Resource Management Issues for “Streaming” Multimedia Applications

Jason Nieh

Columbia University

Requirements of Multimedia

- Quality of service
  - Multiple activities/applications
  - Coexistence of real-time and conventional
- Two scheduling approaches
  - SMART
  - GR3

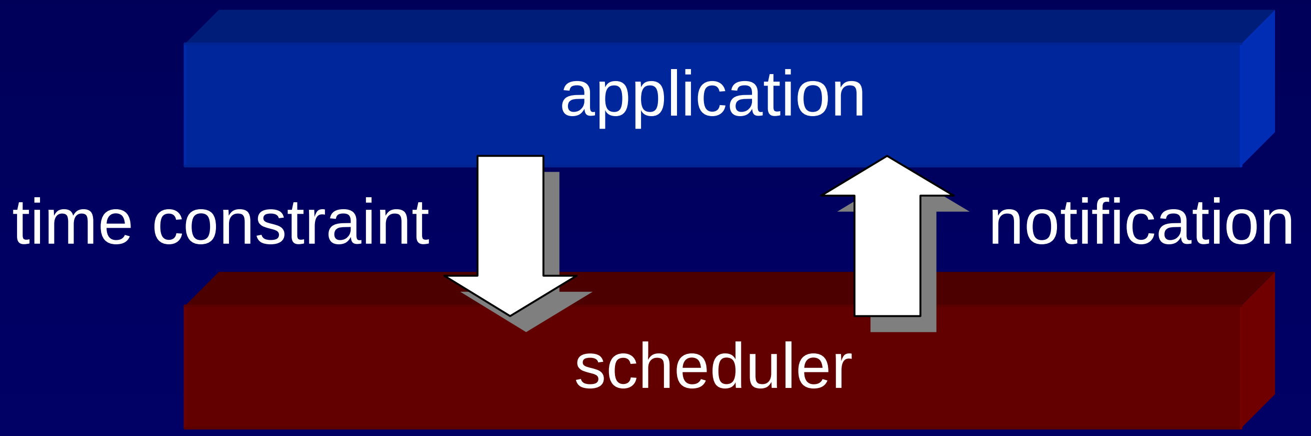
1. The SMART Approach

Scheduler for **M**ultimedia **A**nd **R**eal-**T**ime

- Simple application interface and scheduling algorithm
- Supports requirements of multimedia
  - Explicit management of time constraints
  - Dynamic feedback to allow real-time applications to adapt
  - Resource sharing across real-time and conventional applications

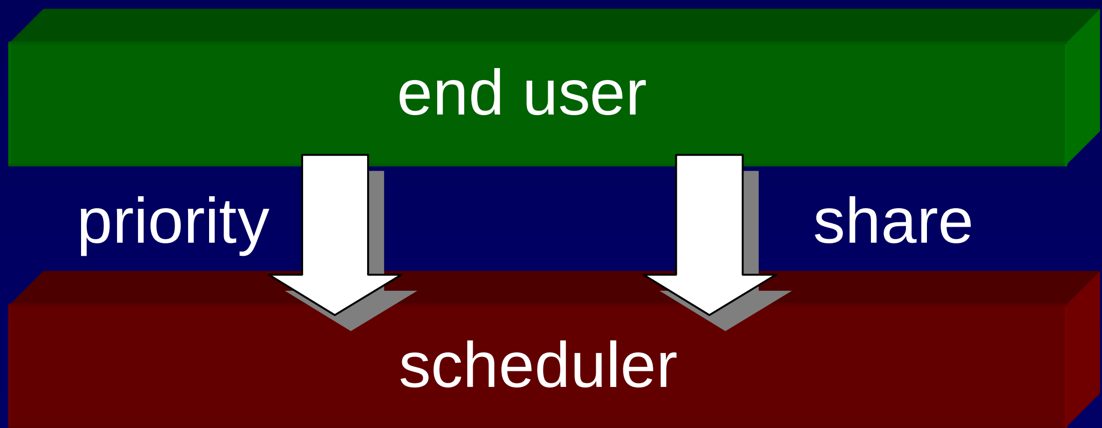
SMART Interface: Application Support

- Reduce the burden of application developer
  - **time constraint**: inform scheduler of deadline, CPU time estimate
  - **notification**: upcall to inform application of missed deadline
- End user need not be aware of application real-time requirements



