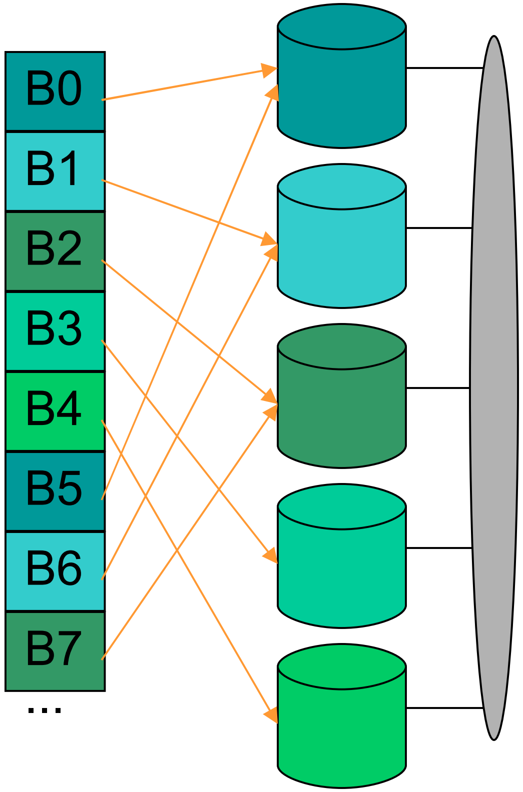
# *A Case for Heterogeneous Disk Arrays*

Toni Cortes and Jesús Labarta
Departament d'Arquitectura de Computadors
Univeritat Politècnica de Catalunya - Barcelona

# Disk Arrays (RAIDs)



- **Group several disks**
  - Single address space
  - High capacity
  - Improved performance
  - *Low* cost
- **Heterogeneous RAID**
  - Not all disks are equal

# Motivation

- n **Heterogeneous disk arrays are becoming a common configuration**
  - l Replacing a new disk
  - l Adding new disks
- n **Current solution**
  - l All disks are treated as equal
    - ü No performance gain is obtained
    - ü No capacity gain is obtained

# ***Objective***

- **AdaptRaid0**
  - Block-distribution policy
  - Take advantage of the *goodies* of each disk
- **Target Environment**
  - Scientific and general purpose
  - Not multimedia
    - Solutions have already been presented
    - Very dependent on some characteristics
  - Disk arrays level 0 (*RAID0*)
    - Level 5 is under development

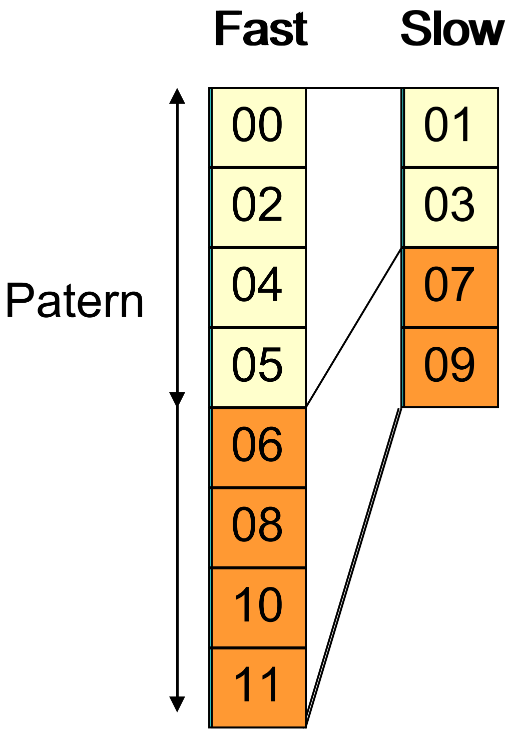
# Related Work

- **Multimedia Systems**
  - Random distribution with replication ***(Santos98)***
  - Policy based on logical disks ***(Zimmerman98)***
  - Use fast disk for hot data ***(Dan95)***
  - Differences:
    - Large blocks, only reads, and sustained bandwidth
- **General purpose**
  - HP AutoRaid ***(Wilkes95)***
  - Disc-Cache Disk ***(Hu98)***
  - Differences:
    - Do not adapt to the existent hardware

# *Disk Arrays and Parallelism*

- **Parallelism within a request**
  - Requests have to large
    - The sub-request of each disk has to be large
    - Seek + search + transfer in all disks
- **Parallelism between requests**
  - The number of disks has to be large
    - Compared to the average number of disks used in a requests

# AdaptRaid0: An Example



- **Basic idea**
  - Load each disk depending on its characteristics
- **Example**
  - 1 fast disk
    - Size = S
    - Performance = P
  - 1 slow disk
    - Size = S/2
    - Performance = P/2

