Introduction to Thin Provisioning Proposal

(and FLASH pre-write)

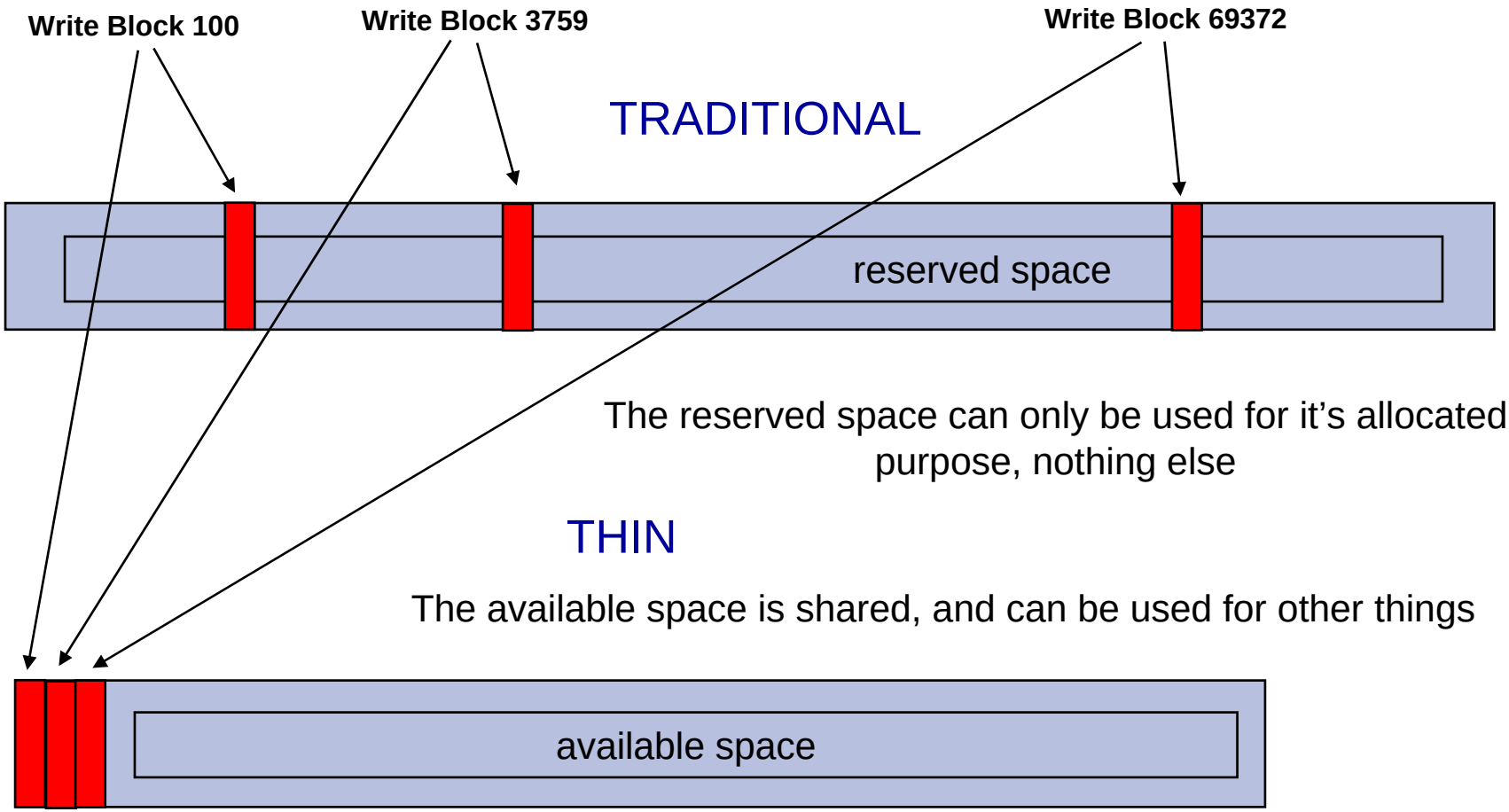
Frederick Knight

Thin Provisioning – What is it?

