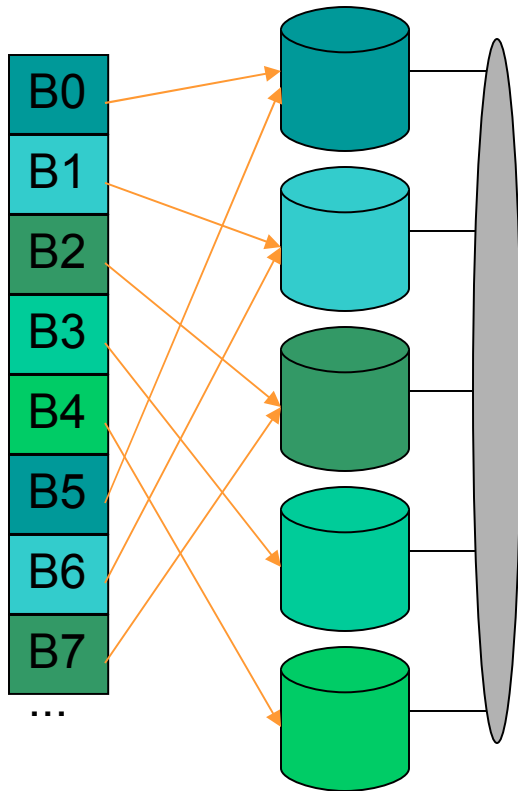


# *A Case for Heterogeneous Disk Arrays*

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

# Disk Arrays (RAIDs)



## n Group several disks

- | Single address space
- | High capacity
- | Improved performance
- | *Low cost*

## n Heterogeneous RAID

- | Not all disks are equal

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

## n **AdaptRaid0**

- | Block-distribution policy
- | Take advantage of the *goodies* of each disk

## n **Target Environment**

- | Scientific and general purpose
- | Not multimedia
  - ü Solutions have already been presented
  - ü Very dependent on some characteristics
- | Disk arrays level 0 (*RAID0*)
  - ü Level 5 is under development

## **n Multimedia Systems**

- | Random distribution with replication (*Santos98*)
- | Policy based on logical disks (*Zimmerman98*)
- | Use fast disk for hot data (*Dan95*)
- | **Differences:**
  - ü Large blocks, only reads, and sustained bandwidth

## **n General purpose**

- | HP AutoRaid (*Wilkes95*)
- | Disc-Cache Disk (*Hu98*)
- | **Differences:**
  - ü Do not adapt to the existent hardware

# ***Disk Arrays and Parallelism***

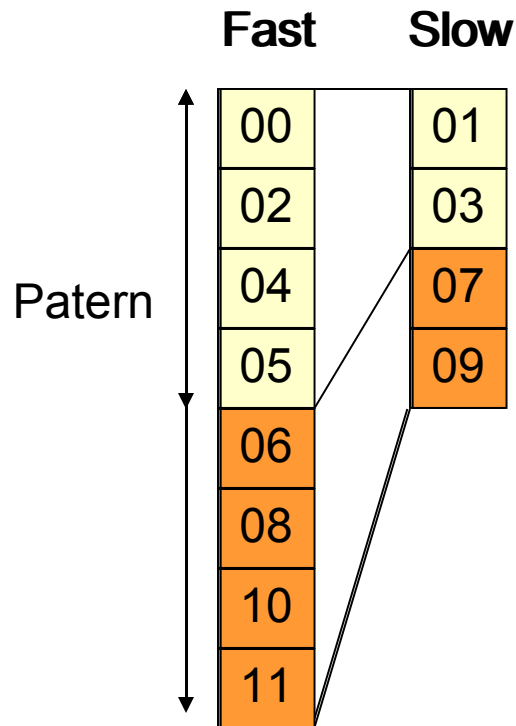
## **n Parallelism within a request**

- | Requests have to be large
  - ü The sub-request of each disk has to be large
  - ü Seek + search + transfer in all disks

## **n Parallelism between requests**

- | The number of disks has to be large
  - ü Compared to the average number of disks used in a request

# AdaptRaid0: An Example



## n Basic idea

- | Load each disk depending on its characteristics

## n Example

- | 1 fast disk
  - ü Size = S
  - ü Performance = P
- | 1 slow disk
  - ü Size = S/2
  - ü Performance = P/2

# ***AdaptRaid0: The Parameters***

## **n Utilization factor (UF)**

- | One factor per disk
  - ü Larger disks have more blocks?
  - ü Faster disks have more blocks?

