# Future I/O Systems

## Tools, Applications, Architectures

Mustafa Uysal

University of Maryland

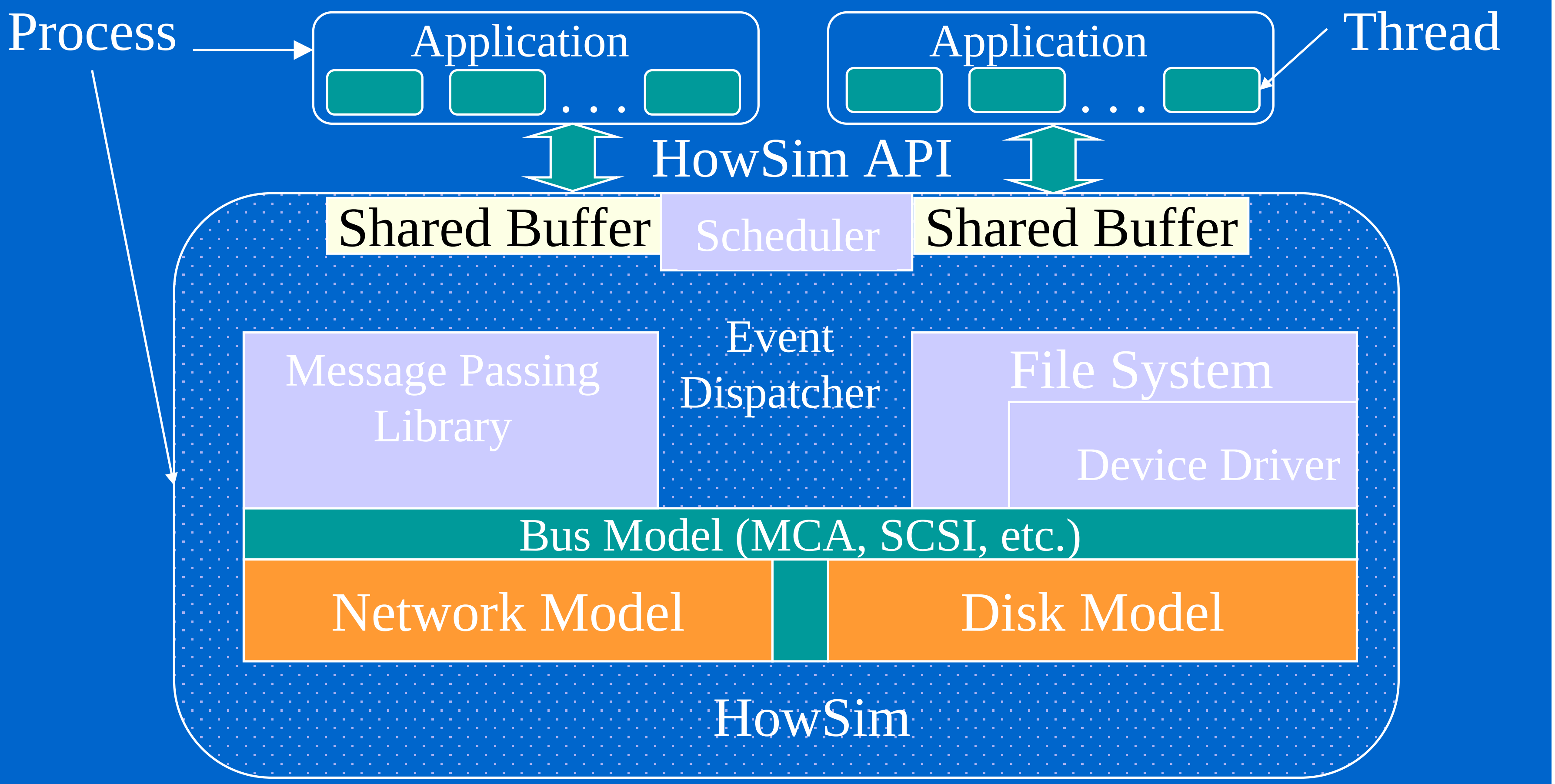
# Introduction

- High-fidelity architecture simulation
  - Detailed study of small to mid-range systems
  - Complex device models and system protocols
  - Validates less-detailed models
- I/O-intensive application suite
  - characterization and trends
- Case study: Active Disks