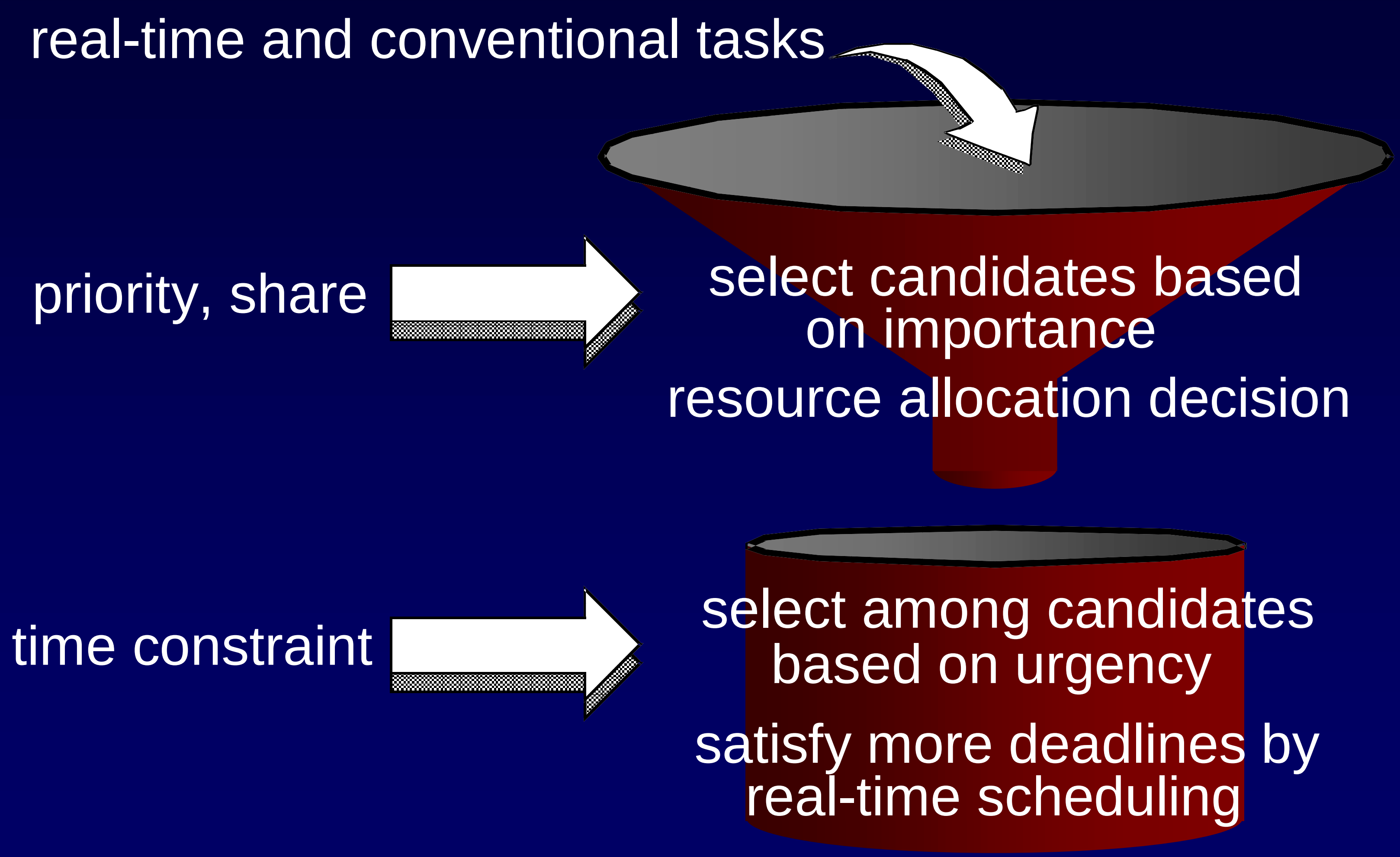
SMART Interface: User Control

- Provide end user with flexible predictable control
  - **priority**: monopolize resource
  - **share**: proportional sharing
- Expected usage for most users
  - All applications at default priority with equal share
  - No user parameters required



