

# Cellular Disco

*SOSP'99 - Kinshuk Govil, Dan Teodosiu,  
Yongjang Huang, Mendel Rosenblum*

*Presented by:  
Himanshu Raj*

## Random Stuff

- Special project on Xen/Linux
- Bunch of virtualization related references on <http://swiki.cc.gatech.edu:8080/phd/798>
- Cool OS links
  - Understanding Linux Memory Model
    - <http://www-128.ibm.com/developerworks/linux/library/l-memmod/>
  - Understanding the Linux Memory Manager
    - Book available for free download from <http://www.skynet.ie/~mel/projects/vm/>

## Problem

- Extending modern Operating systems to run efficiently on shared memory multiprocessors.
- Software development has not kept pace with hardware development.
- Common operating systems don't utilize resources efficiently.
  - Poor scalability.
  - Suboptimal performance

## What we need....

- the system should be reliable
- it should be scalable
- it should be fault-tolerant
- it should not take too much of development time or effort.

## Traditional approaches

- Hardware partitioning
  - Makes physical clusters
  - Problems?
    - Lacks resource sharing
- Software-centric approaches
  - Modify existing OS
  - Develop new OS
  - Problems?
    - significant development time and cost

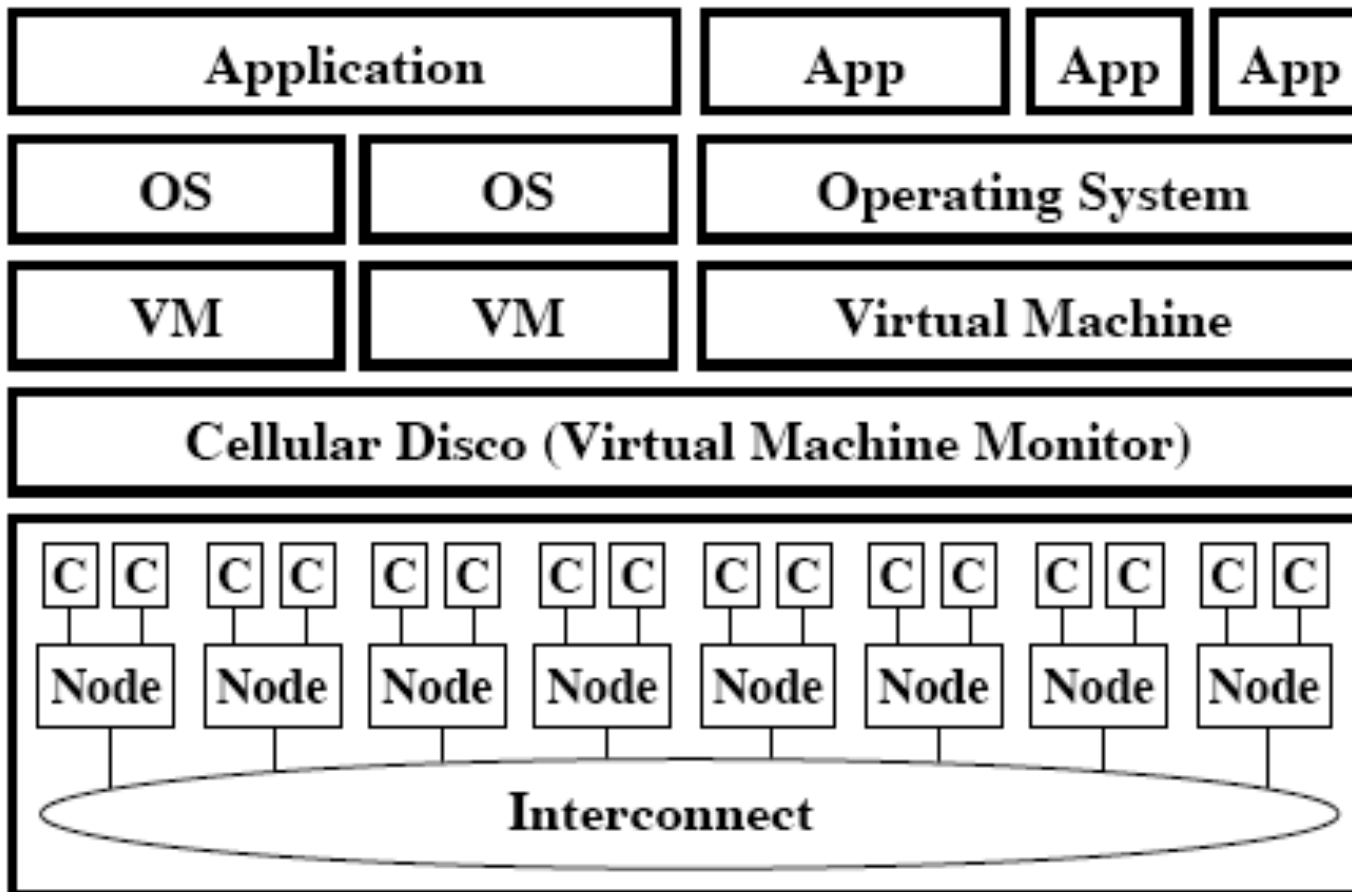
## Solution : Cellular Disco

- Extension of previous work - *Disco*
- Uses the concept of Virtual machine monitors
- Partitions the multiprocessor system into virtual clusters.

## Issues it addresses

- Scalability
- NUMA awareness
- Resource management
- Fault-tolerance
  - Soft crashes
  - *Hard crashes (new in cellular disco)*

# Basic Cellular Disco Architecture



## Prototype

- Runs on a 32-processor SGI-Origin 2000
- *Supports other SMPs based on MIPS R1000 architecture (e.g. FLASH).*
- Hosted design
- The host OS is made dormant and is only used to invoke some device drivers.

# Architectural Features of MIPS

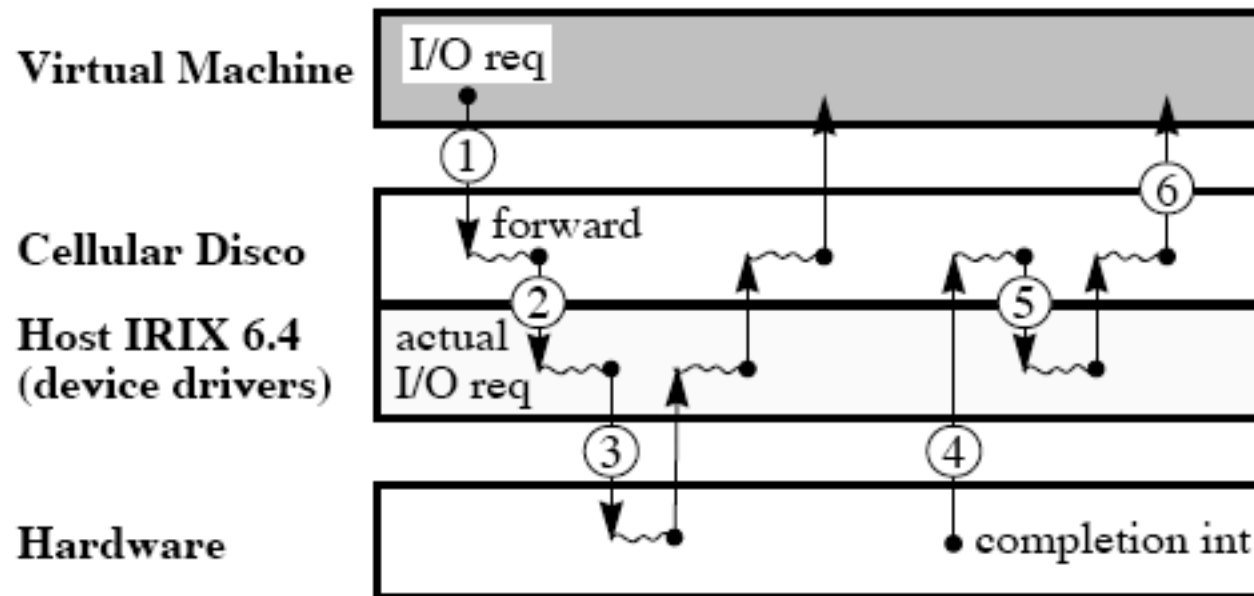
- Soft TLB
- 3 Levels of Protection
  - User mode
  - Supervisor mode
  - Kernel mode
- Compare with x86

## Hardware Virtualization

- Physical Resources - visible to a virtual machine
- Machine Resources - actual resources; allocated by Cellular Disco
- CD operates in the *kernel mode* of the MIPS processor
- Redirection of system calls