# AdaptRaid0: The Parameters

- **Utilization factor (UF)**
  - One factor per disk
    - Larger disks have more blocks?
    - Faster disks have more blocks?
- **Lines in pattern (LIP)**
  - We define a pattern using the UF
    - Large patterns allow more requests with good disks
    - Small patterns allow a better distribution

# AdaptRaid0: The Algorithm

- **Algorithm**
  - Decide LIP and Ufd
  - Compute number of blocks per disk in the pattern
    - `Blocksd= int(UFd * LIP)`
  - Distribute blocks in a round-robin way
    - Use the available disks
    - A disk becomes unavailable when Blocksd have already been placed in it
  - Repeat step 3 until one disk becomes full

# Methodology

n **Parameters**

- l UF based on the size of the disk
- l Lines in pattern
  - ü 100 lines for 8-disk arrays
  - ü 10 lines for 32-disk arrays

n **Simulation**

- l Simulator: HRaid ***(Cortes99)***
- l Workload from HP labs ***(1999)***

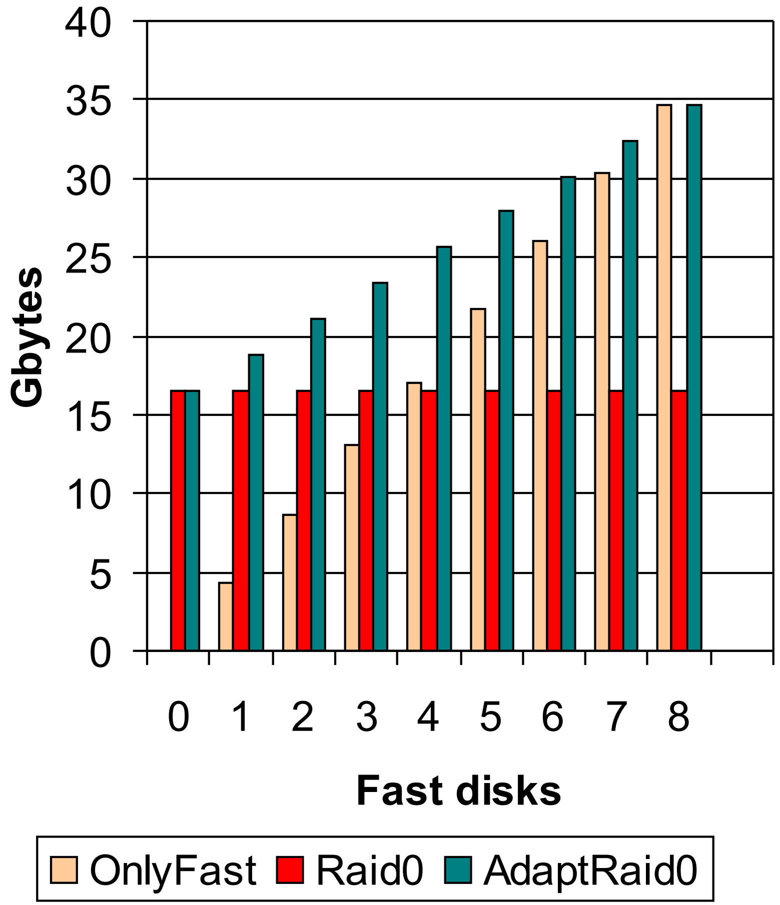
n **Reference systems**

- l Raid0 and OnlyFast

# Environment

- **Disks**
  - Fast disk
    - Seagate Barracuda 4LP (4.339 Gbytes)
  - Slow disk
    - Seagate Cheetah 4LP (2.061 Gbytes)
- **Bus**
    - 10us latency
    - 100Mbit/s bandwidth
- **File system**
    - 10 requests in parallel

