# A Study on the use of Tertiary Storage in Multimedia


- Leana Golubchik
  - Univ. Maryland, College Park
- Raj Kumar Rajendran
  - Columbia University

# Motivation

- Tertiary media is very inexpensive relative to secondary storage
- Digital Video is coming of age, and will require storage that is much larger than used today
- Is there a cost-effective way of using tertiary-storage in a storage hierarchy?

# Motivation

- Potential problems
  - high latency
  - multiplexing

# Constraints

- Heterogeneity requirements
- Real time constraints
- supports efficient multiplexing

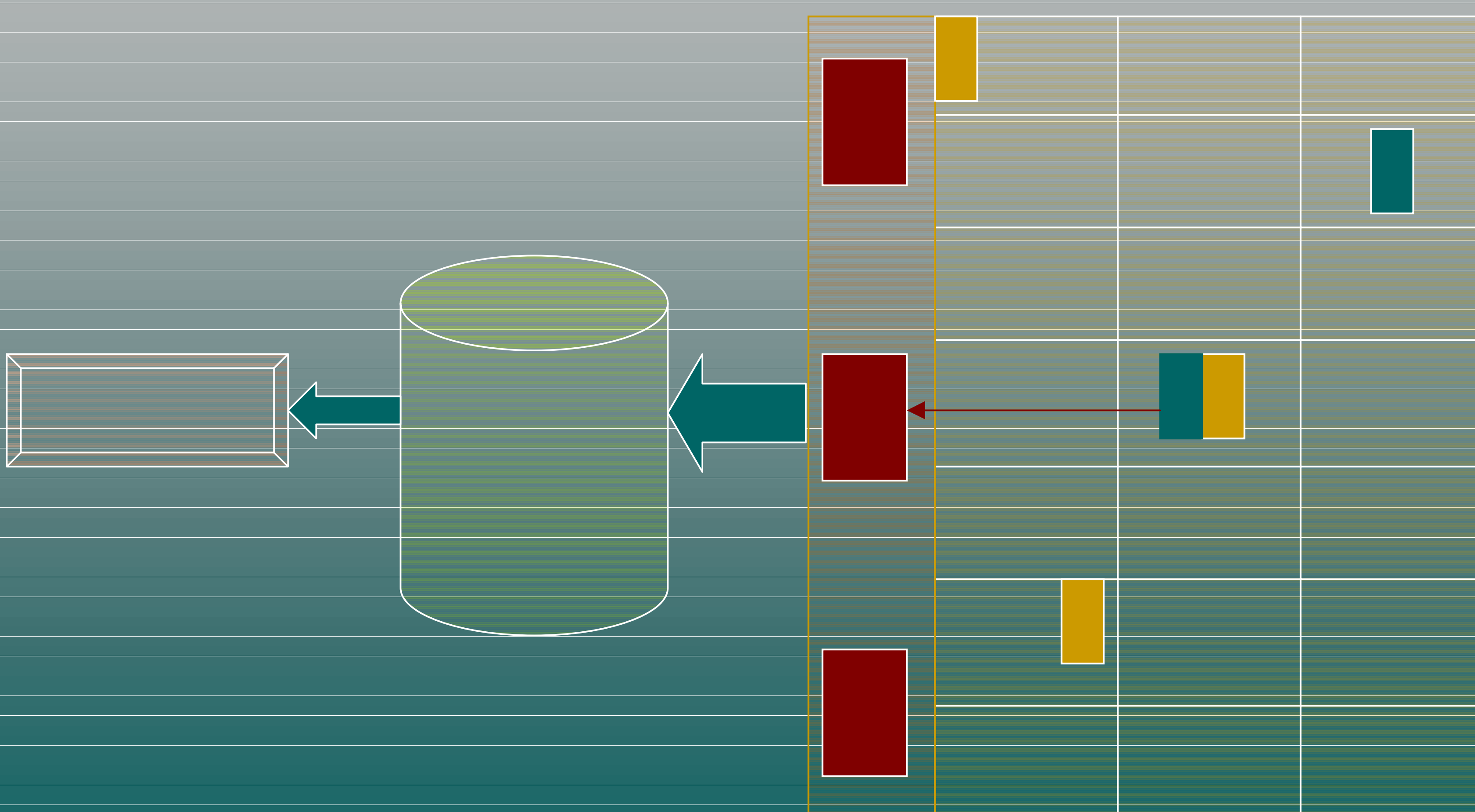
# Aims

- Given these constraints how do we design systems that
  - efficiently use system resources
  - provide low variance in latency at a given load