# Hardware Virtualization ...

- Hosted architecture
  - Piggybacked on IRIX6.4



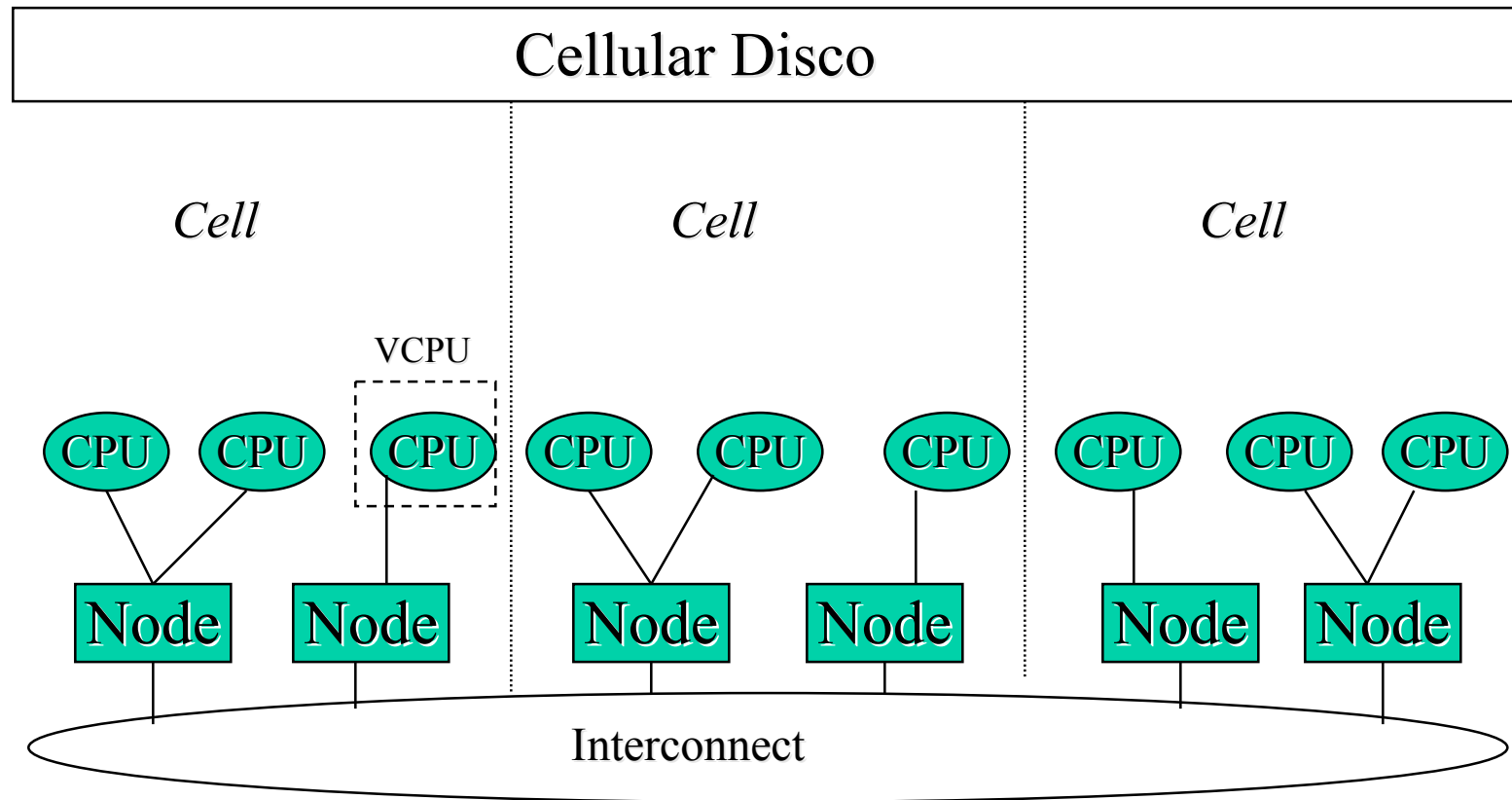
# Hardware Virtualization ...

- Memory
  - Use of L2TLB for fast translations

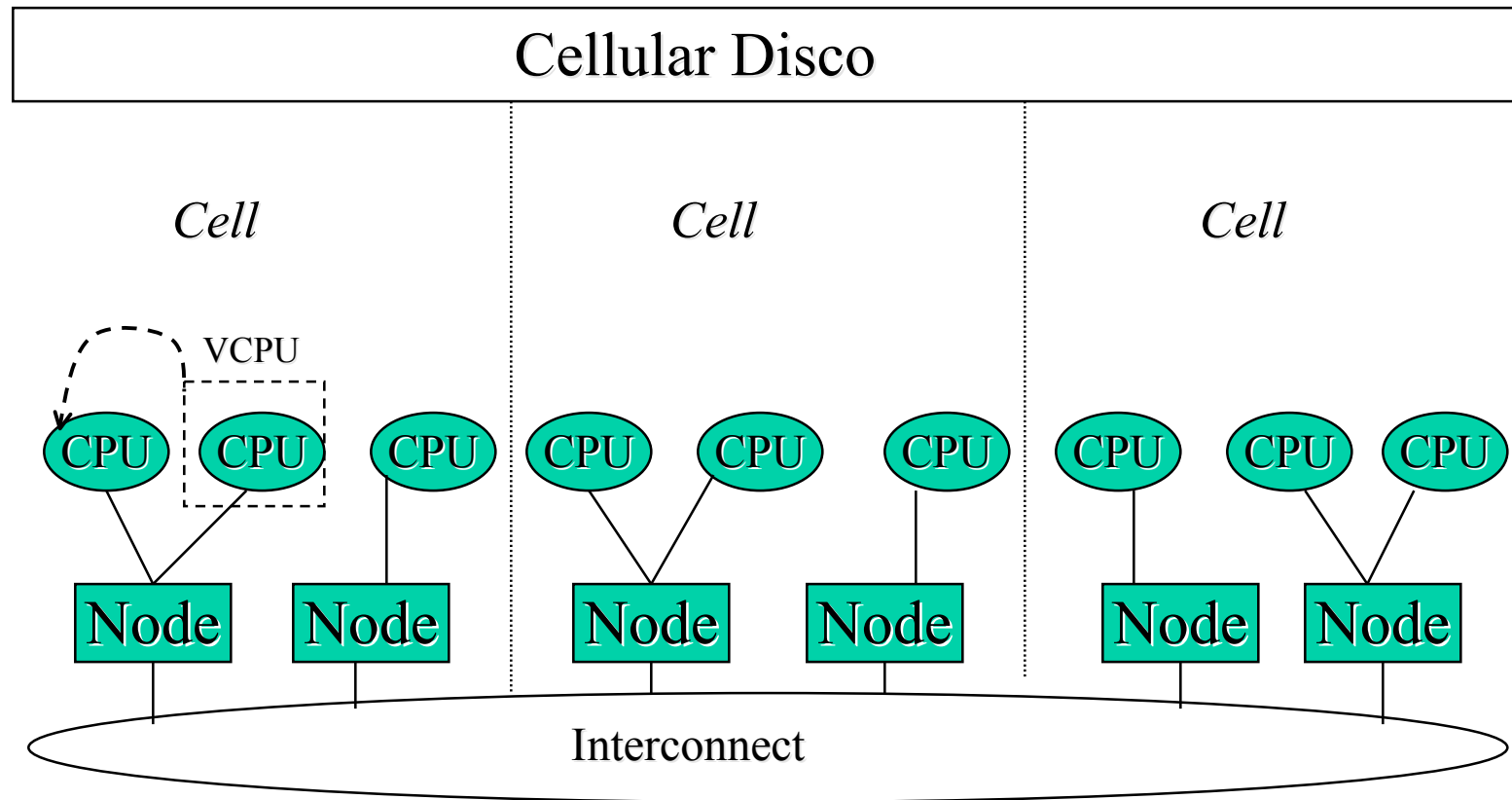
## Resource Management

- CPU management - Each processor maintains its own run queue
- Memory Management - Memory borrowing mechanism
- Each OS instance is only given as many resources as it can handle. Large applications are split and communications between the parts is established by using the shared-memory regions.