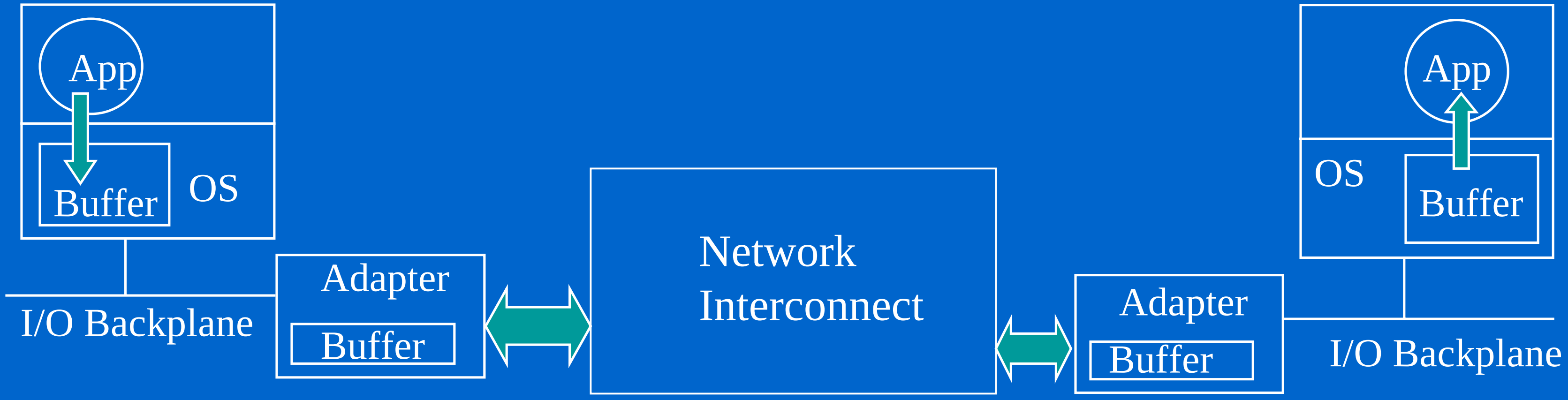
# HowSim

- Focus on I/O and network operations
- Accurate simulation of key hardware
  - disk, I/O bus, adapters, network, etc.
- Detailed simulation of key software
  - filesystem, scheduler, message passing libraries, interrupts, etc.
- Simulate multiple applications and processes

# HowSim Architecture



# Network Model



- Data transmission: DMA, latency-bandwidth model, flow-control
- Message Passing Library (blocking/non-blocking send/recv)
- Customizable for different networks (Multistage switch, ATM)

# Disk Model (1)

- Seeks
  - Long seeks: linear in distance
  - Short seeks: head-acceleration - sqrt(distance)
- Disk Geometry
  - Positioning: nominal rotational speed
  - Zone modeling
  - Sparing
  - Track and cylinder skew

# Disk Model (2)

- Data transfer : SCSI Protocol
- Disk Cache
  - Segmented cache
  - Read-ahead
  - Immediate writes
- Controller: Request queuing/scheduling
- Source Base: Ruemmler[94] and Kotz[94]

# Bus/Adapter/CPU Models

- Bus: latency+bandwidth, streaming
- Adapter: memory+protocol (e.g. SCSI)
- CPU: processing, interrupts, context-switch
  - coarse-grain: no cache/pipeline/FPU models
  - application-driven
  - user/system tasks are differentiated

# File system Model

- UNIX-like interface (i.e. read/write)
  - file access semantics
- Extent-based file allocation (user-specified)
- Fixed-size file cache buffer, write-behind
- C-SCAN request-scheduling