- **Traditional storage devices reserve physical storage for every possible disk address (block) whether it contains useful data or not.**
- **Hosts rarely use 100% of possible addresses**

- **“Thin” devices allocate physical storage only for those blocks that contain useful data.**
  - **Useful data is data that has been written.**
- **The Storage is virtualized**

Storing LUN data



Thin Provisioning – Why?

- **“Thin” devices require less real H/W**
- **“Thin” devices require less space**
- **“Thin” devices require less power**
- **“Thin” devices substantially reduce TCO**

- **Storage virtualization brings substantial benefits to customers just like host virtualization**

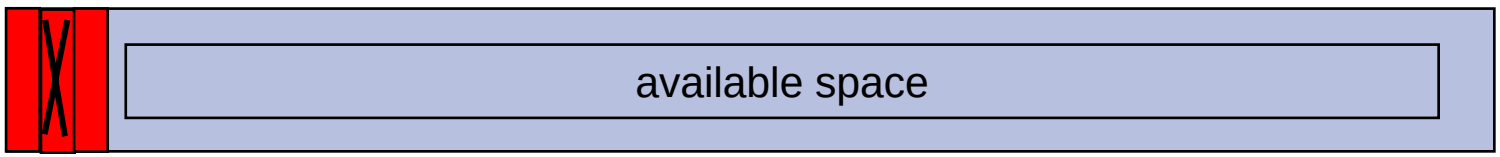
Thin Provisioning – Opportunity

Deleting a file – reusing deallocated space

Since the space is already reserved deleting a file has no impact on utilization

THIN

Reclaiming the now unused space is now a desired action

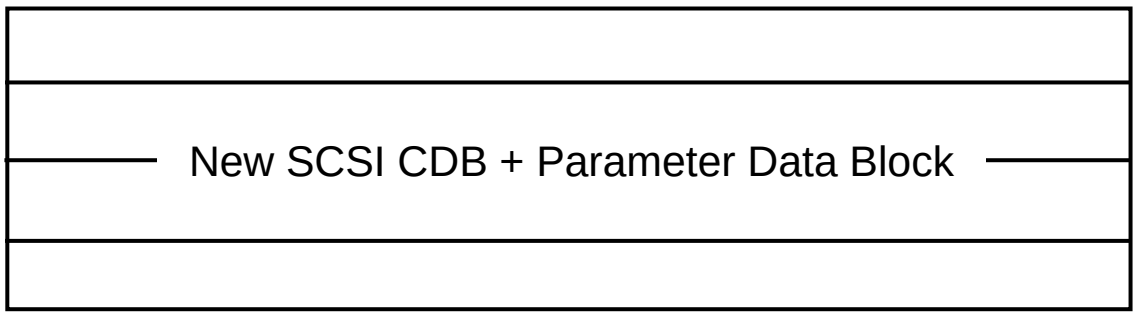


Thin Provisioning – Issues

- Today, hosts assume the traditional model
- Hosts don't tell storage when the data in a block is useful and when it is not (such as when a file is deleted from a file system).
- Storage has no standard way to inform the host of the provisioning status changes.
- Need to create standard APIs for Thin Provisioning

Standard APIs

- **Create a new SCSI Command for Hole Punch/Delete/Pre-Write Function**
  - **Block numbers not in CDB (in parameter data block)**



LBA
Length
LBA
Length
.....