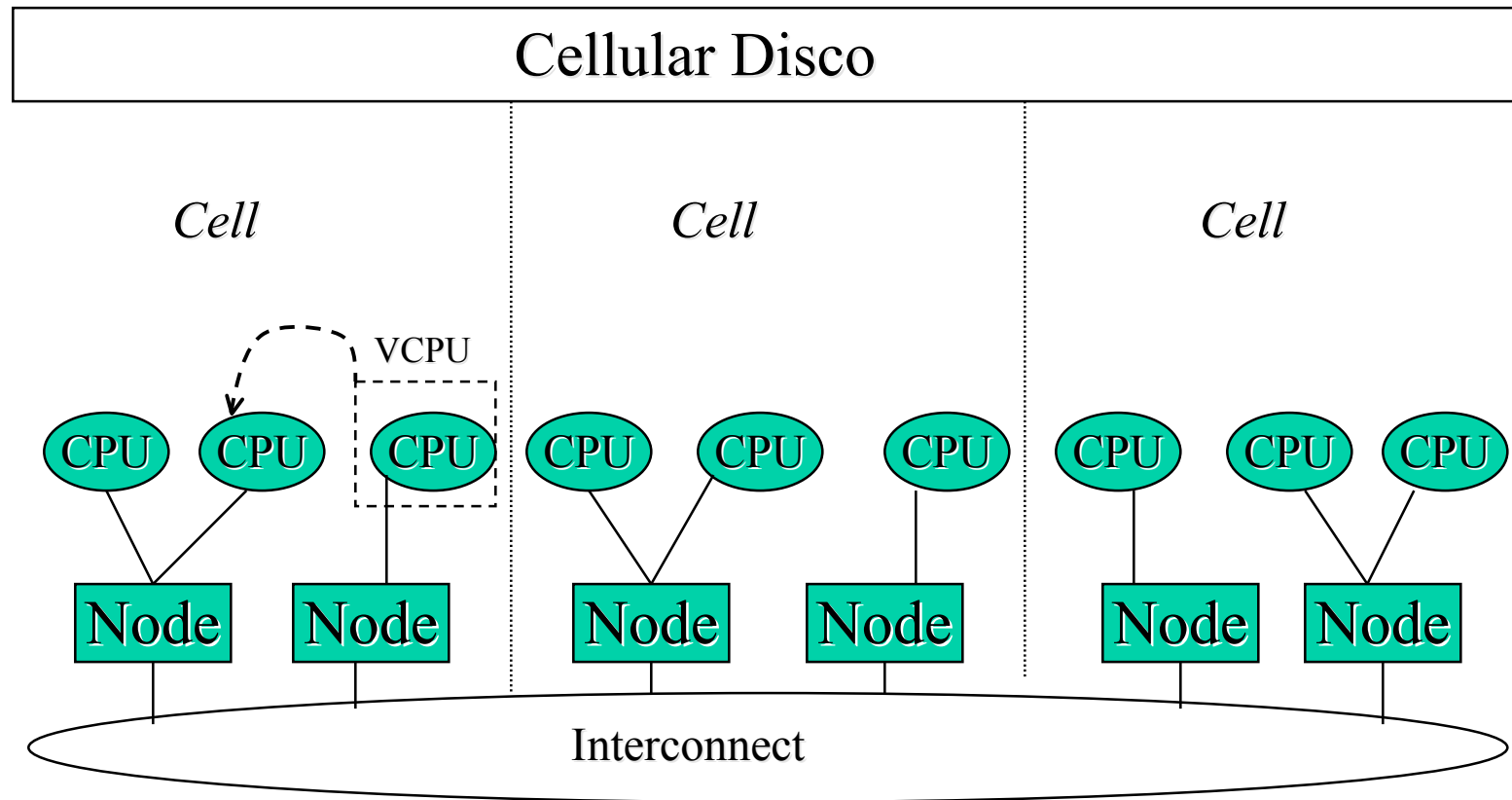
# VCPU migration



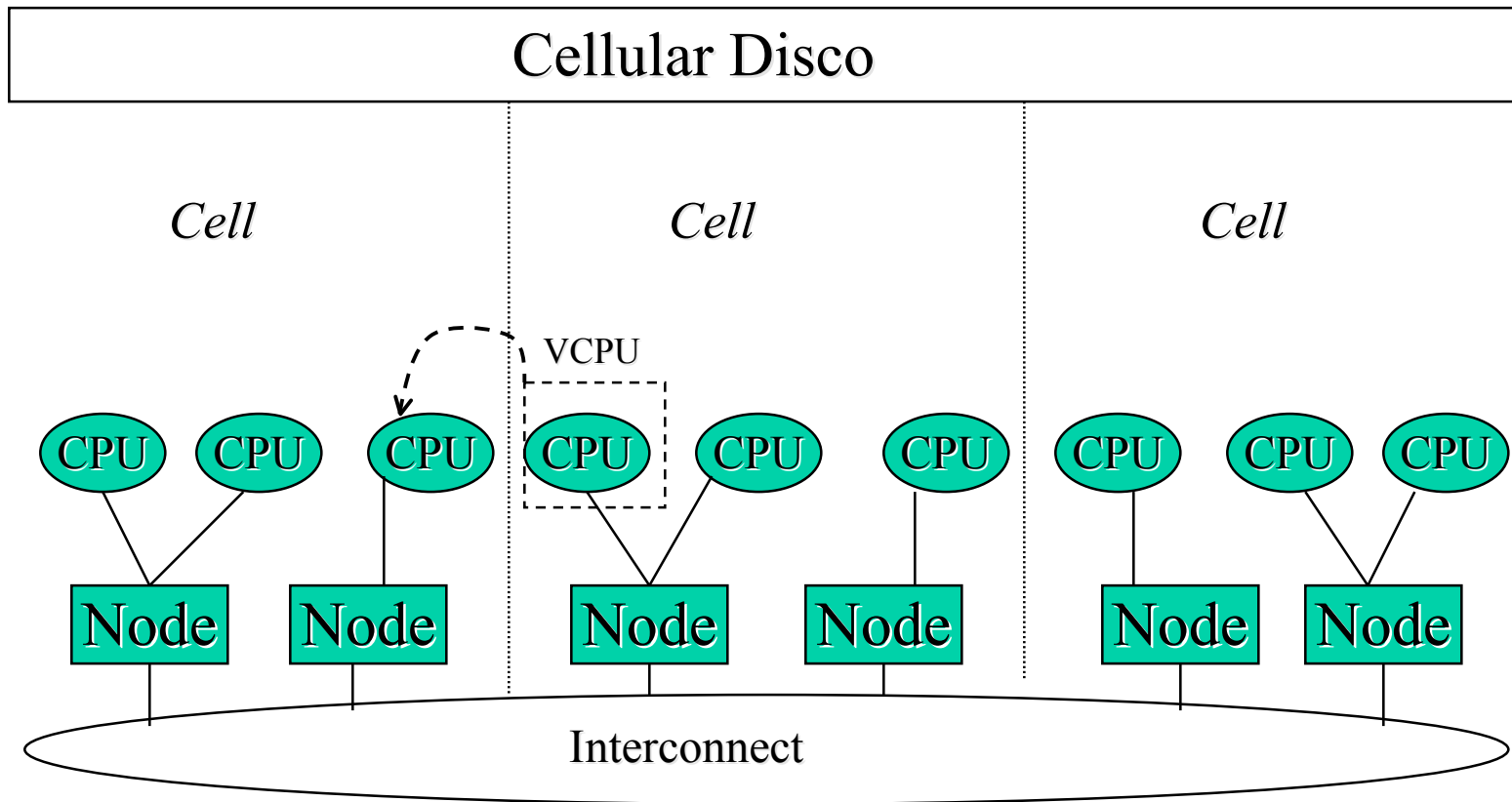
# Intra Node



# Inter Node

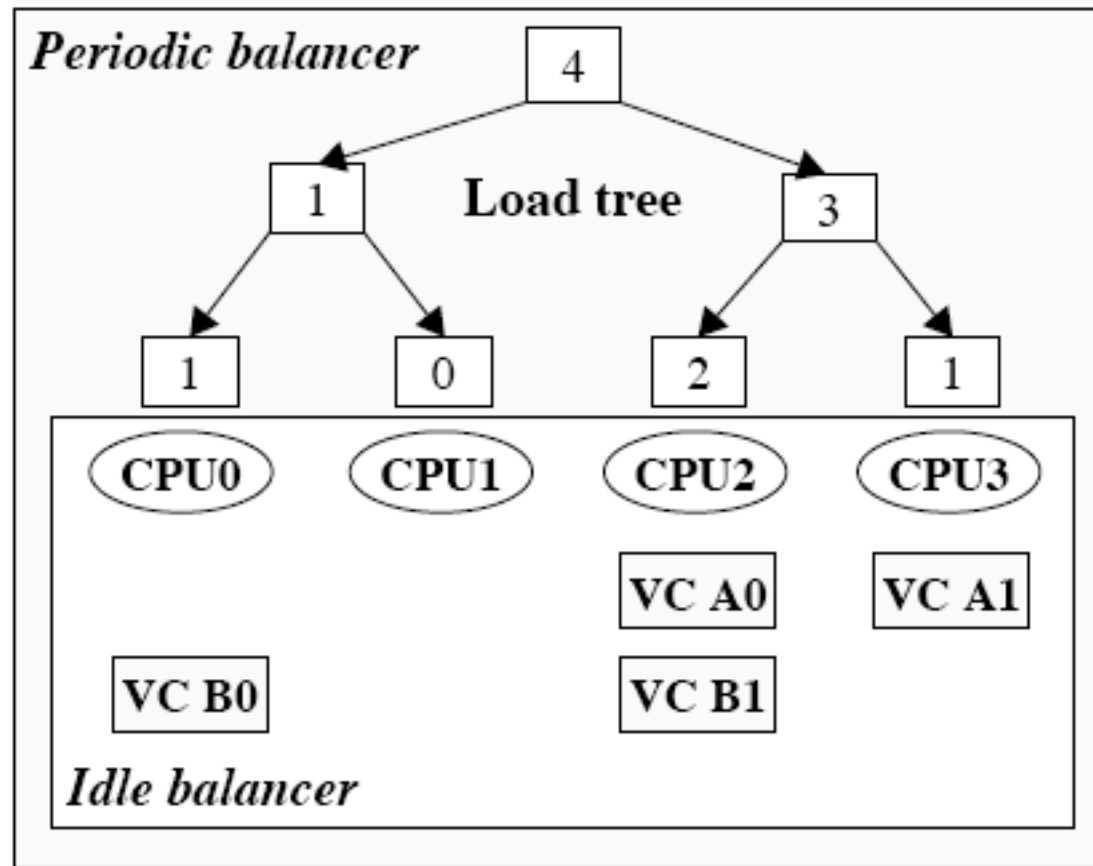


# Inter Cell



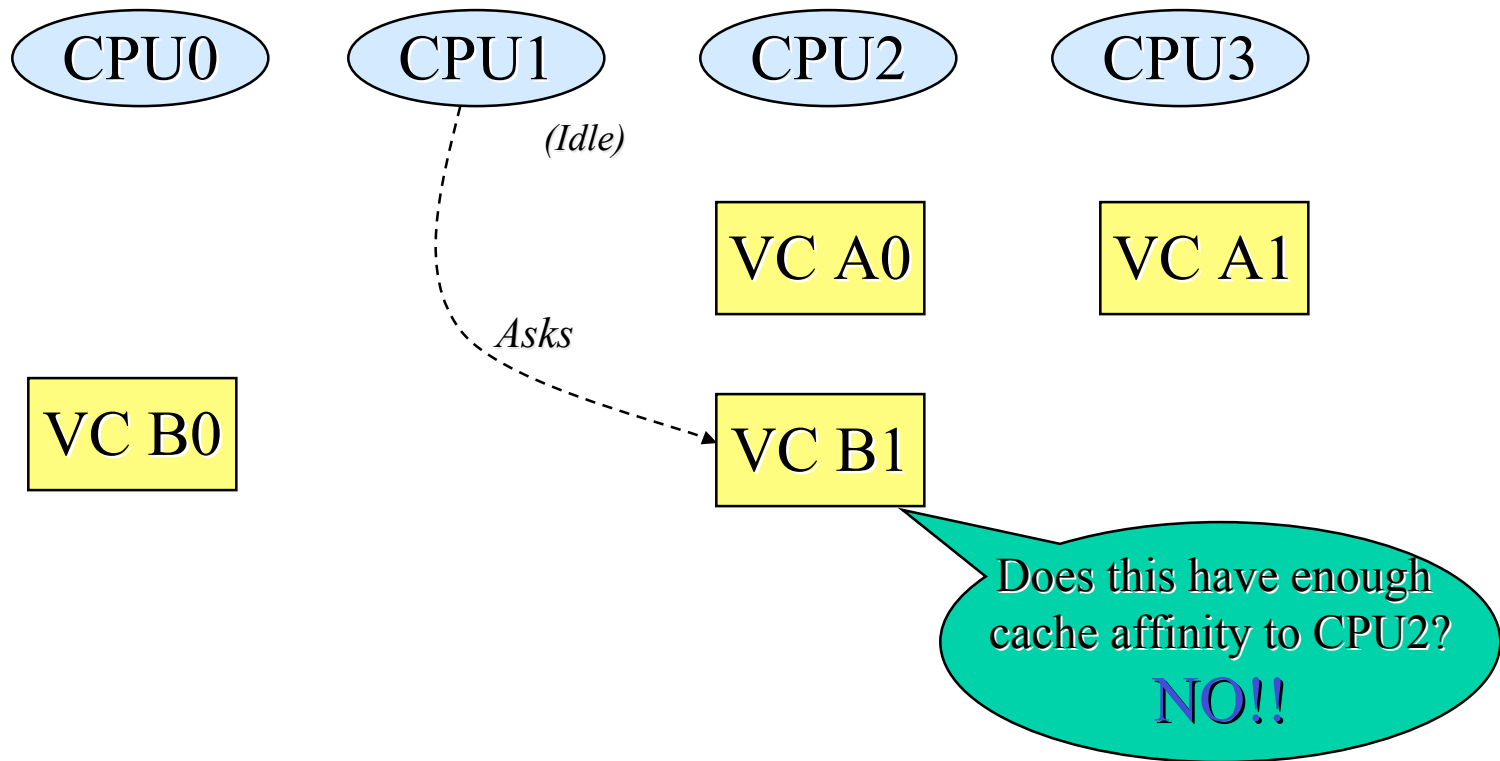
## CPU Management (contd.)