# Outline

- System Architecture
- Introduction to the problem
- Some solutions to the problem
  - Two Simple Solutions, and Rounds
- Analysis of the solutions
  - Simulation and Cost/Performance analysis
- Conclusions

# System Architecture



# System Architecture

- Three level hierarchy consisting of tertiary, disk and memory storage
- The tertiary system is a robotic library (many tapes, few drives)
- Objects are stored in randomly distributed fixed-size blocks called Pages
- The disk subsystem is used to buffer data from tertiary storage

# System Architecture

- Requests are for tuples (p,b) p Pages at a bandwidth b
- Delivering each page requires reserving robot and drive resources. Each Page requires a
  - robotic tape exchange (robot reservation)
  - the read
  - tape load, a seek, rewind & unload

# System Architecture

- Satisfying the request implies reserving periodic slots
  - a request for a 0.5 Mb/s object in a system with 3 MB/s drives requires reserving one slot in 6
- The system tries to fit a request into the reservation schedule as early as possible

# System Architecture

- slot
- stagger

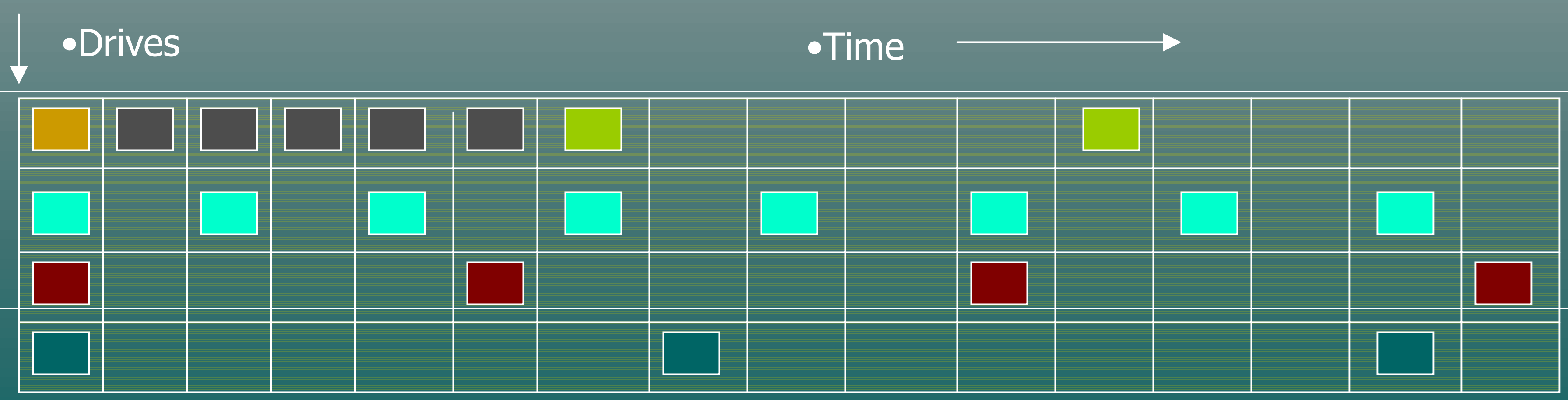
# Two Simple Solutions

- The Simple schemes
  - Allocate each request to a single drive
  - Choose the drive that can satisfy a request the earliest
- The Buffered scheme
  - Allocate each request to some combination of drives
  - Buffer enough data so that the stagger caused by robot contention is overcome