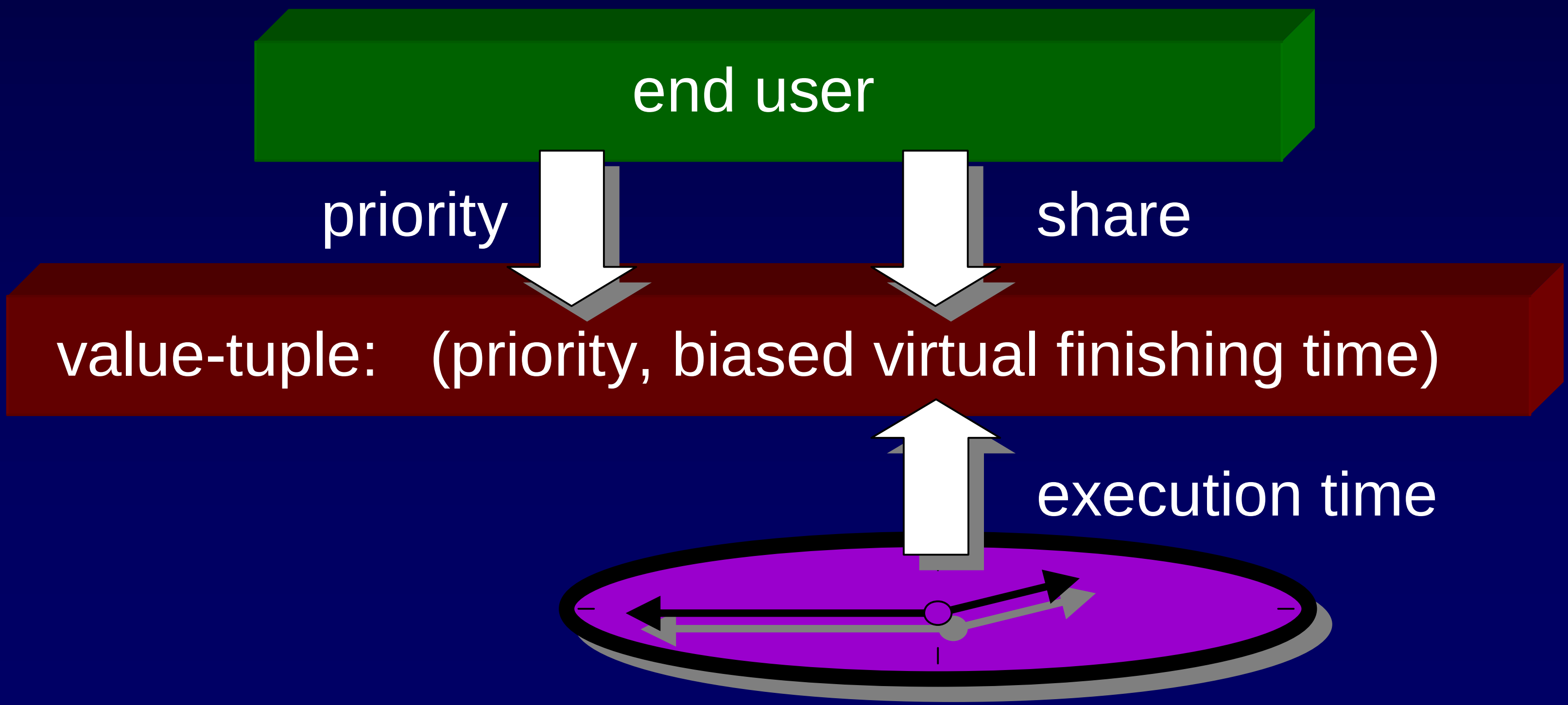
SMART Algorithm: Overview

- Separate importance and urgency



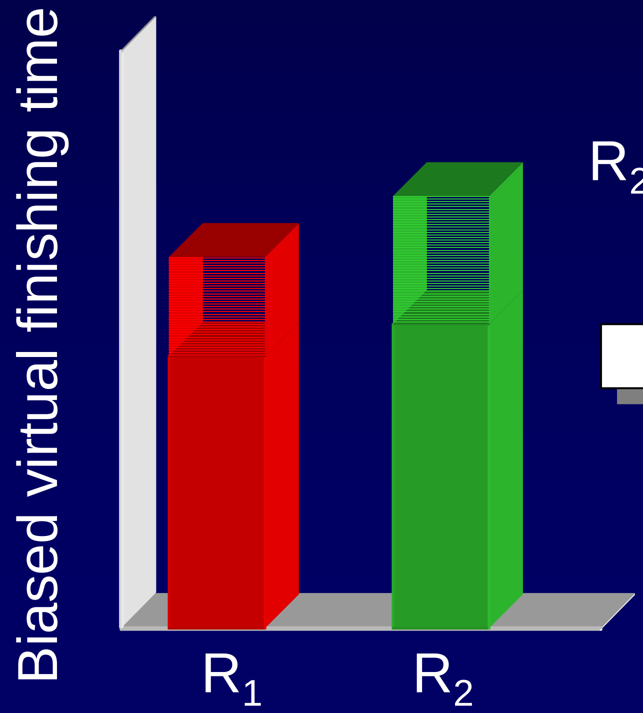
SMART Algorithm: Select Candidates

- Value-tuple
  - Priority: from user controls
  - Biased virtual finishing time: measure of weighted resource consumption



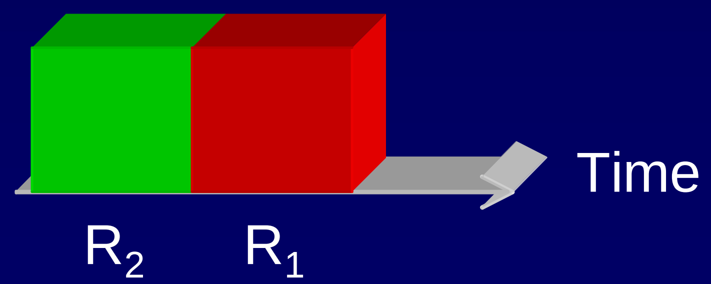
SMART Algorithm: Real-time Scheduling

- Execute candidate with earliest deadline without causing tasks with larger value-tuples to miss deadlines
  - Shed earlier deadline tasks if needed in overload to ensure larger value-tuple tasks can meet their deadlines
  - Reduces to earliest deadline scheduling in underload