- CPU balancing : Idle Balancer  
Periodic balancer

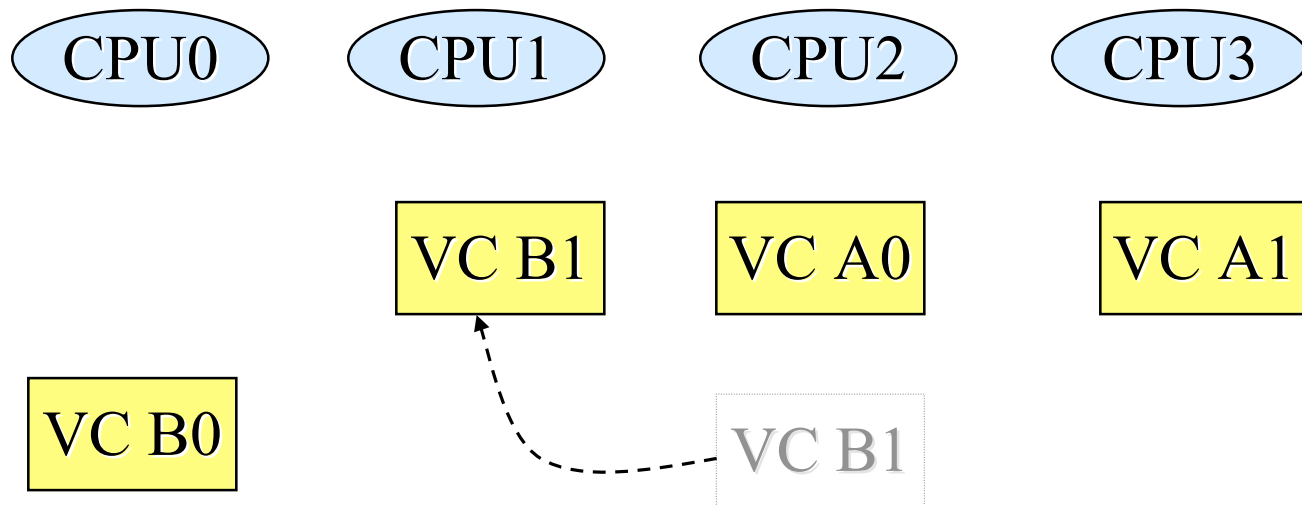




# Idle balancer

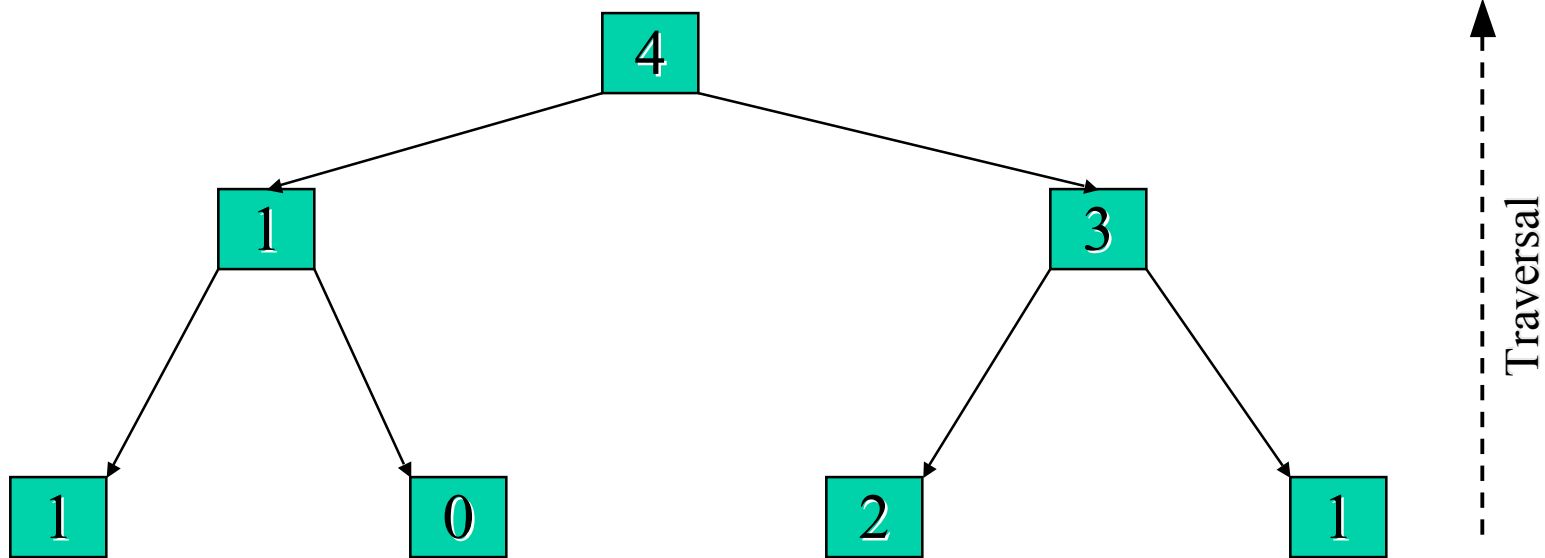


# Idle balancer



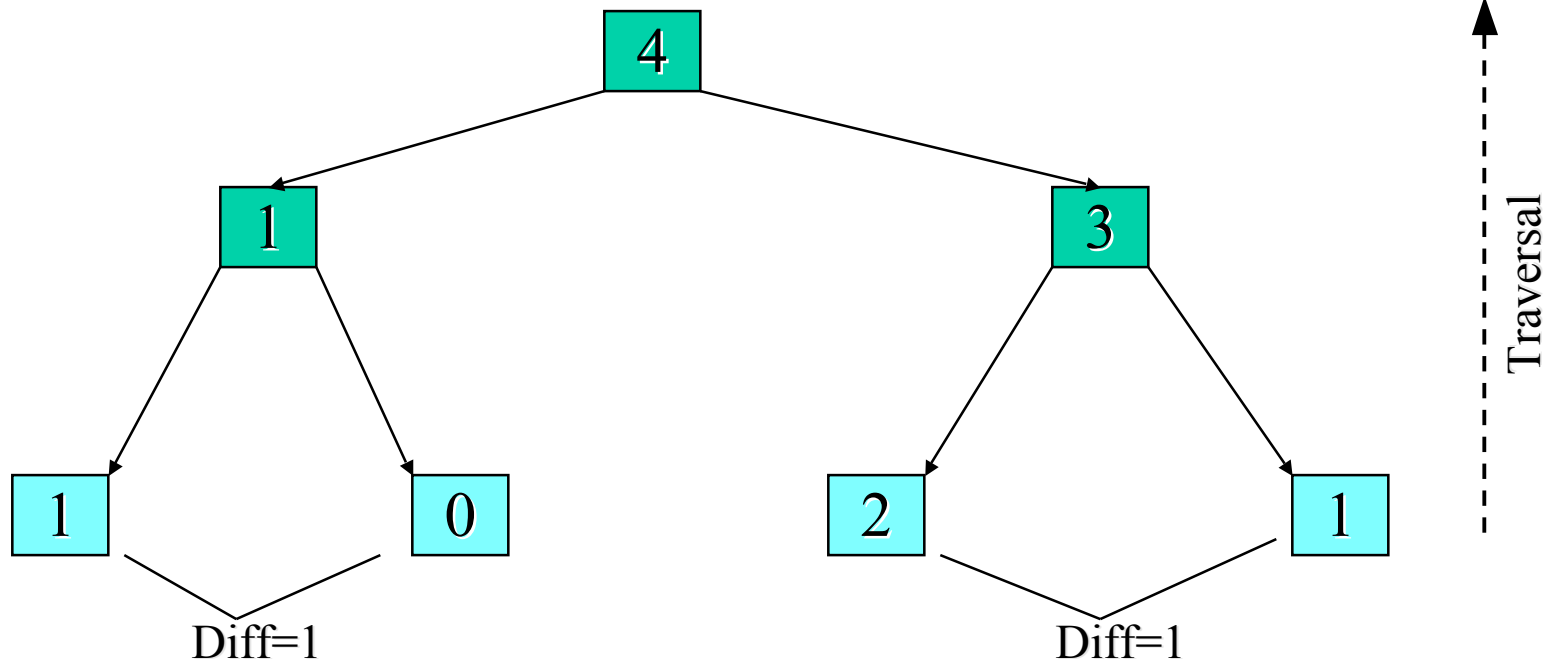
# Periodic Balancer

- Does depth-first traversal of the load tree



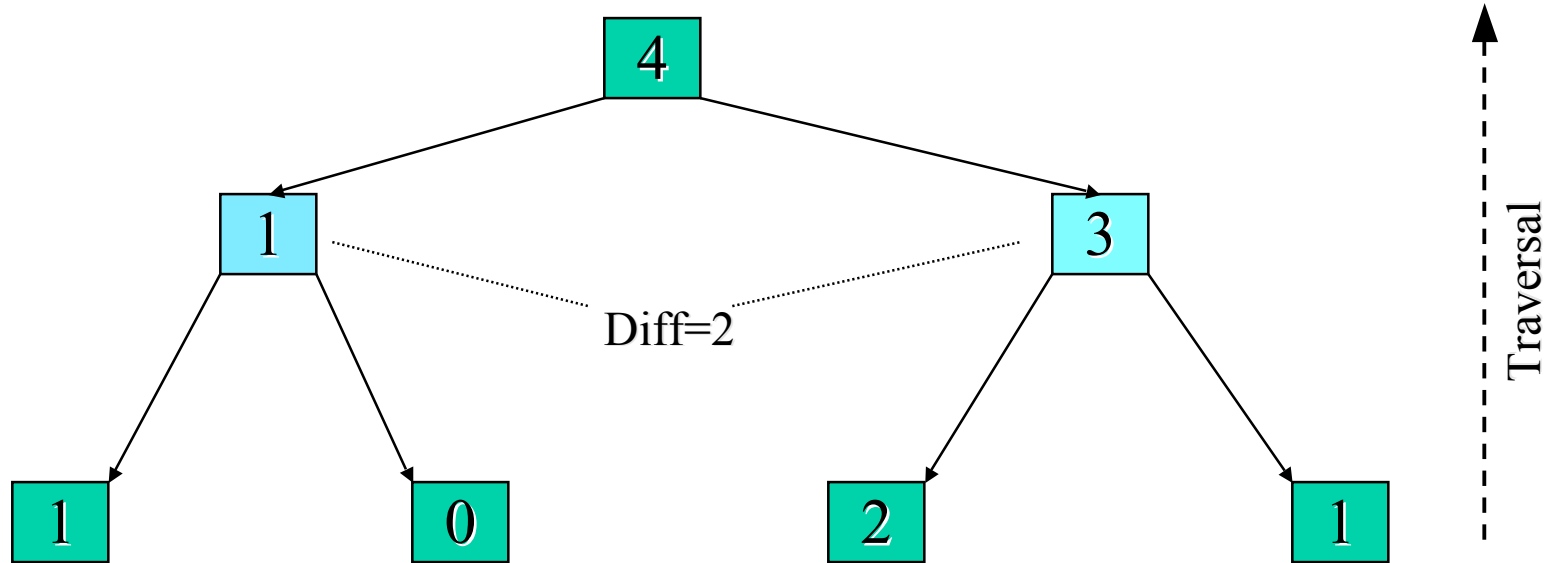
# Periodic Balancer

- Checks difference of 2 siblings, ignores if  $< 2$



# Periodic Balancer

- If  $\text{diff} \geq 2$  does load balancing if  $\text{benefit} > \text{cost}$



## Gang Scheduling

- For all the CPU's we select the VCPU that is to run on the physical CPU.
- The VCPU selected is the highest priority *gang-runnable* VCPU