# Simple scheme

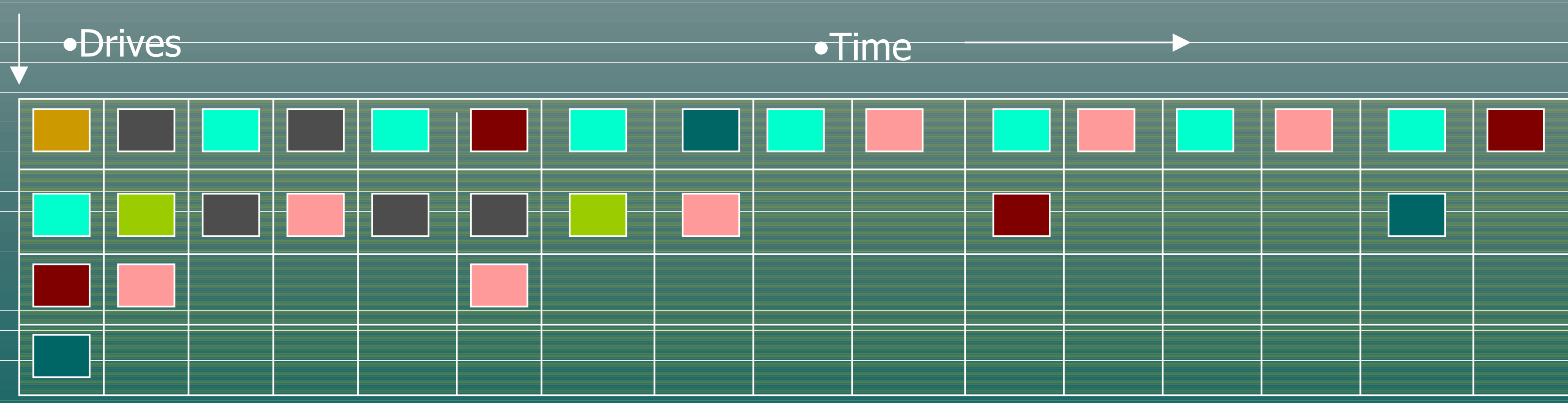
- 2,1/16
- 5,1/5
- 10,1/3
- 7,1/1
- 6,1/5
- 5,1/7
- 4,1/1