## **n Lines in pattern (LIP)**

- | We define a pattern using the UF
  - ü Large patterns allow more requests with good disks
  - ü Small patterns allow a better distribution

# *AdaptRaid0: The Algorithm*

## n **Algorithm**

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

## **n Parameters**

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

## **n Simulation**

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

## **n Reference systems**

- | Raid0 and OnlyFast

## **n Disks**

- | Fast disk
  - ü Seagate Barracuda 4LP (4.339 Gbytes)
- | Slow disk
  - ü Seagate Cheetah 4LP (2.061 Gbytes)

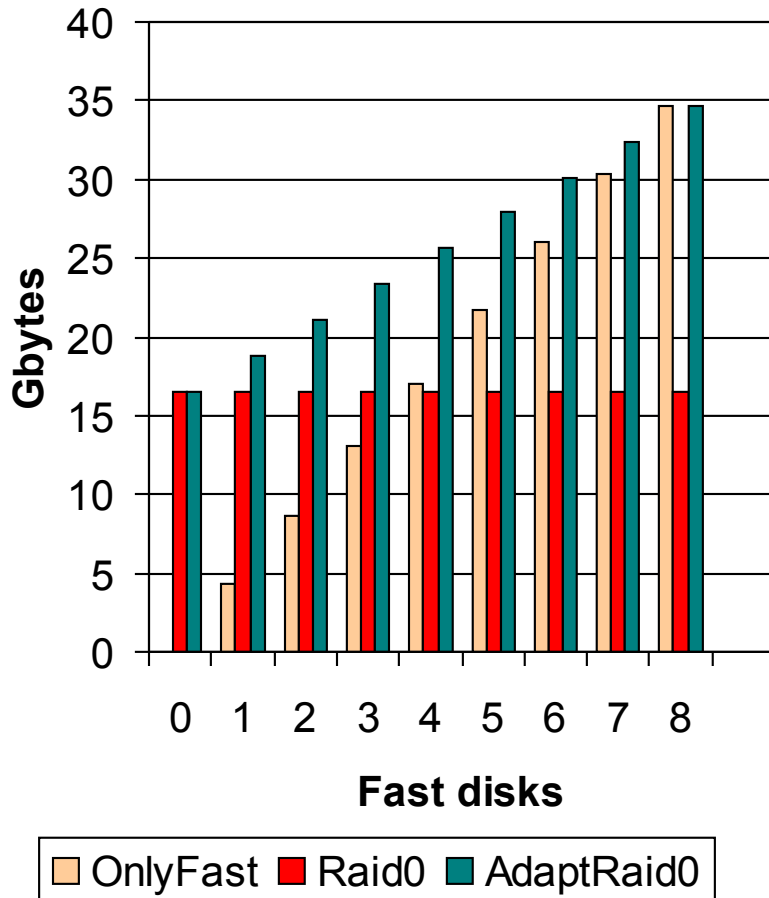
## **n Bus**

- ü 10us latency
- ü 100Mbit/s bandwidth

## **n File system**

- ü 10 requests in parallel

# Capacity Evaluation



## n Raid0

- | Constant capacity