# Current Work

- Multiple device models
  - Integrating Ganger's Disk simulator
  - Support for SMP node
  - Gigabit-ethernet interconnect
- Multiple interfaces
  - Trace-driver
  - Single-application mode

# I/O-Intensive Applications

- Characterization Effort
  - I/O requirements
  - Access patterns, compute patterns
  - Inter-processor and intra-processor locality
- Modeling
  - Application traces vs. emulation

# MAMBO Suite

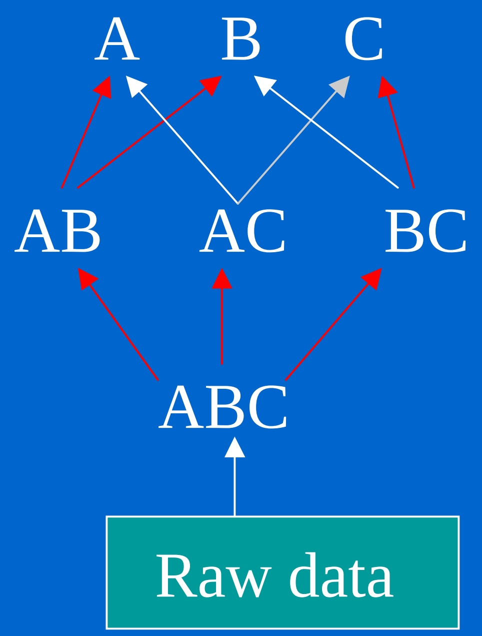
Maryland Applications for Measurement and Benchmarking for I/O

- DB2 Parallel Edition
- Data Mining
- Parallel Web Server
- Parallel Text Search
- Data Cube
- NOW-Sort

- Titan
- Virtual Microscope
- Out-of-core LU
- Rendering
- Hartree-Fock
- Electron Scattering
- Synthetic Aperture Radar
- others ...

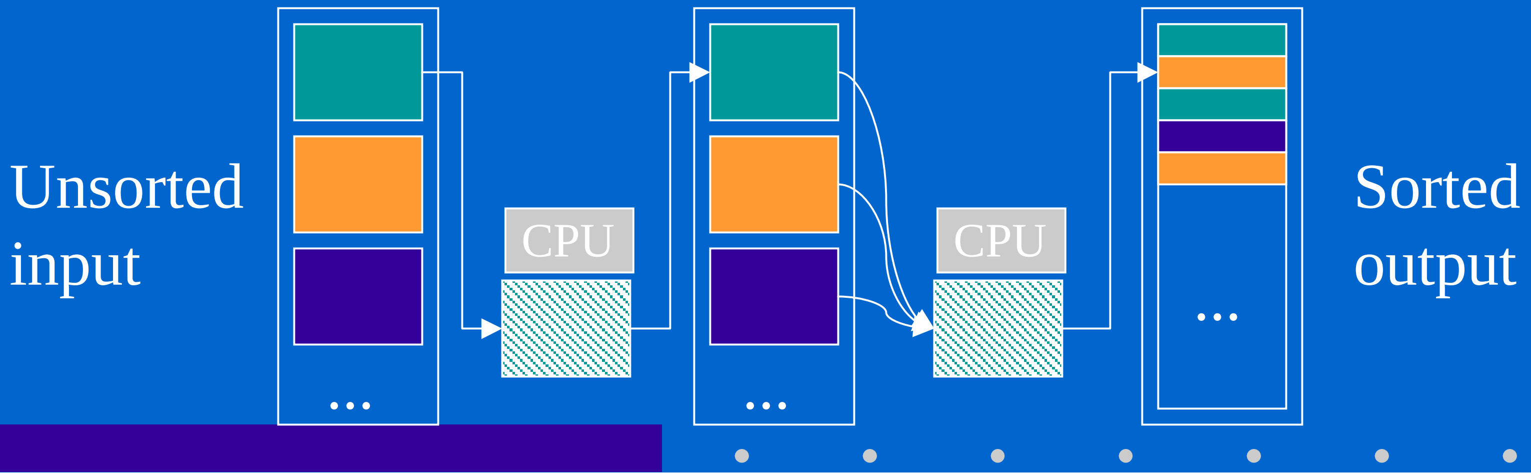
# Data Cube

- Multidimensional aggregates
- Domain: decision support
- Algorithm: PipeHash
  - Search lattice
  - MST computation
  - datacube: sequence of pipelines