- Only fits in slot 24



# Buffered scheme

- 2,1/16
- 5,1/5
- 10,1/3
- 7,1/2
- 6,1/5
- 5,1/7
- 4,1/1



# The problem with these Solutions

- Poor utilization of tape-drive bandwidth
- Unpredictable quality-of-service

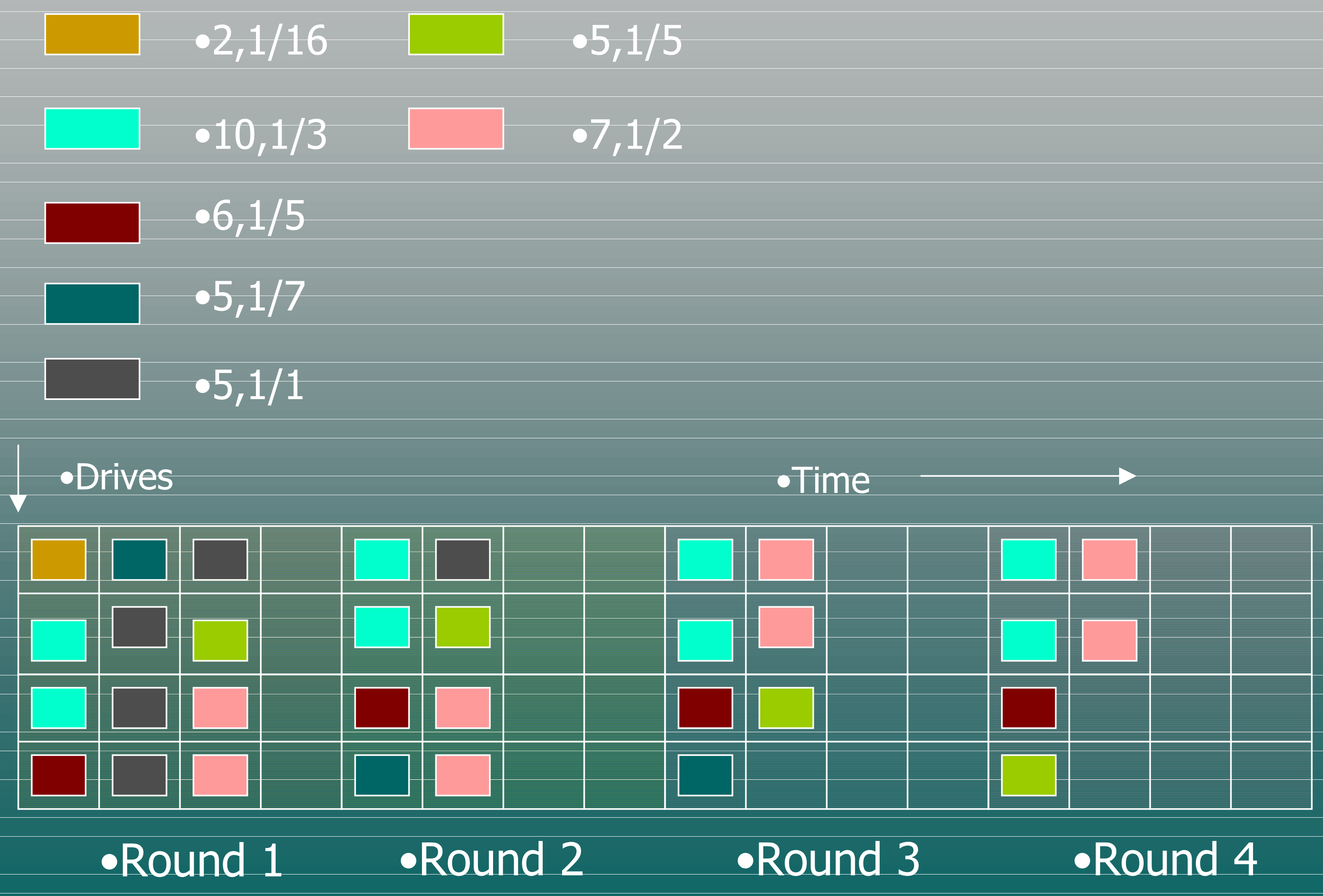
# Tradeoffs

- performance
- buffer-space

# Our Solution (Rounds)

- We aggregate a set of slots to form a Round
- Each slot in a round is homogeneous in time and space
- All Pages in a slot are staged before streaming begins
- When a request comes in the required number of slots in a round are allocated