R_2 deadline < R_1 deadline

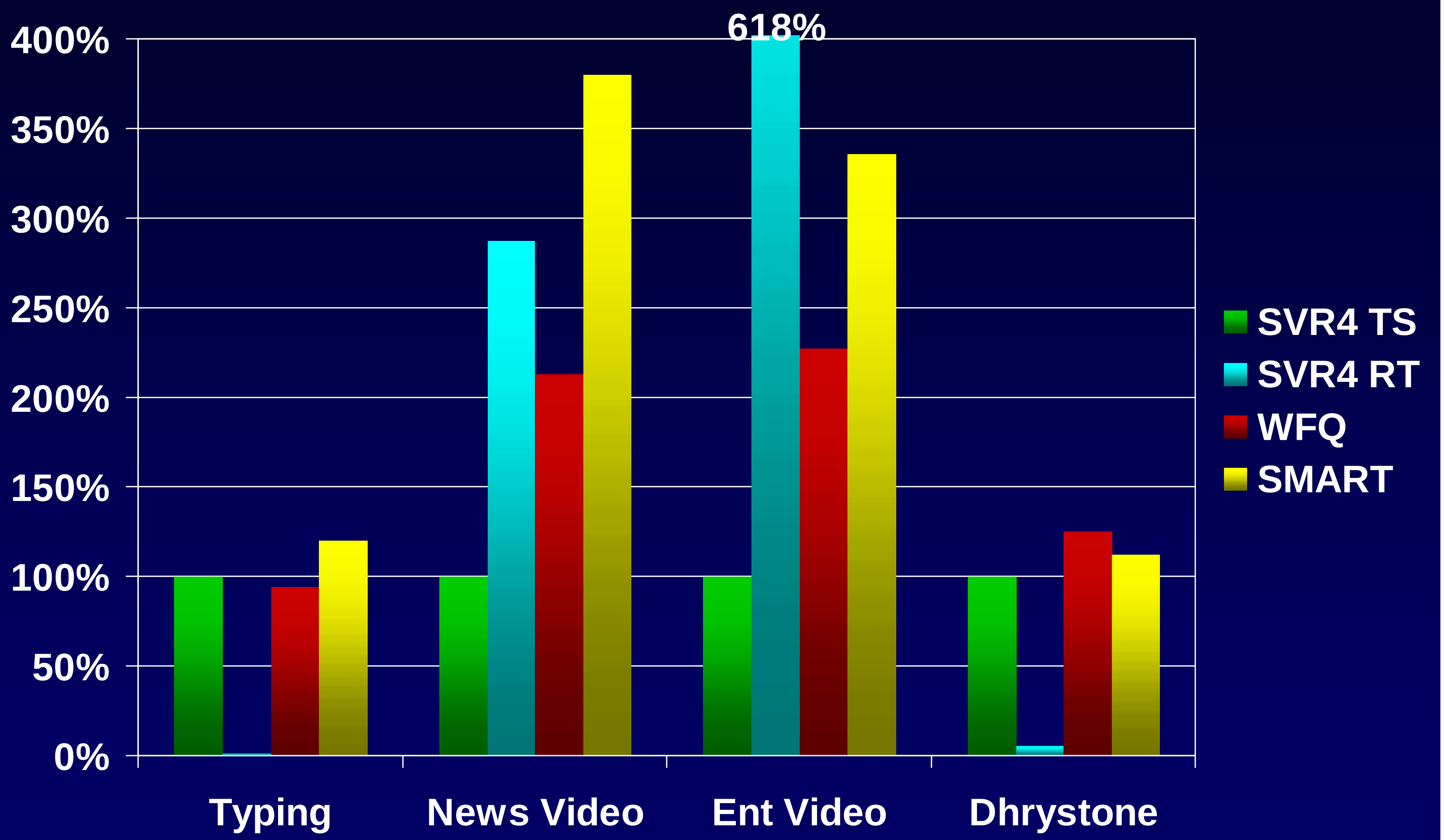
deadline-order schedule



SMART Summary

- High priority applications not degraded due to low priority applications
- Proportional sharing among real-time and conventional applications in the same priority class
- Graceful transitions during fluctuations in loads
- Satisfy real-time constraints in underload, optimally
- Trade off instantaneous fairness for better real-time and interactive response time
- Notification of resource availability