  - all non-idle VCPU's of that VM are either
    - running or,
    - waiting on run queues of processors running lower-priority VM's.

## Memory Management

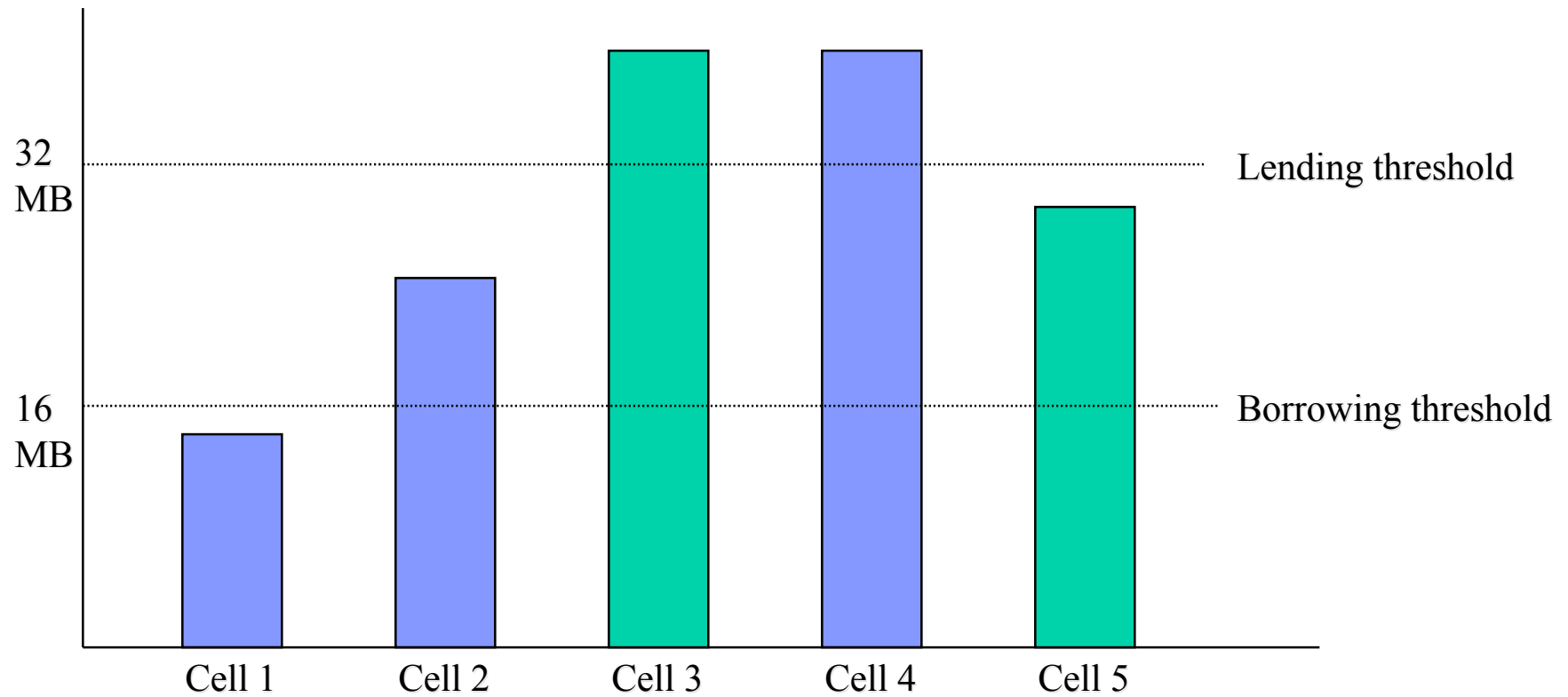
- Each cell maintains its own *freelist*, and allocates memory to other cells in its *allocation preference list* on request(RPC).
- Speed - 758  $\mu$ sec for 4 MB.
- A threshold is set for min. amount of local free memory
- As far as possible Paging is avoided.

## Memory Borrowing

- *freelist* - list of free pages in the cell
- *allocation preference list* - list of cells from which borrowing memory is more beneficial than paging.

