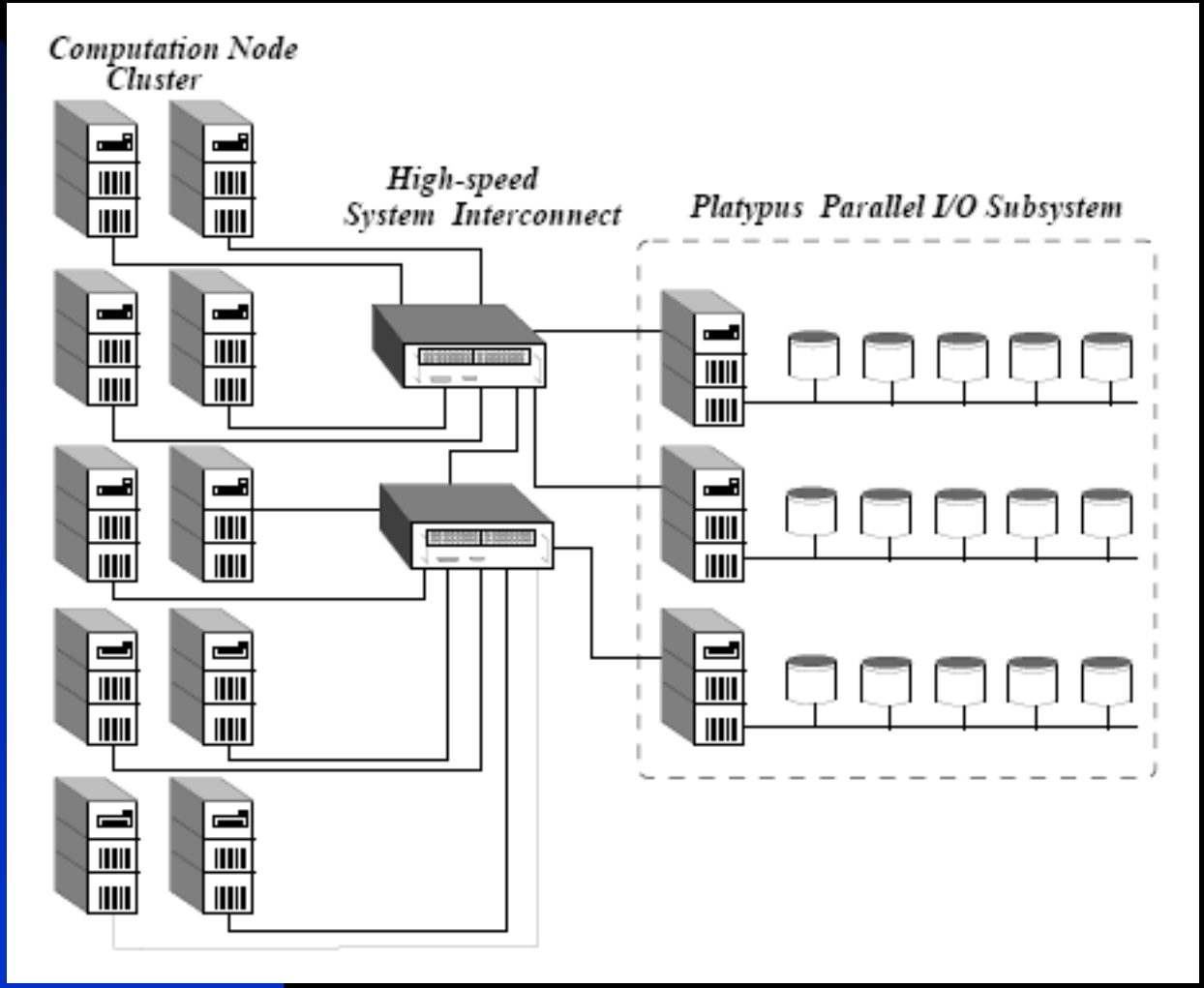
Stony Brook University

http://www.ecsl.cs.sunysb.edu/

Motivation

- PC-based compute clusters now become a norm
- Multiple parallel applications may run concurrently
- Key research question: how to provide performance isolation or guaranteed quality of service (QoS) among concurrent applications on a compute cluster's parallel storage system while maximizing its overall utilization efficiency
- Leverage multi-dimensional storage virtualization research: Stonehenge ➔ <A, B, C, D, E>
- Project Goal: design, implement and evaluate a scalable QoS-aware parallel I/O system called Platypus

System Architecture

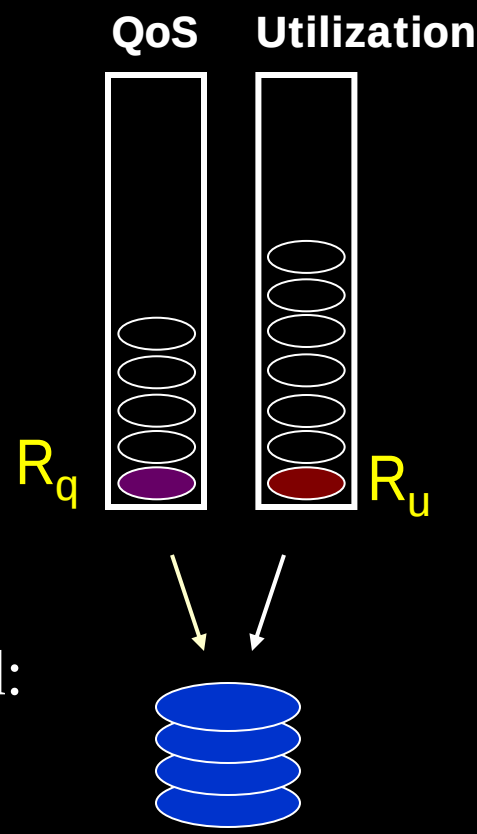


Disk QoS Guarantee

- Disk bandwidth requirement
  - Proportional to number of processors assigned to an application
  - Application-specified
- Proportional-share vs. priority-based
- How to account for an application's disk bandwidth consumptions across servers and disks
- Attribution of "storage virtualization" overhead
  - Overhead of switching disk head among virtual disks should be distributed according to their access locality