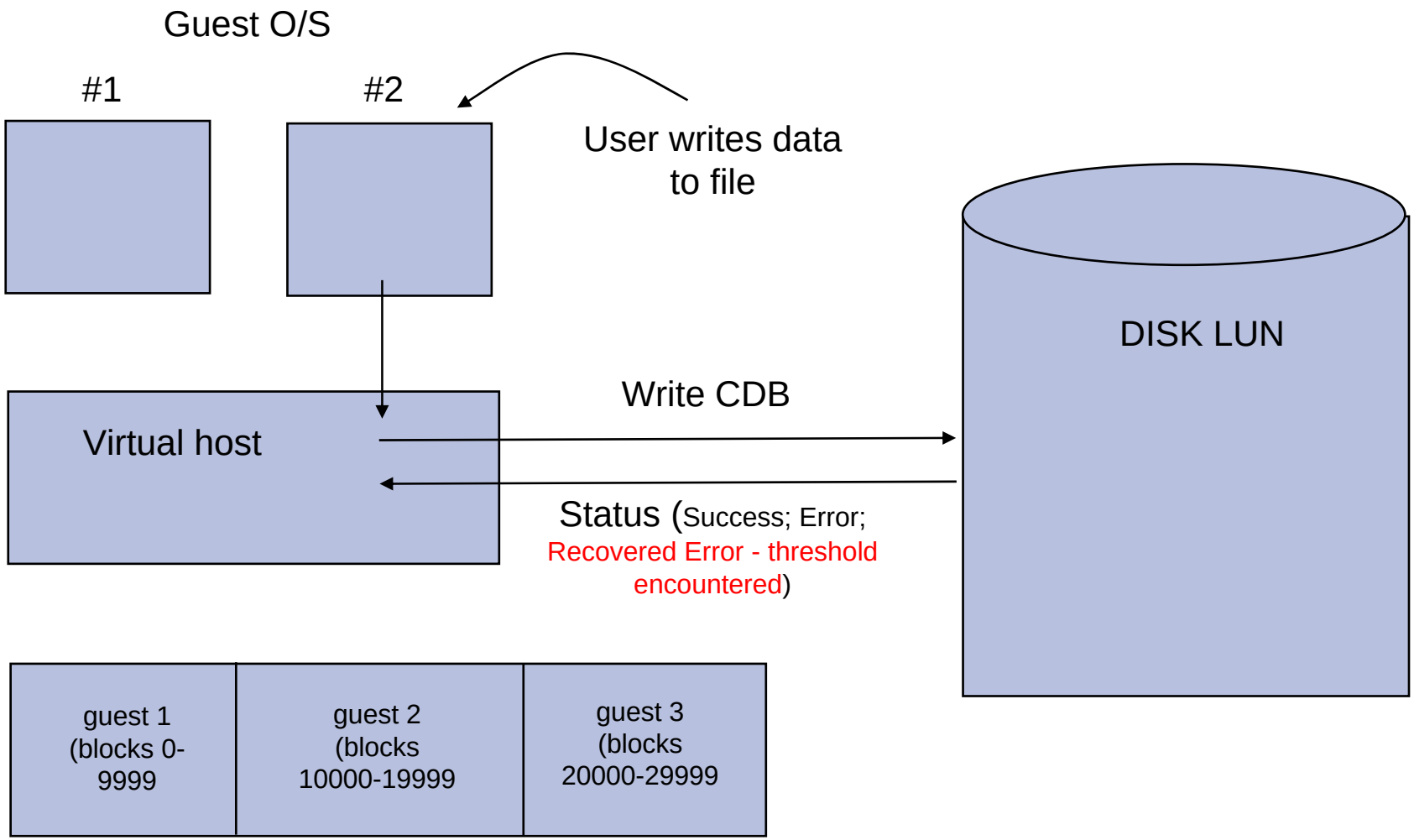
Standard APIs

- **Settable Thresholds**
  - **Soft Threshold – like tape programmable early warning – Begin reporting to host (recovered error)**
  - **Hard Threshold – like tape EOM – Thin space ran out – Writes start to fail (hard error)**
  - **Mode Page readable/settable?**
- **Query state of Thin/Thick status of block:**
  - **Send LBA/allocation length – returns bit/byte map**

