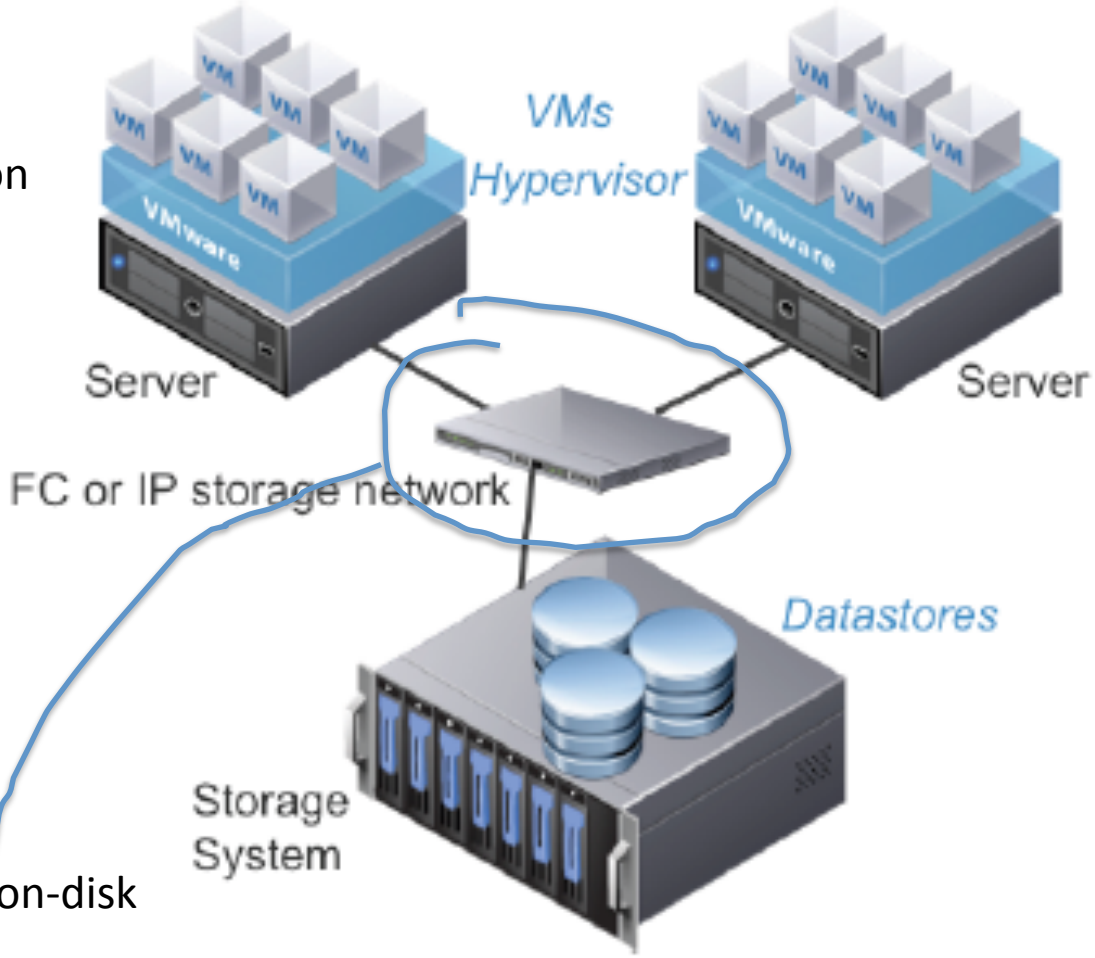
VMFS paper
by Satyam Vaghani

http://dl.acm.org/citation.cfm?id=1899935

Design goals

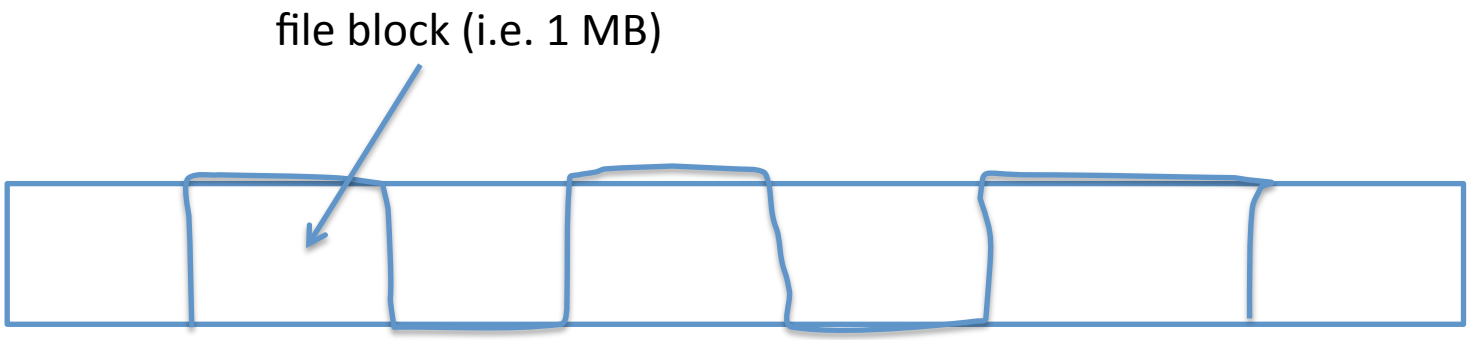
- clusters, orchestrate access to shared LUNs across nodes without them directly talking to each other.
  - Do not add things like a primary, secondary server into the picture (which were typical in clustered file systems).
- mask errors, SANs generate many errors that typical OSes are not designed to handle
- designed for small number of huge files: this is critical for FS design.