SMART Scheduling Performance



2. The GR3 Approach

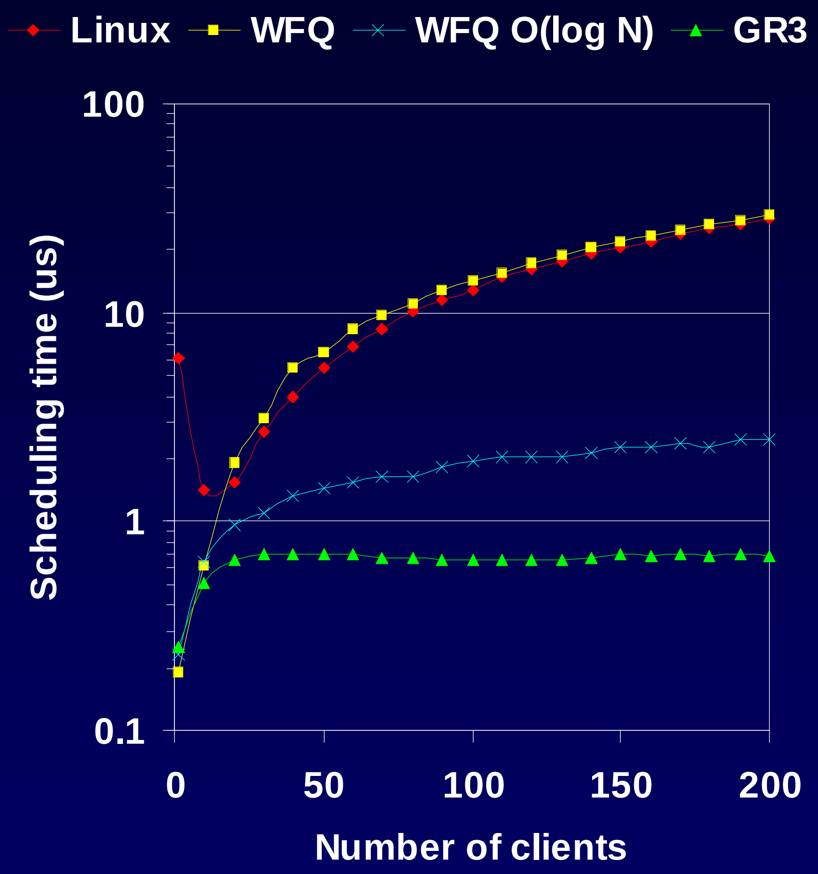
Group **R**atio **R**ound **R**obin

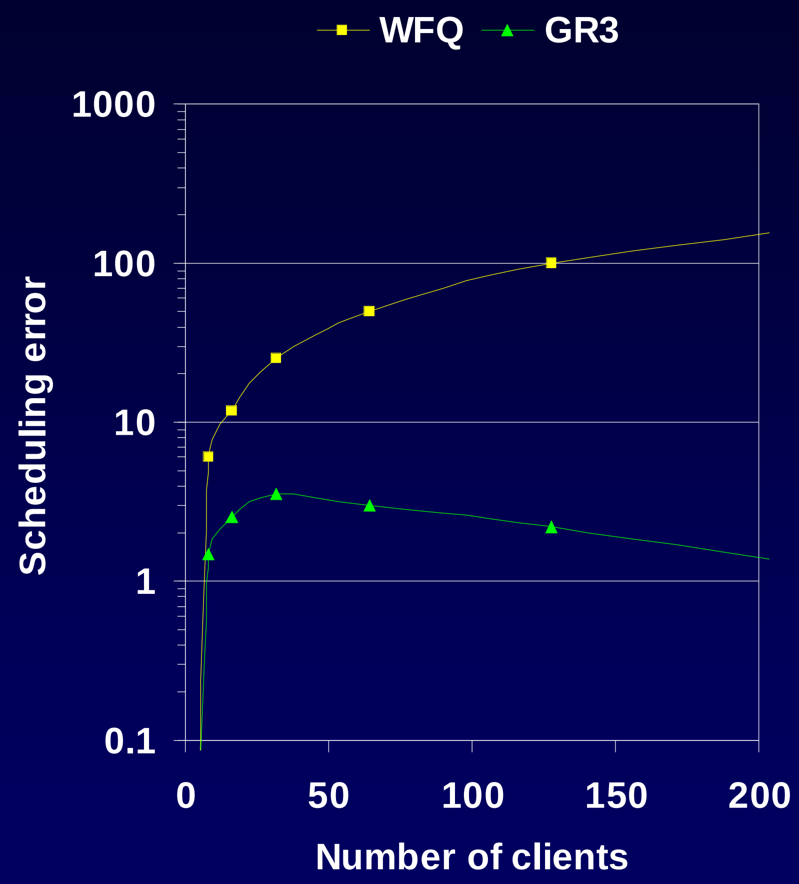
- Simple proportional-share model
- Fine-grain performance isolation
- Accurate and scalable
- O(1) scheduling overhead