# The Rounds scheme



# Performance Analysis

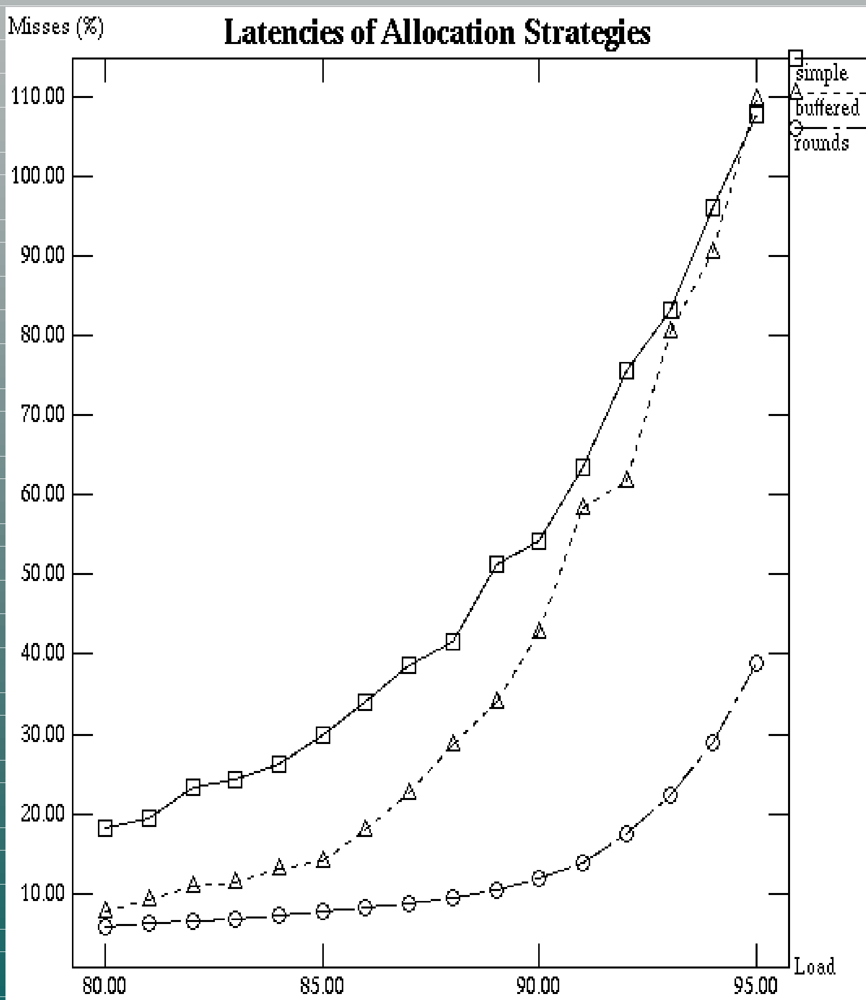
We studied the performance of the three solutions using simulation with the following metrics:

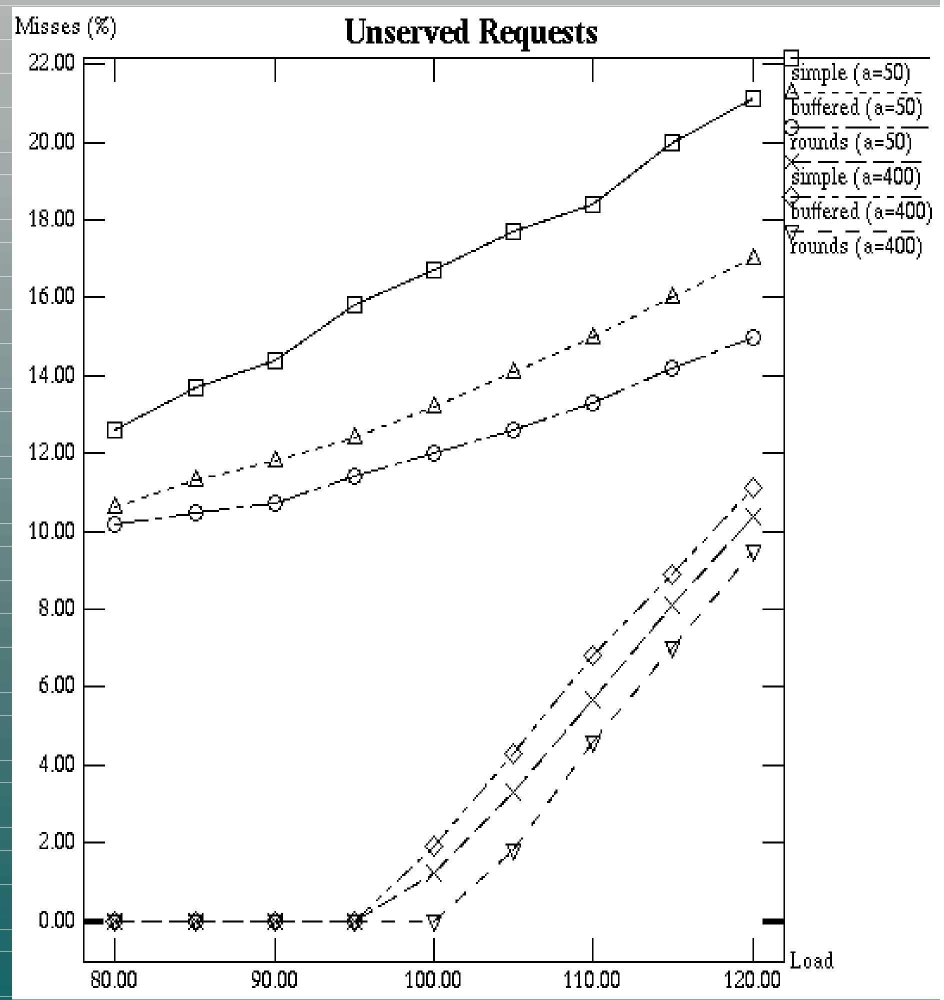
- service latency
- miss percentage
- cost/stream at a specific level of performance