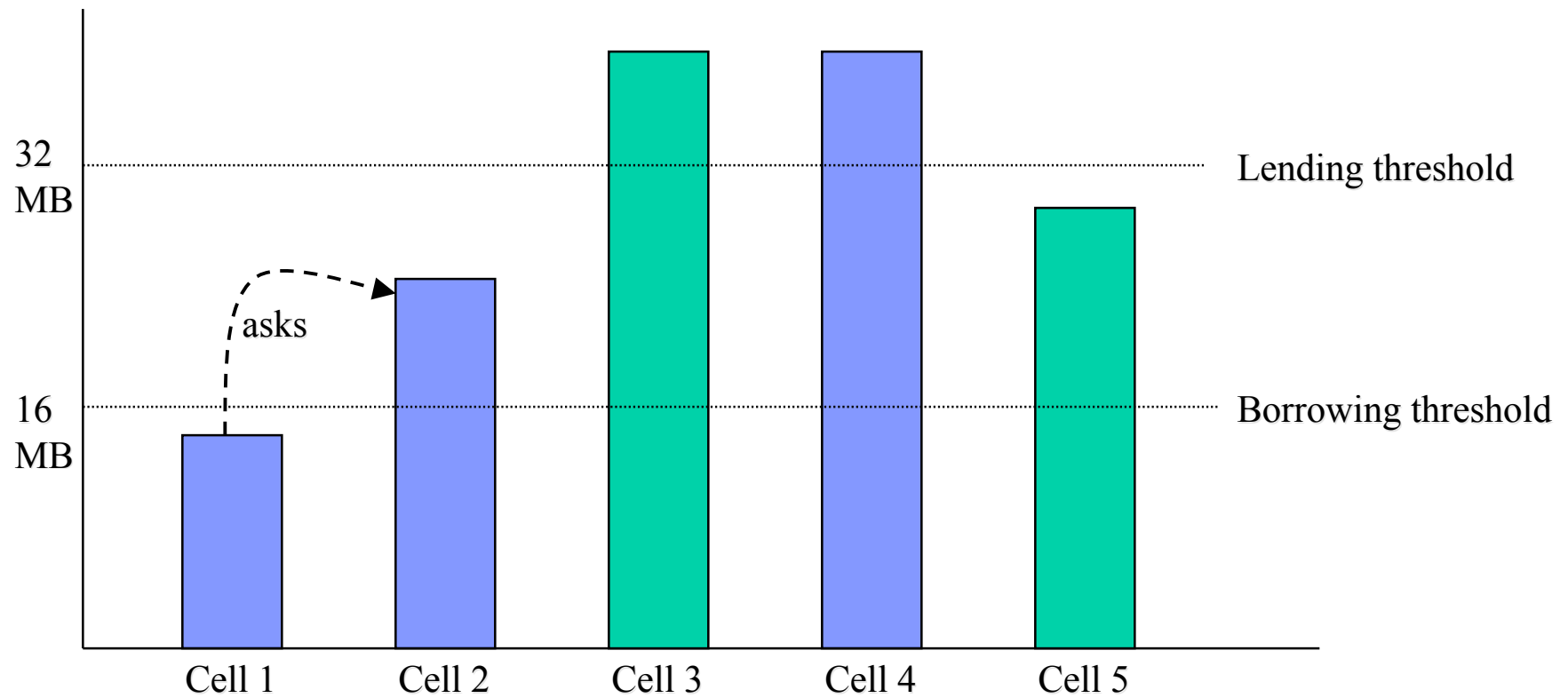
# Memory Borrowing

## *Freelist sizes*



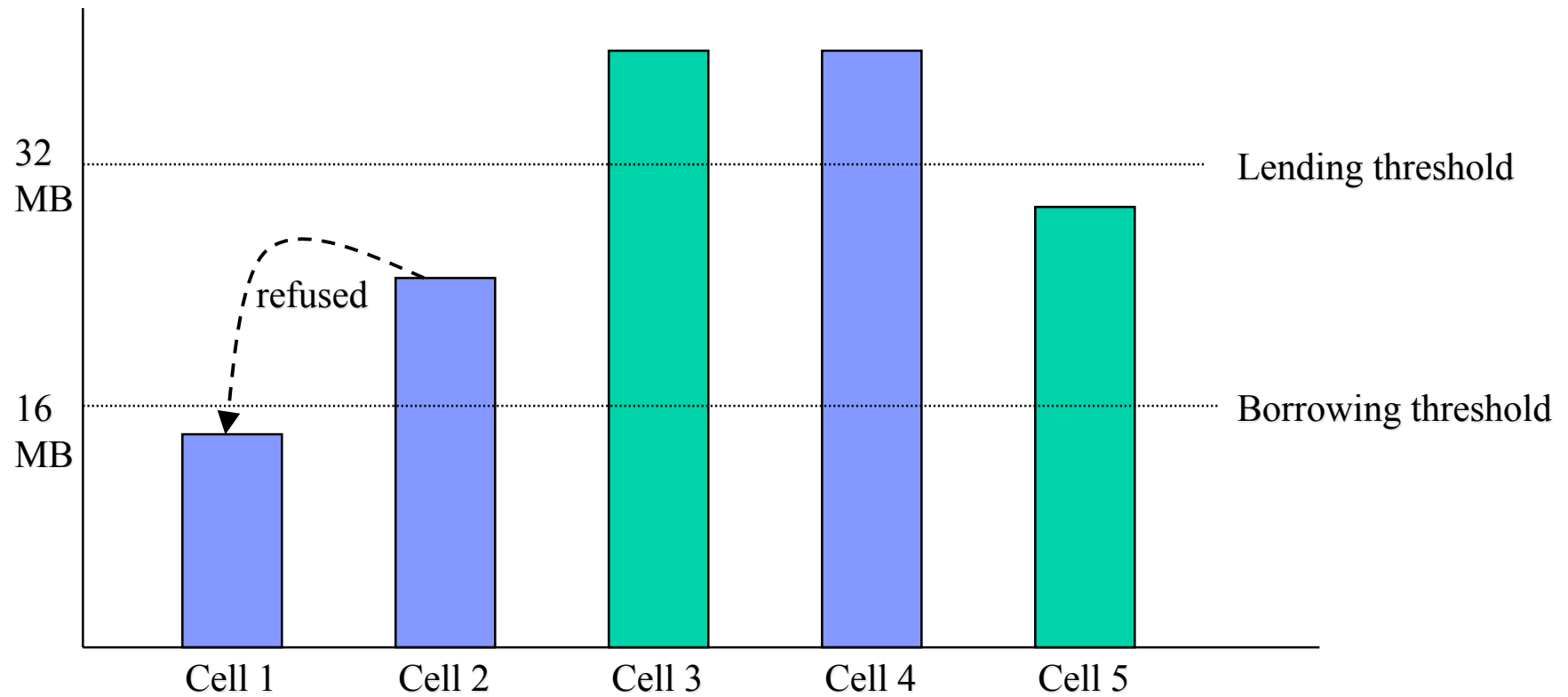
# Memory Borrowing

## *Freelist sizes*



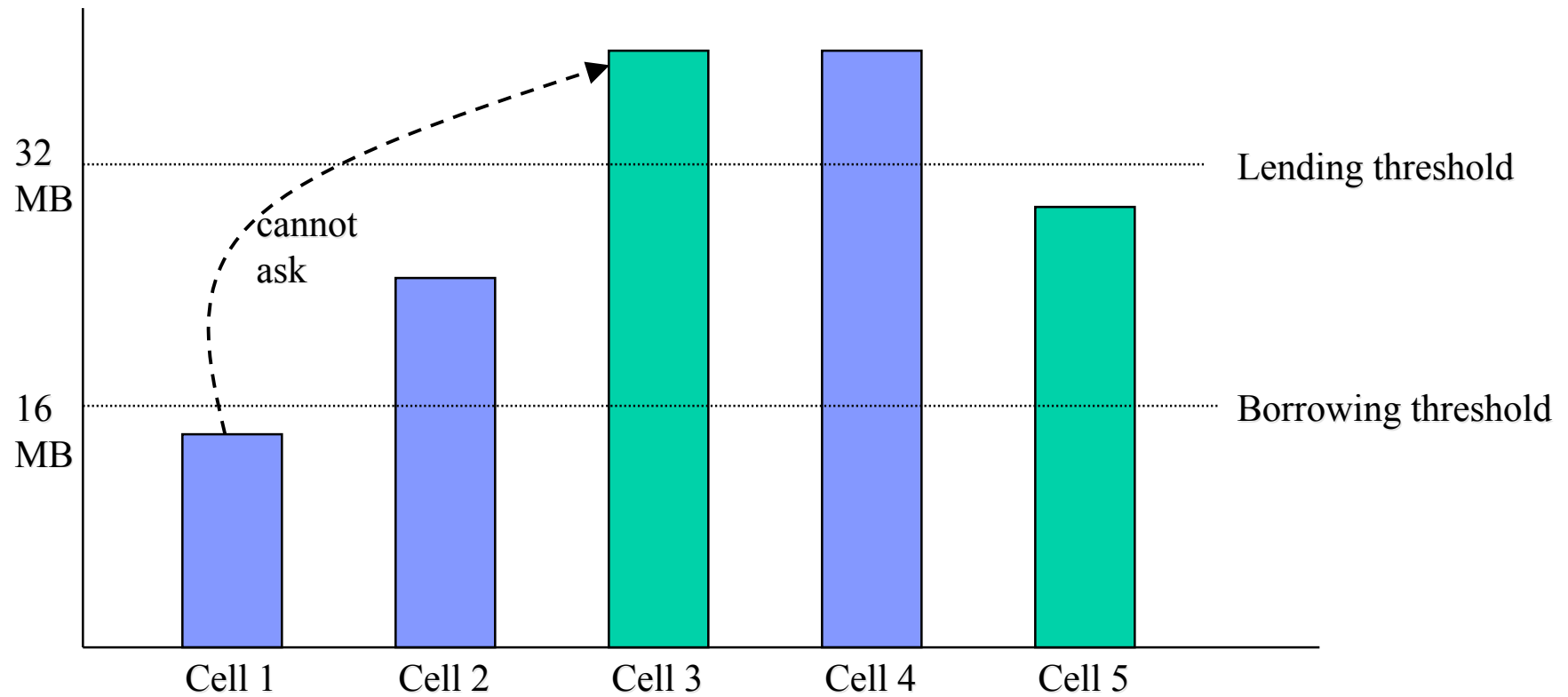
# Memory Borrowing

## *Freelist sizes*



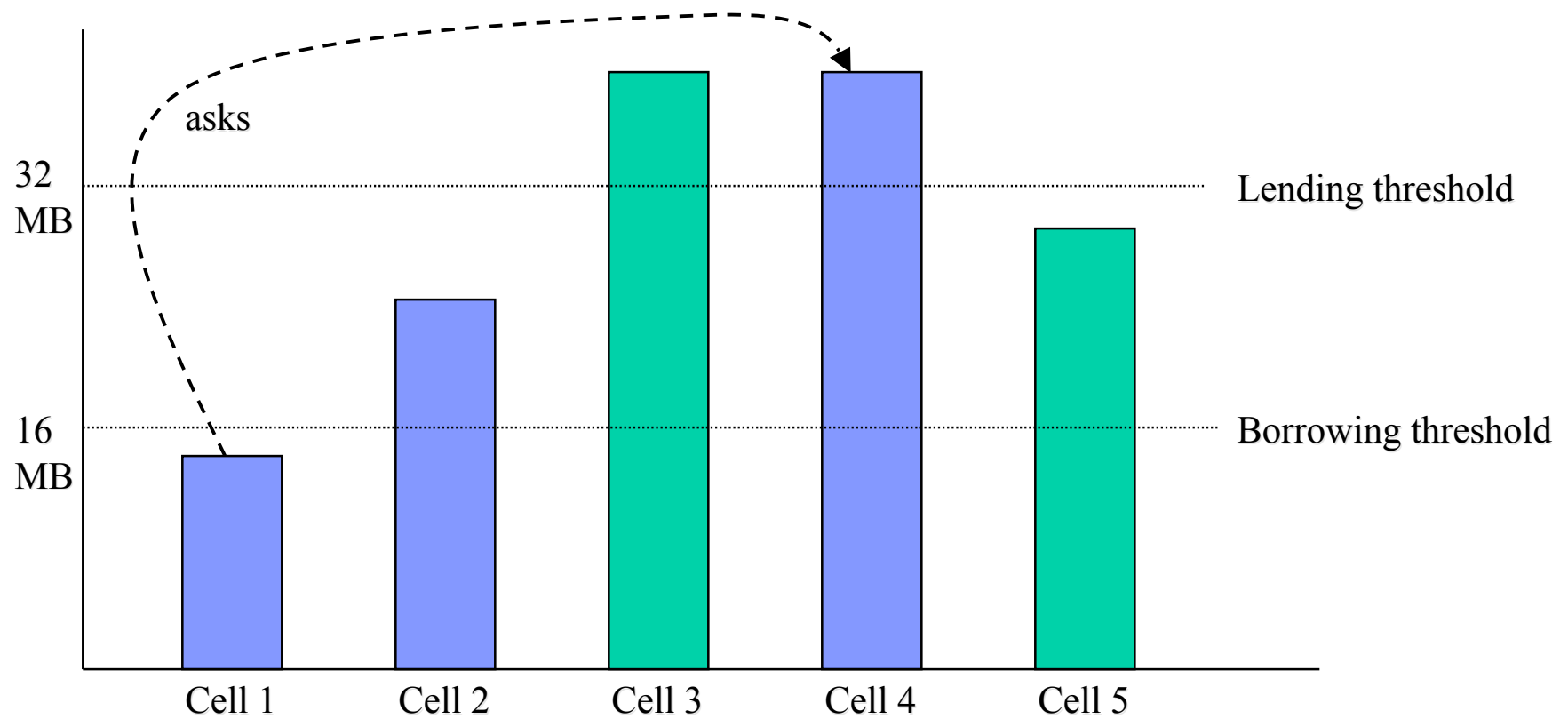
# Memory Borrowing

## *Freelist sizes*



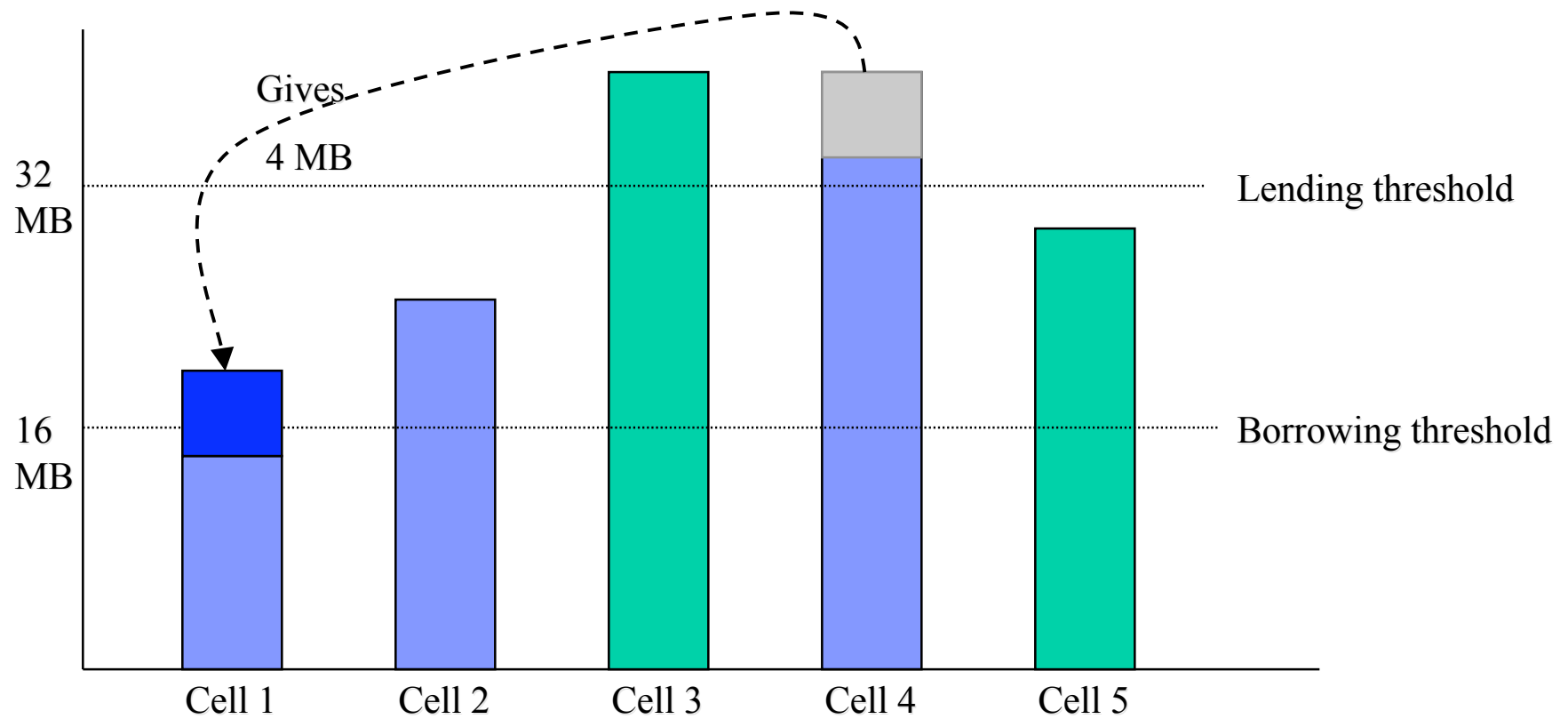