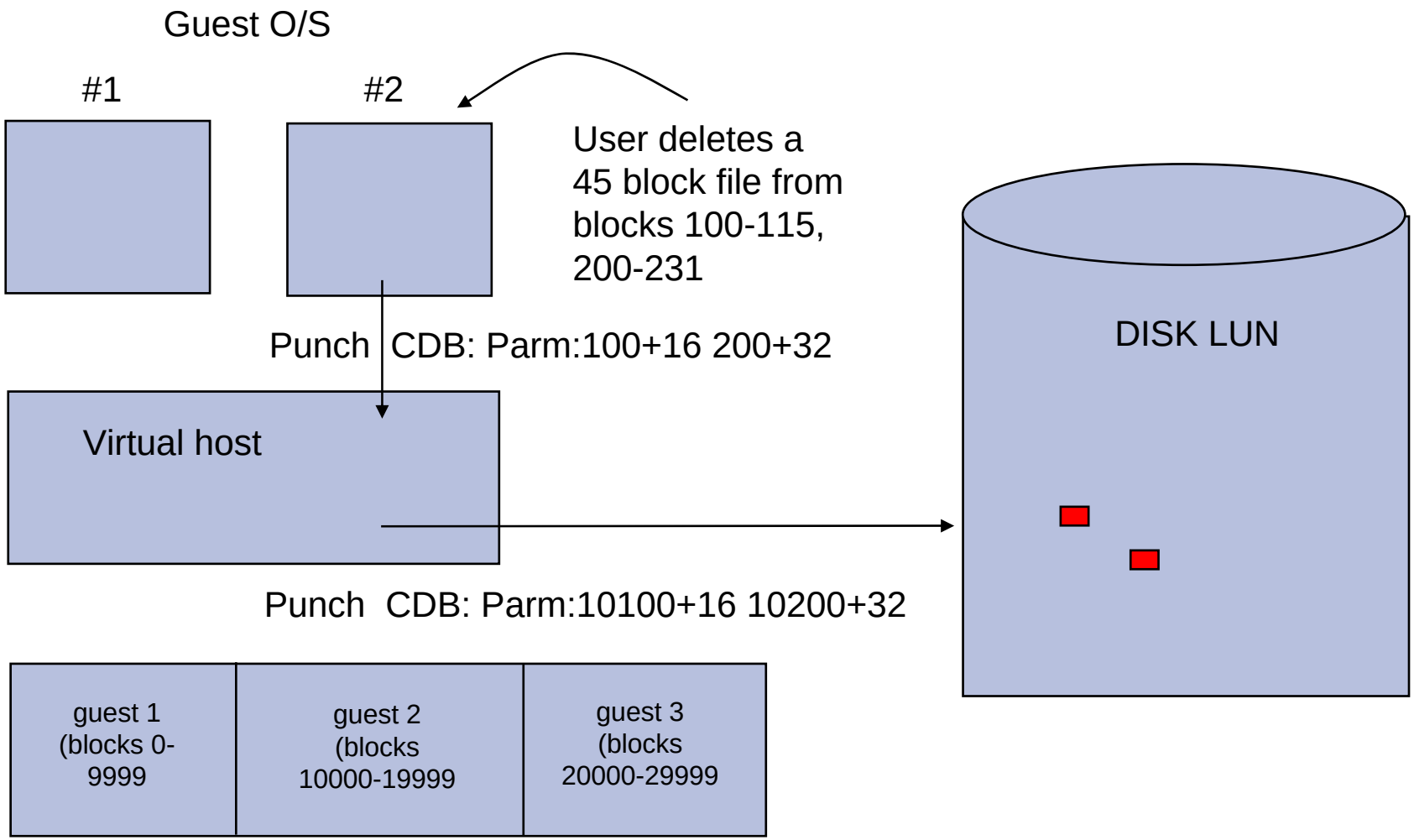
Standard APIs

- **Report Supported Operation Codes**
  - Tells host new command exists
- **Create new SCSI return codes (samples)**
  - CC: RECOVERED ERROR + SOFT CAPACITY THRESHOLD ENCOUNTERED
  - CC: RECOVERED ERROR + HARD CAPACITY THRESHOLD ENCOUNTERED
  - CC: HARDWARE ERROR/DATA PROTECT + HARD CAPACITY THRESHOLD ENCOUNTERED