# External Sort

- Two-pass algorithm
  - – Pass 1: Produce memory-sized sorted partitions
  - – Pass 2: Merge partitions
- Optimizations: large requests, overlapped I/O, multiple outstanding requests, etc.



# Application Trends

- Exponential rate of increase in online data
  - data storage doubles every 5 months
  - Petabyte satellite data repositories
- Shift in users' expectations
  - archival to frequent reprocessing of entire data
  - overnight data mining

# Technology Trends

- Disk performance/capacity improves
  - 15 MB/s and 18 GB/disk, around $1700
- Cheap, powerful cpu and large memory
  - $100 = 200 MHz Cyrix/6x86 and 16 MB (now)
  - $100 = 300 MHz + 32-64 MB (Y2K)

# Active Disks

- Disks with embedded CPU and memory
- Application-specific code executes on disk
- Processing partitioned: disk and host
  - Active disk performs bulk of the processing
  - Host coordinates/schedules/combines

- #CPUs increase with #Disks
- Processing power evolves with disks

# Programming Model

- User code (i.e. disklet) downloaded to disk
- Disklets use streams to access data
  - host- and disk-resident streams
  - pipe streams
  - data delivered in fixed-size buffers
- Disklet safety restrictions...

# OS support

- DiskOS - thin OS layer at disk
  - memory management
  - stream communication
  - disklet scheduling
- HostOS - disklet management
  - check/install/revoke disklets
  - manage host-resident streams

# Application Restructuring

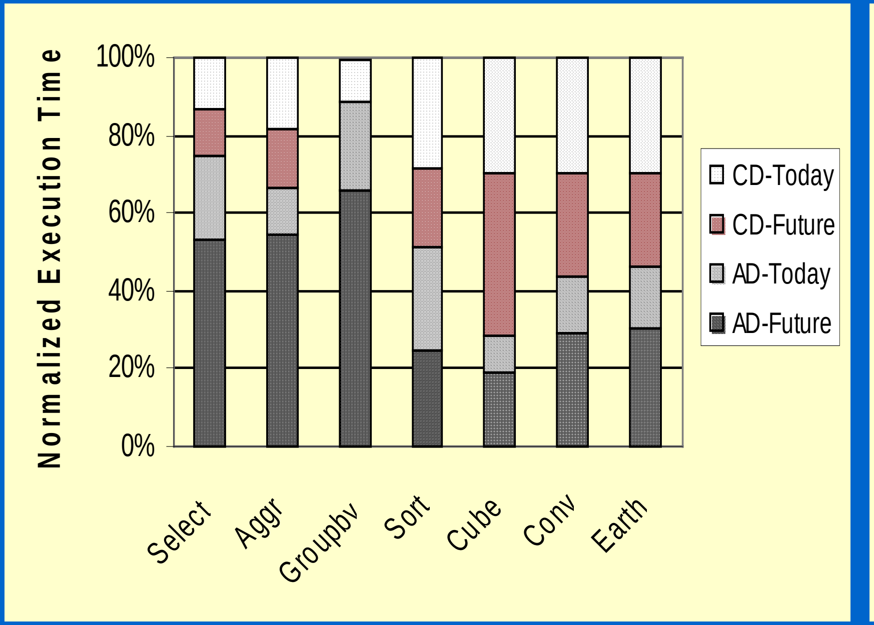
- Datacube: separate disklet per pipeline
  - disklets compute local group-bys (pipehash)
  - partial results shipped to host
  - host combines and computes global groupbys
- Sort, pass 1 - partition data
  - partitioner/sorter disklets, host moves data
- Sort, pass 2 - local merge
  - merger disklet