# Capacity Evaluation



n **Raid0**

l Constant capacity

ü Small

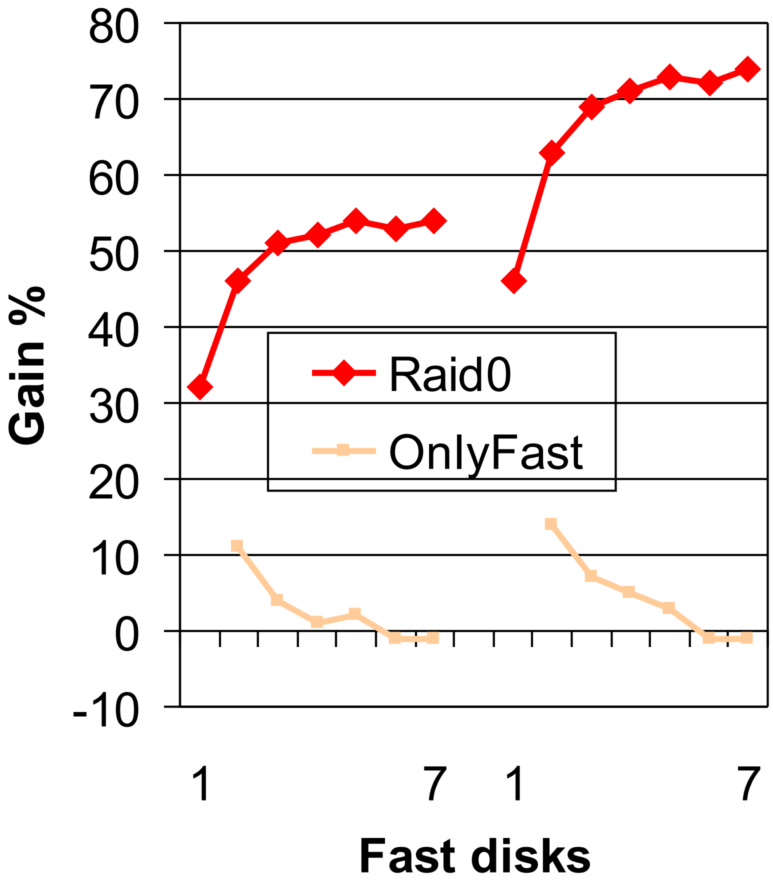
n **OnlyFast**

l Small capacity with few disks

n **AadaptRaid0**

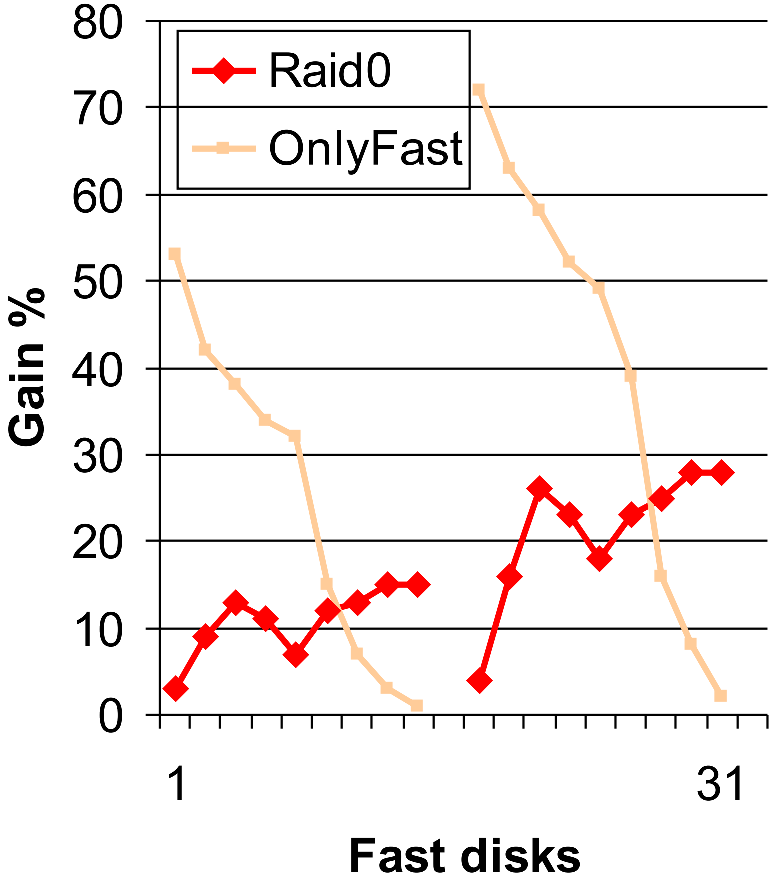
l Offers the best size

# Performance Evaluation (8 disks)



- **Raid0**
  - Does not use
    - Characteristics of good disks
- **OnlyFast**
  - Does not use
    - Parallelism between requests

# Performance Evaluation (32 disks)



- **Raid0**
  - Does not use
    - Characteristics of good disks
  - It uses
    - Parallelism between requests
- **OnlyFast**
  - Does not use
    - Parallelism between requests

# Conclusions

- n **AdaptRaid0**
  - l Performance
    - ü It knows how to use the disks
    - ü Allows parallelism
  - l Size
    - ü It uses all the available capacity

# Future Work

- **Solve the same problem for Raid5**
  - Problem of parity blocks
  - Less scalable
    - No parallelism among requests