- ü Small

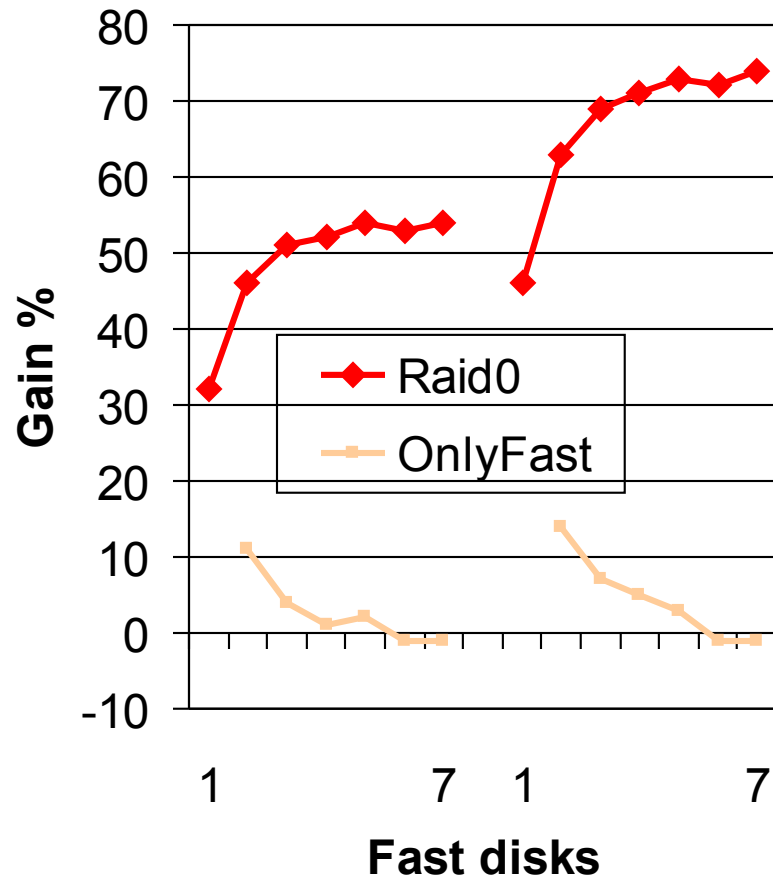
## n OnlyFast

- | Small capacity with few disks

## n AadaptRaid0

- | Offers the best size

# Performance Evaluation (8 disks)



## n Raid0

- | Does not use

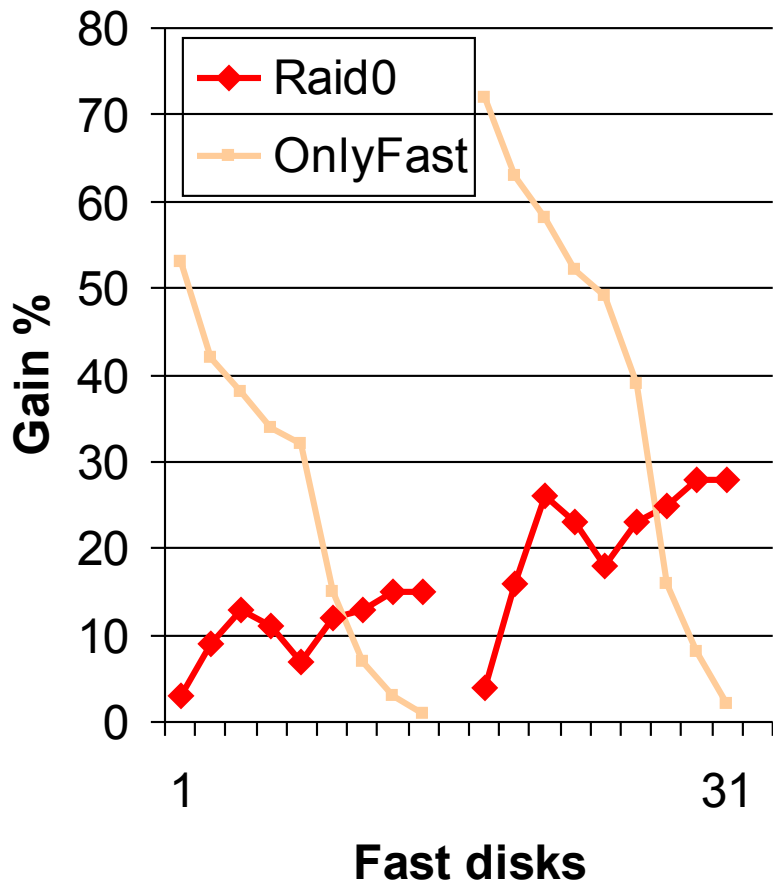
- ü Characteristics of good disks

## n OnlyFast

- | Does not use

- ü Parallelism between requests

# Performance Evaluation (32 disks)



## n Raid0

- | Does not use
  - ü Characteristics of good disks
- | It uses
  - ü Parallelism between requests

## n OnlyFast

- | Does not use
  - ü Parallelism between requests

## **n AdaptRaid0**

### **| Performance**

- ü It knows how to use the disks
- ü Allows parallelism

### **| Size**

- ü It uses all the available capacity

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