# Simulation Infrastructure

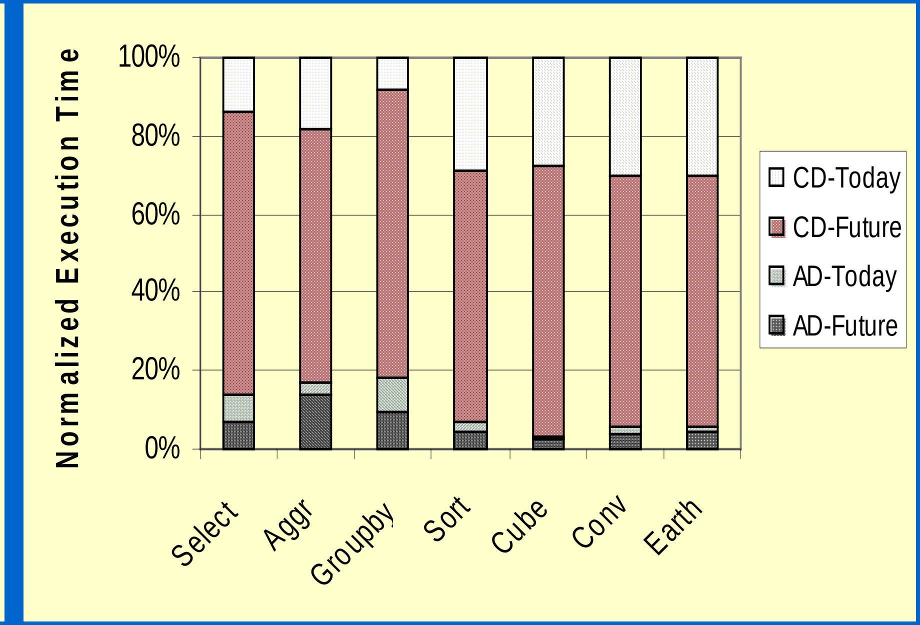
- Howsim node model changes:
  - – DiskOS controls disk operations
  - – Map stream communication to SCSI
  - – Extend file system functions for disklets
- Simulation parameters:
  - – **Today:** 350/200 MHz CPU, 1GB/16MB memory, 200 MB/s I/O bus, 15MB/s, 10000 RPM disk
  - – **Future:** 500/300 MHz CPU, 1GB/32MB memory, 300 MB/s I/O bus, 20 MB/s, 12500 RPM disk