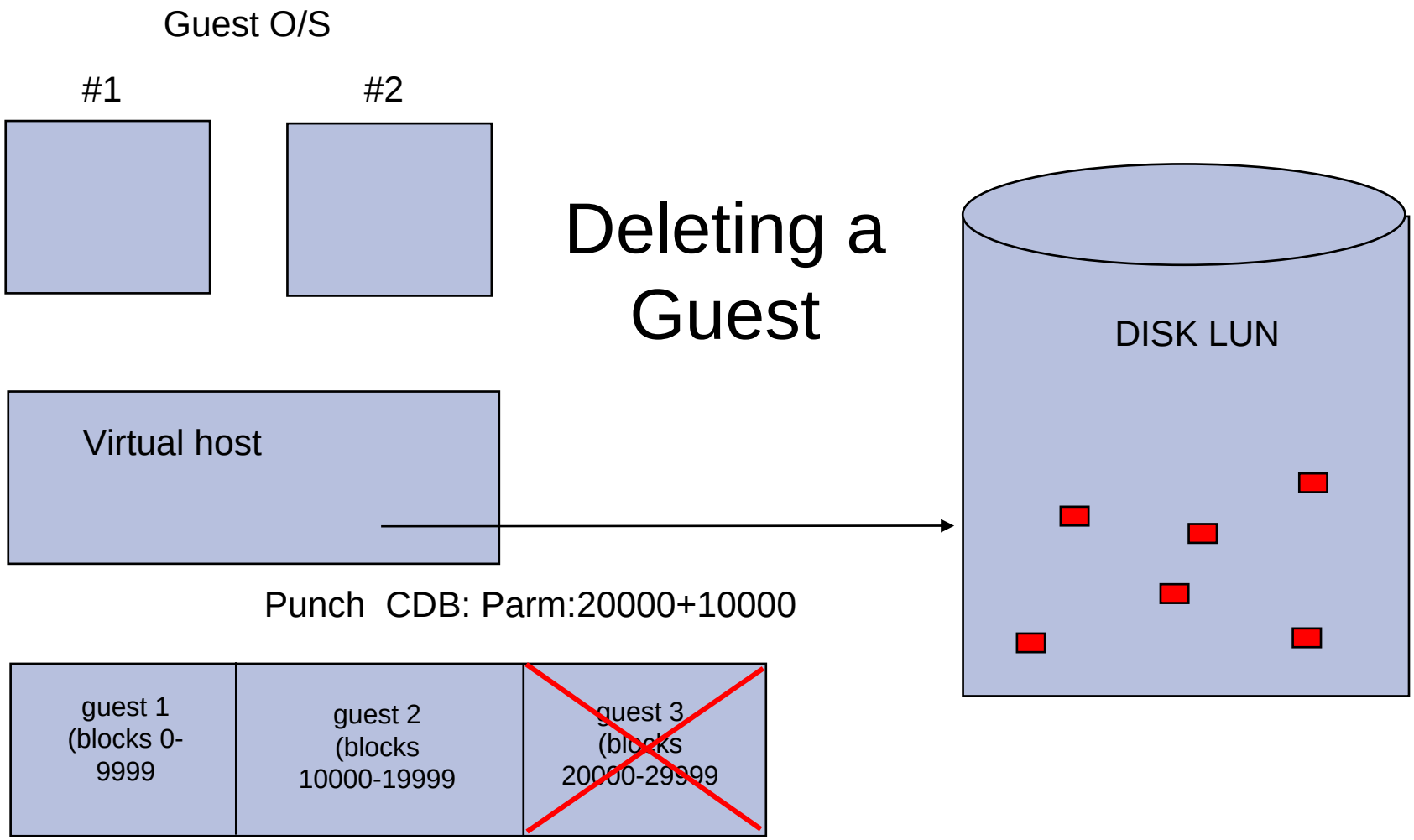
Use Cases


Guest O/S
#1
#2
User writes data to file
DISK LUN
Write CDB
Virtual host
Status (Success; Error; Recovered Error - threshold encountered)
guest 1 (blocks 0-9999
guest 2 (blocks 10000-19999
guest 3 (blocks 20000-29999

Use Cases


Guest O/S
#1
#2
User deletes a 45 block file from blocks 100-115, 200-231
Punch CDB: Parm:100+16 200+32
Virtual host
DISK LUN
Punch CDB: Parm:10100+16 10200+32
guest 1 (blocks 0-9999
guest 2 (blocks 10000-19999
guest 3 (blocks 20000-29999

Use Cases


Guest O/S
#1
#2
Deleting a Guest
DISK LUN
Virtual host
Punch CDB: Parm:20000+10000
guest 1 (blocks 0-9999
guest 2 (blocks 10000-19999
guest 3 (blocks 20000-29999

Other Uses

- **Flash Memory “disks”**
  - **Read = 20us**
  - **Write (to erased area) = 200us**
  - **Write (to non-erased area) = 1.7ms**
  - **Erase = 1.5ms**
- **Erase on delete will speed up next write.**
  - **Perfect use for Punch/Delete/Pre-Write CDB**

DISCUSSION