# Memory Borrowing

## *Freelist sizes*



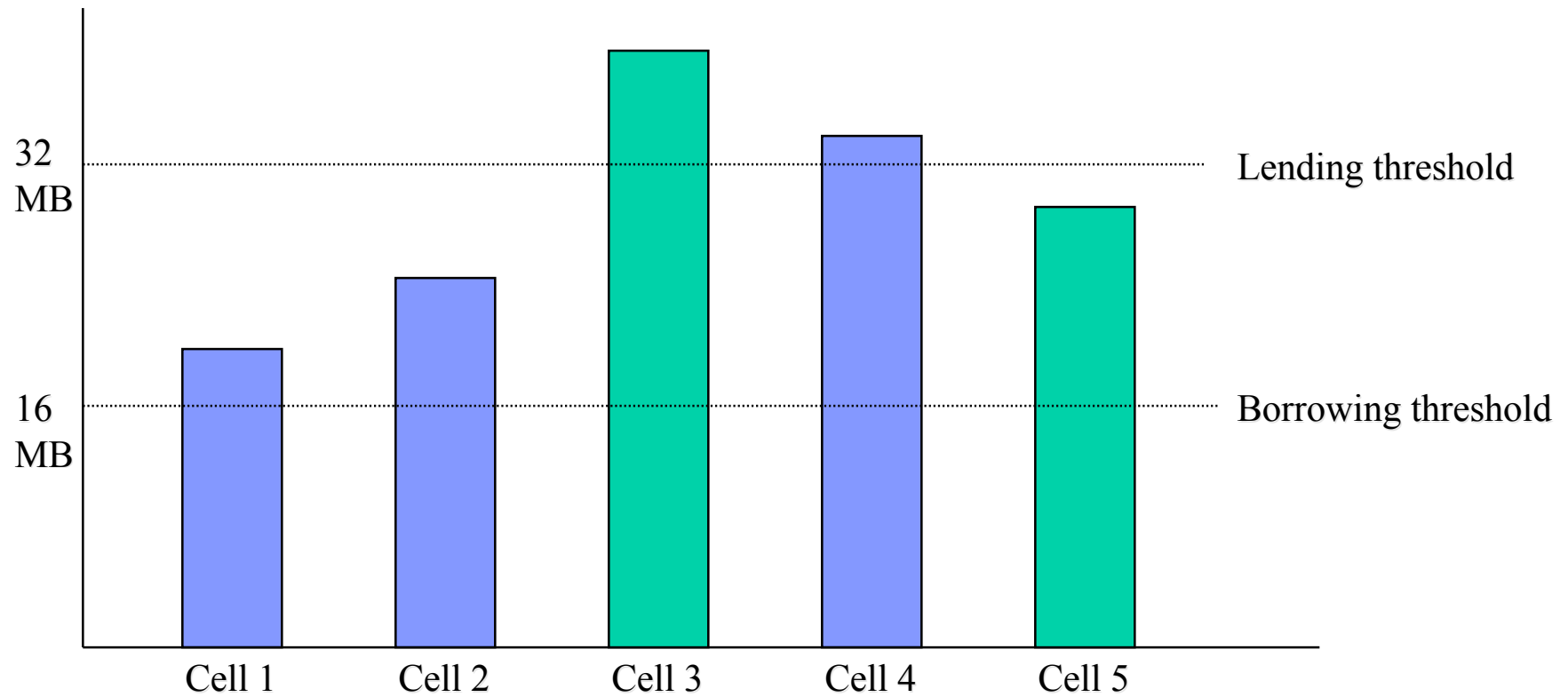
# Memory Borrowing

## *Freelist sizes*



# Memory Borrowing

## *Freelist sizes*



## Memory Management *(contd.)*

- Paging using second chance FIFO
- Page sharing information by some control data structure
- Cellular Disco traps all read and write requests made by the Operating Systems
- Problems?
  - Double paging
  - Use virtual paging disk