# Active Disk Performance

4-disks



32-disks

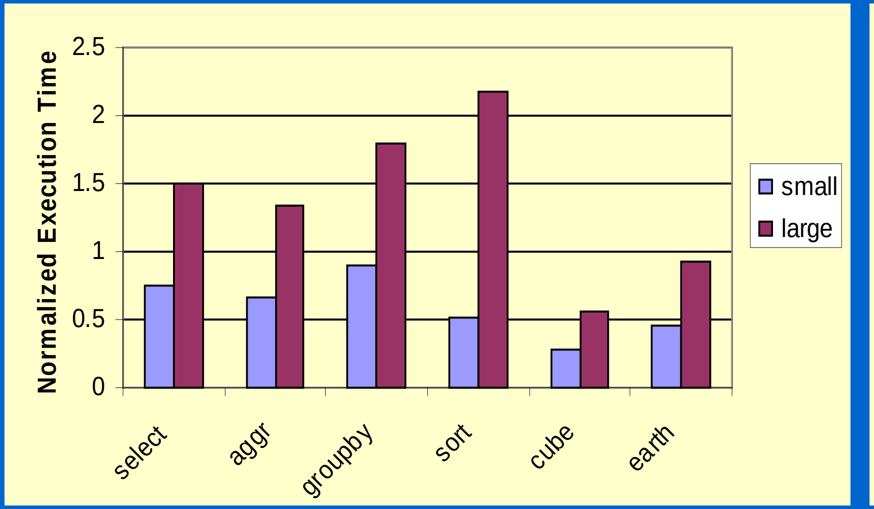


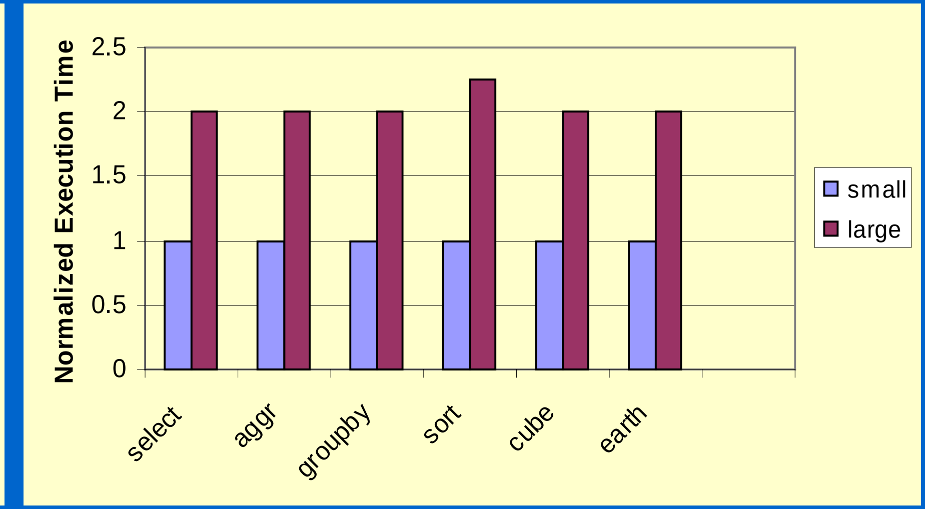
Conventional disks vs. active disks, varying technology

**Key performance issues: parallelism, interconnect bandwidth**

# Scaling datasets

4 disks, today's configuration





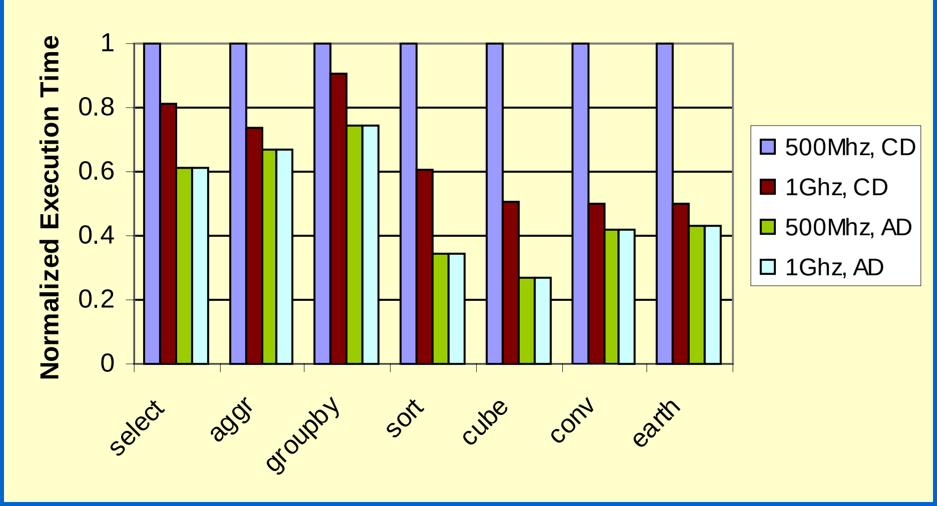
Active Disks

Conventional Disks

**Active disks perform better with larger datasets**

# Next Generation CPU

4-disks



Host CPU becomes more powerful

**Active disks remain superior to conventional disks**

# Conclusions and Future Work

- Flexible, accurate architecture simulation
- Emulators for database applications
  - Simple structure, easy to calibrate
- Active Disks
  - Initial results encouraging
  - multiprocessor study underway
  - new applications development underway