no direct communication between servers



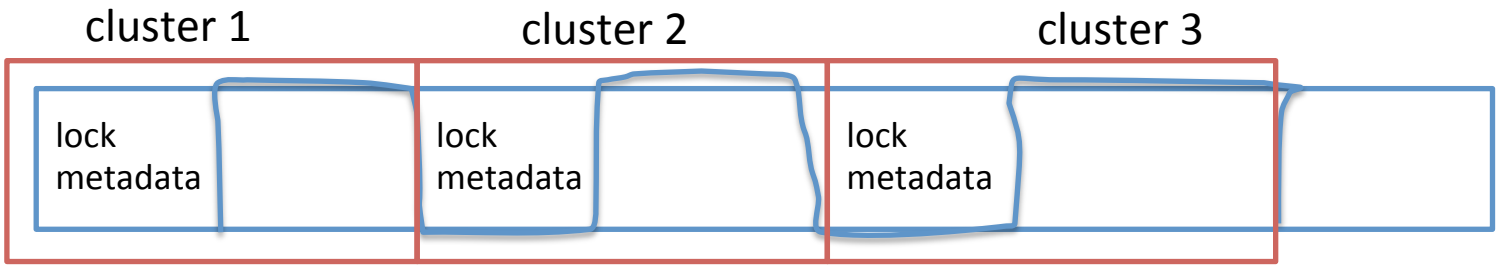
locks are on-disk

Data layout

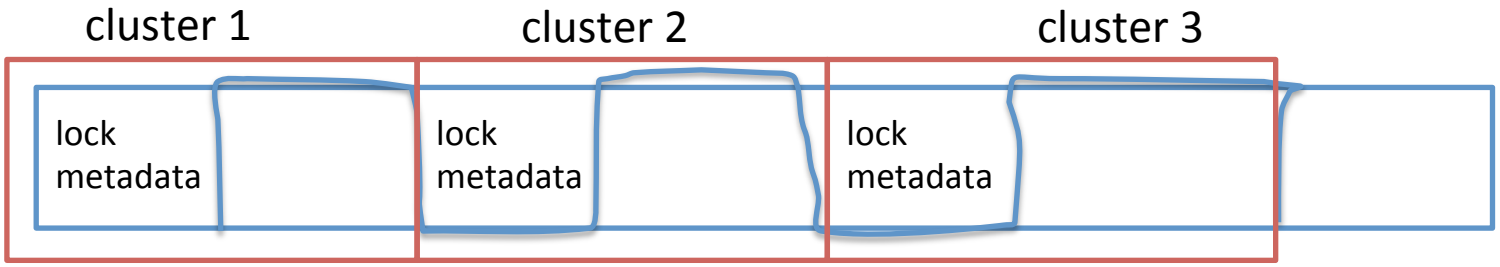


LUN exports a single linear address space
VMFS divides it in file blocks

Data layout


cluster 1
cluster 2
cluster 3
lock metadata
lock metadata
lock metadata

Size of clusters



example: allocation of 1GB

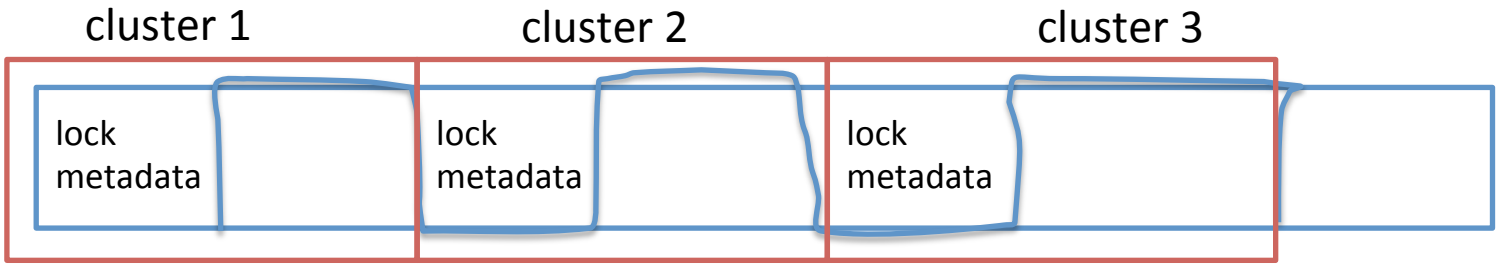
if clusters are 1MB each, then