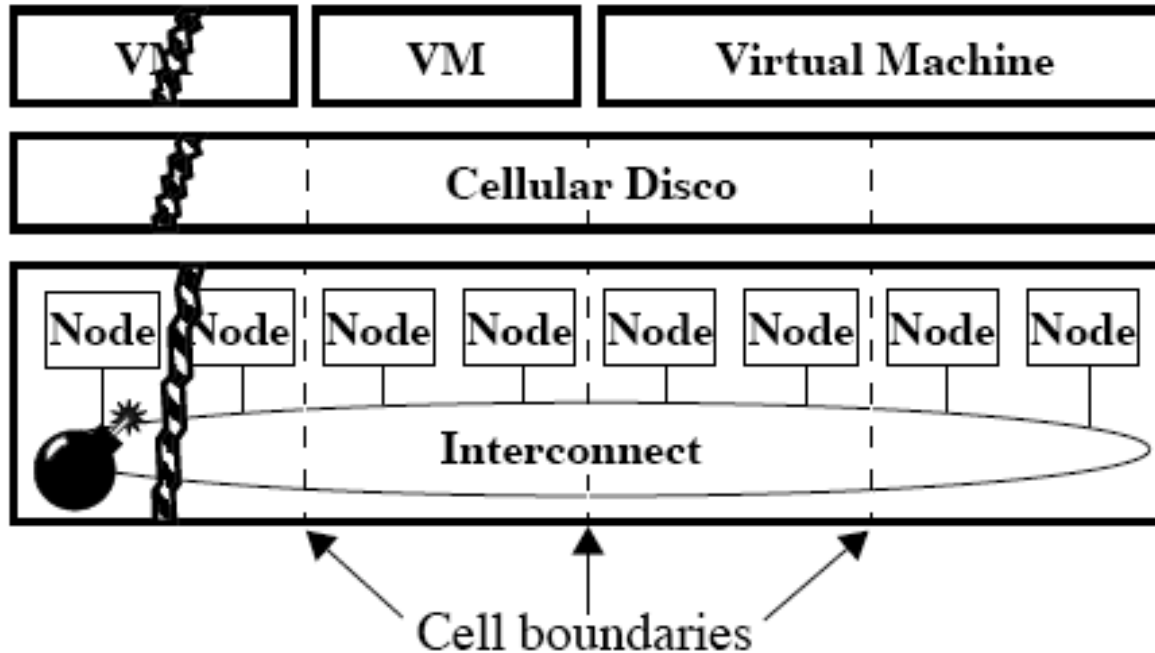
## Second-chance FIFO

- A reference bit is added to each page in FIFO scheme
- Every time the page is accessed the bit is set to 1
- If the page is selected by FIFO, and the reference bit is 1, then it is set to 0 and another page is looked for.
- A page is the target page if it is selected by FIFO and the reference bit is 0

## Hardware fault-containment

- Failure rate increases with increase in processors.
- Internally structured as a set of semi-independent cells.
- Failure in one cell does not impact VM's running in other cells (*localization of faults*)
- *Assumption*
  - CD is a trusted software layer

# Cellular Structure



Fault in one cell does not affect others

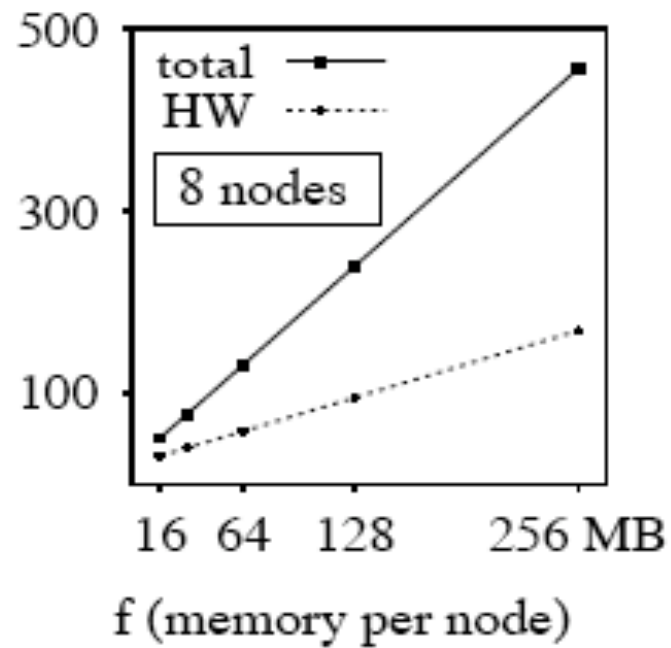
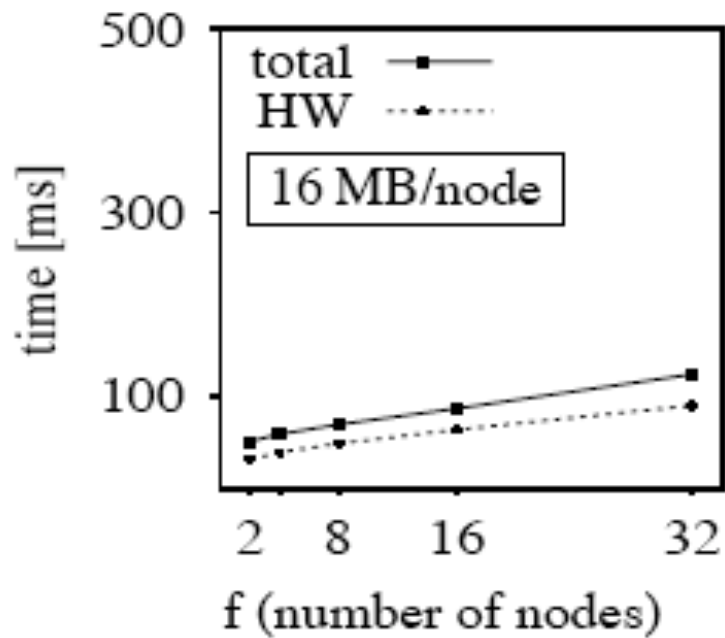
## Hardware fault-containment *(contd.)*

- Replication of CD on every cell
- Communication modes
  - Fast inter-processor RPC
  - Message
- Software fault containment already by virtue of hardware virtualization
- Simulation results on FLASH SMP
  - Why?

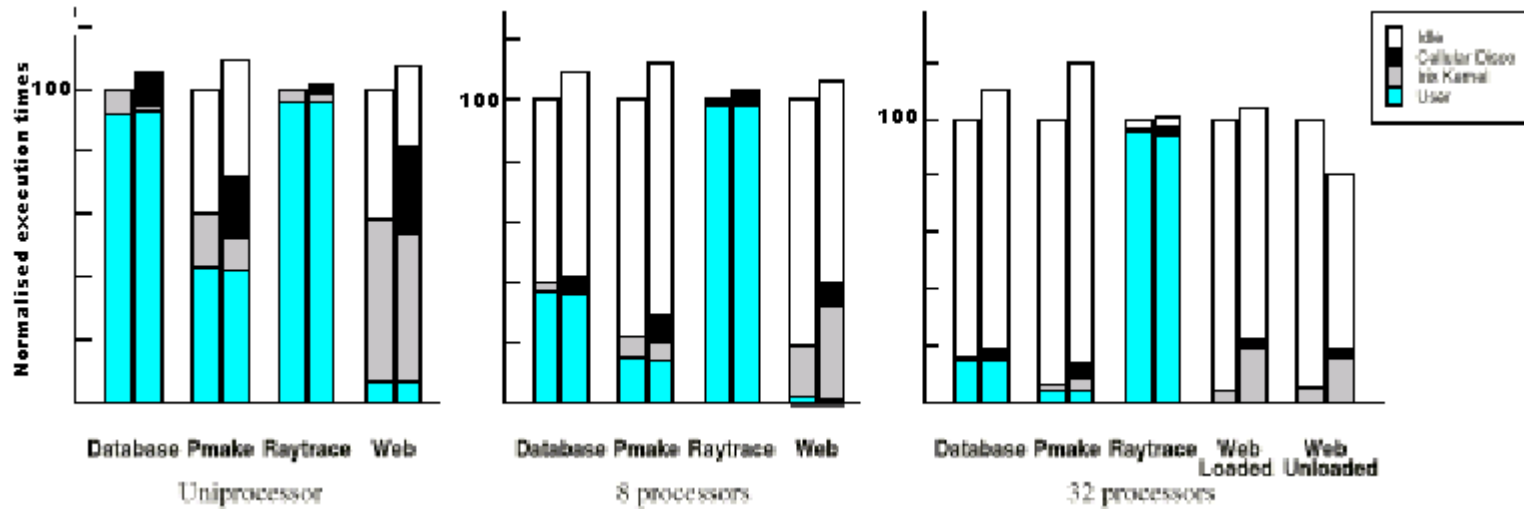
## Hardware-Fault recovery

- *liveset* - set of still functioning nodes.
- Failure - removal from *liveset*
- Recovery - insert back to *liveset*
- Virtual machines dependent on the failed cell are terminated.
- Memory dependencies are updated when a cell fails.

## Fault-Recovery overhead



# Virtualization Overheads



*(the first column shows the exec. Time on IRIX 6.4 and the second shows the exec. time on Cellular Disco).*

## Conclusion

- Cellular Disco - Scalable VMM for large SMP machines
  - Architectural support for easy platform virtualization (MIPS)
  - Time sharing for CPUs
  - Static partitioning of memory with some dynamic memory management
- HW Fault Tolerance via Cellular Design