# Latency & Miss %

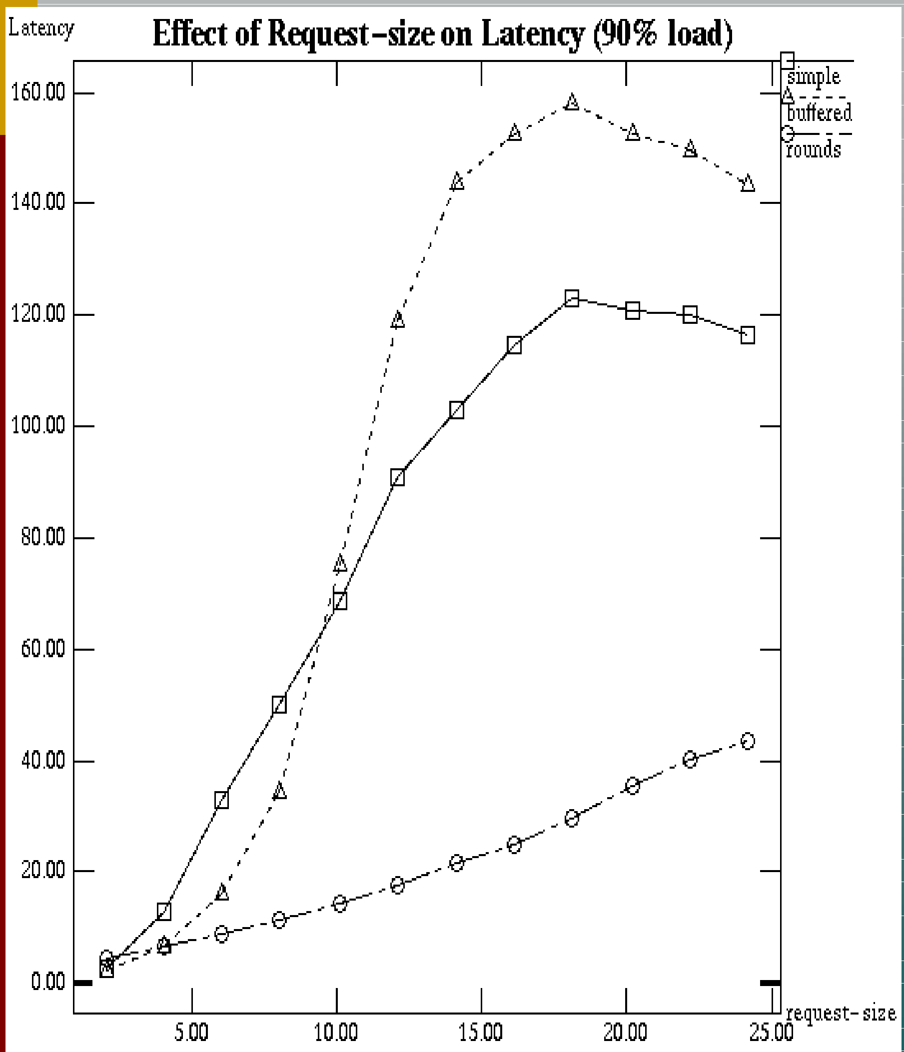
- At light workloads all three strategies exhibit similar performance
- At moderate and high workloads the "simple" and "buffered" strategies exhibit much higher latency and miss %
- Round's performance advantage increases as contention increases