GR3 Scheduling Algorithm

- Binary grouping
  - Group tasks by share S: $2^k < S < 2^{k+1}$
- Intergroup scheduling
  - Order groups by sum of shares in group
  - Select group based on ratio of group shares
  - Limited number of groups
- Intragroup scheduling
  - Modified round-robin within group
  - Limited share range within group

GR3 Scheduling Performance



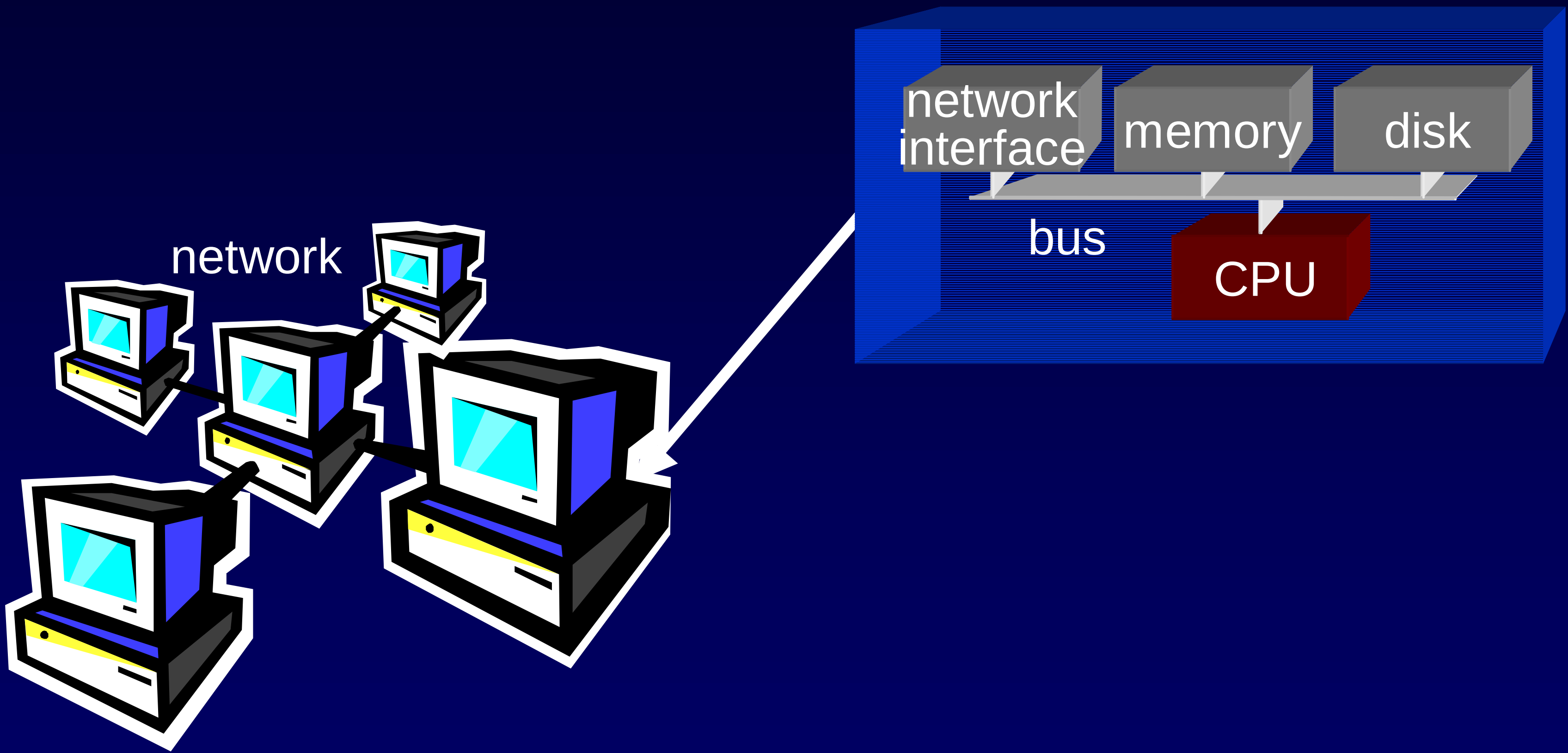


What About Multiprocessors?

- Still need
  - Integrate real-time and conventional
  - Flexible control and efficient use of hardware
- But unanswered questions as well
  - Which processor to use for a given task?