Disk Request Scheduling

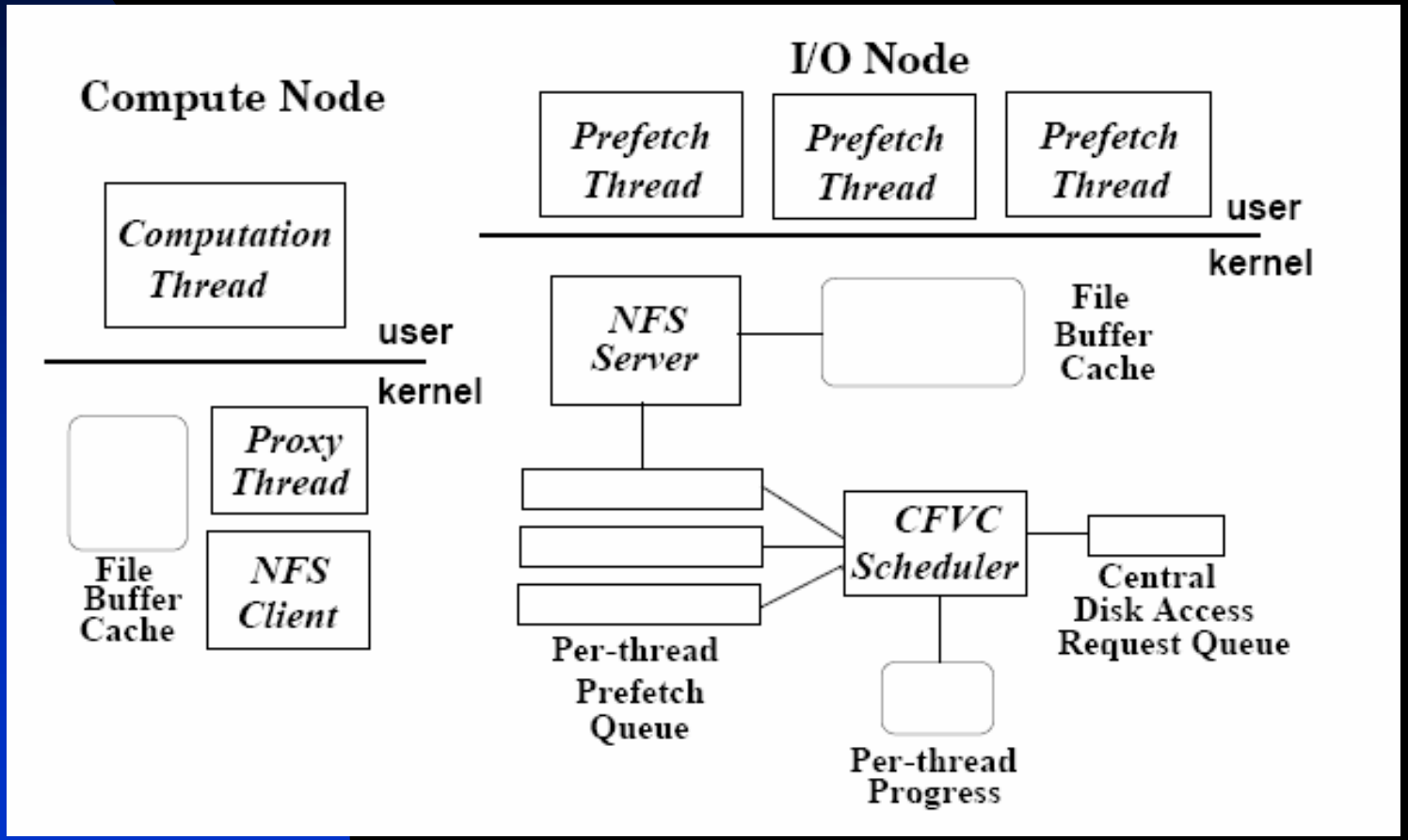
- Long-term fairness
  - Weighted fair queuing, virtual clock, earliest deadline-first
- Disk utilization efficiency
  - SCAN ➔ need real time
  - Shortest service time
- Dual-queue disk scheduling
- The degree of short-term unfairness of Virtual clock disk scheduling is unbounded: intelligent clock leap-forward



Decoupled File Prefetching

- Decoupled architecture
  - Separation of computation and access stream
  - Originally proposed to bridge CPU/memory performance gap
- Applied to parallel disk I/O to reduce access latency *and* to smooth out disk access burstiness
  - Automatic generation of an I/O thread from a parallel application, and synchronization communication between them
  - I/O threads run on Platypus nodes ➔ Active storage
  - Minimize performance overhead due to synchronization between computation and I/O threads

Software Architecture



Evaluation Tool

- Trace-driven approach to network file server evaluation➔ NFS trace play-back tool (TBBT)
  - Collecting traces once
  - Replay them again and again
- Adapt it to parallel file server evaluation
  - Inference of initial file system image (how to incorporate file system aging effect)
  - Clean up traces to ensure they are well-formed
  - Scale up and down the playback of traces along both temporal and spatial dimensions
  - Ensure playback timing accuracy among correlated requests

Year 1's Goal

- Completion of a multi-node, multi-disk disk request scheduler that
  - Distributes storage virtualization tax fairly
  - Achieves the optimal balance among short-term fairness, long-term fairness and disk utilization efficiency