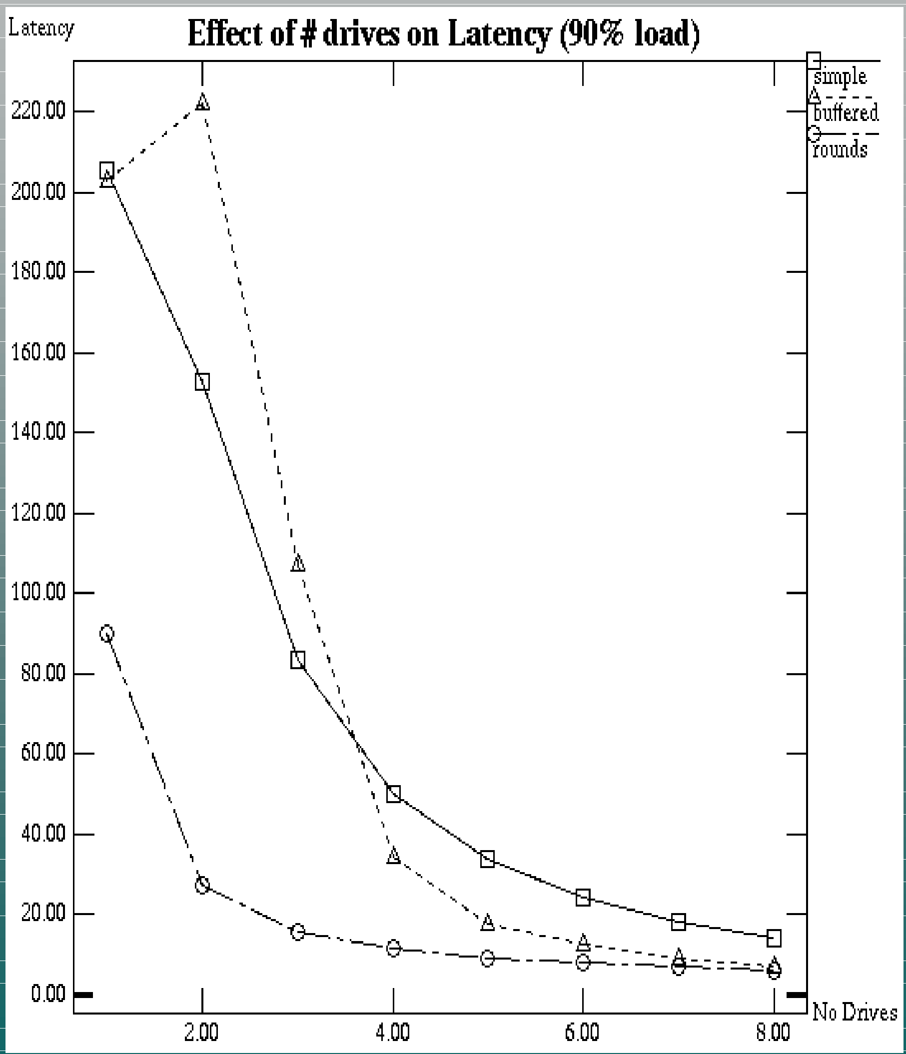
# Simulation Results



# Simulation Results



# Cost/Performance

We chose two operating points (levels of performance) and computed the cost using real systems.

- Operating Point 1: Rounds showed a consistent factor of two improvement
- Operating Point 2: The performance varied from a 0.2% degradation to a 40% improvement

# Conclusions

- Performance of the three schemes is similar at low workloads
- Rounds performs significantly better as contention increases
- Depending on the architectural configuration the cost/stream varies from a 3% degradation to a *factor or two* improvement

# Summary

- High Latency and drive/robot contention are problems but innovative design can solve these problems
- We propose an architecture that
  - supports heterogeneous multimedia objects
  - efficiently utilizes resources

# Future considerations

– Support for Objects with variable bandwidth requirements

– Service latency reduction by partial storage on disks