  - How to balance load across processors?
  - No optimal real-time scheduling?
  - Resource sharing abstraction?
- Greater potential for mismanagement

Multi-resource Management Problem

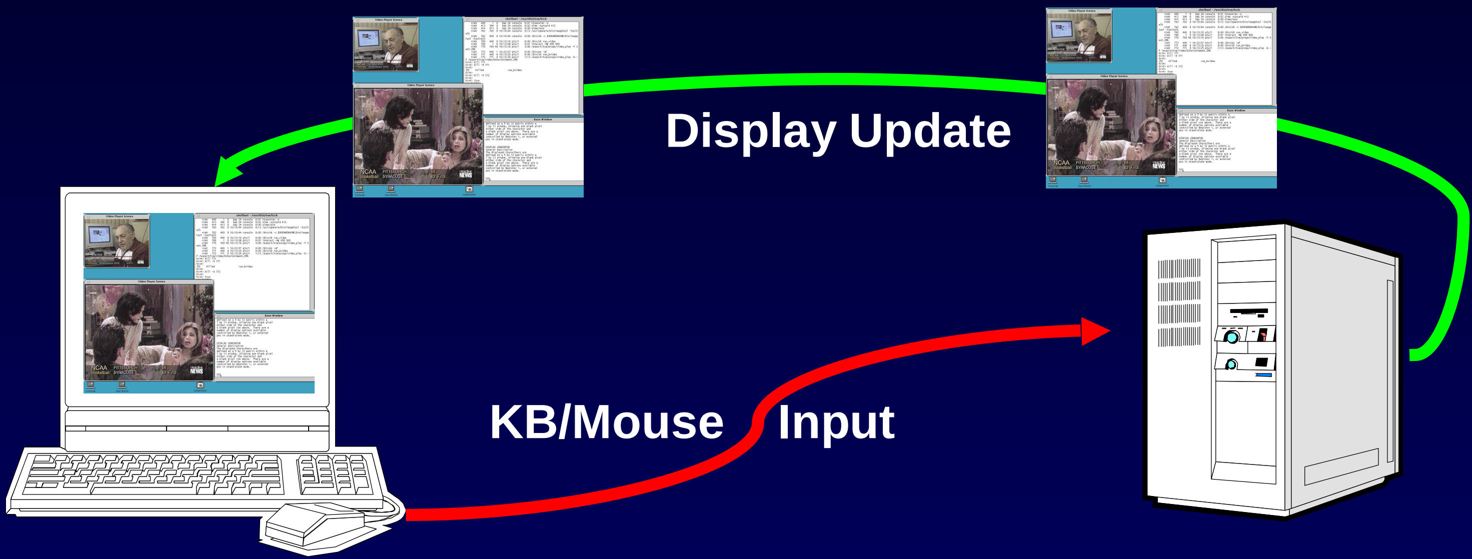
- CPUs just one component in the system



- Need distributed resource management for superior end-to-end performance of multimedia applications

Other Streaming Research Areas

- Multimedia desktop streaming



- Transparent migration of streaming multimedia applications
- Flow computing streaming network system architecture

More Information

Networking Computing Laboratory

http://www.ncl.cs.columbia.edu