1. acquire 1000 locks
2. less likely for different hosts to use the same cluster

if clusters are 100MB each

1. acquire 10 locks
2. more likely for different hosts to use the same cluster

Cluster allocation to hosts



try to make hosts operate in different and distant cluster groups

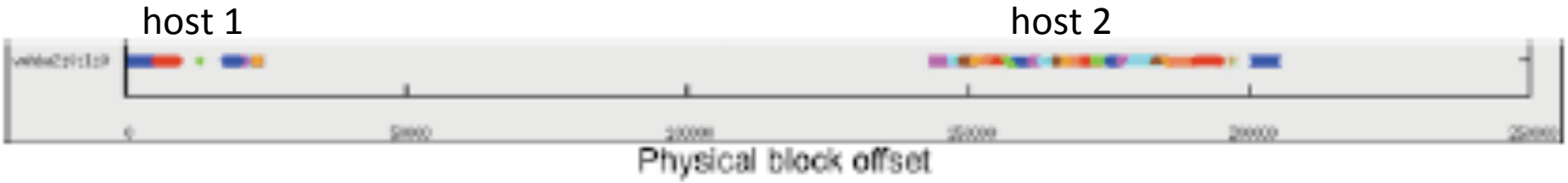
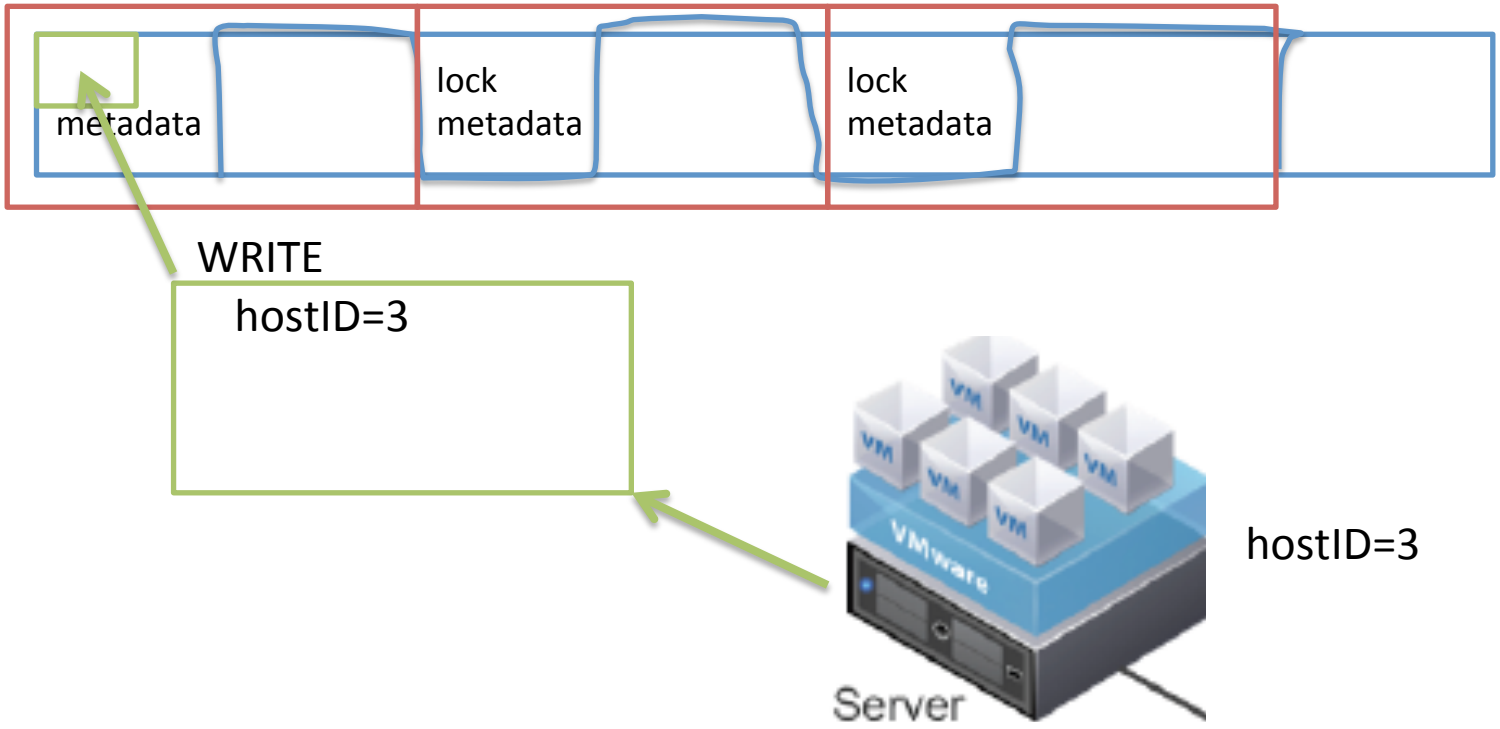


Figure 4: Physical location of files on a shared volume

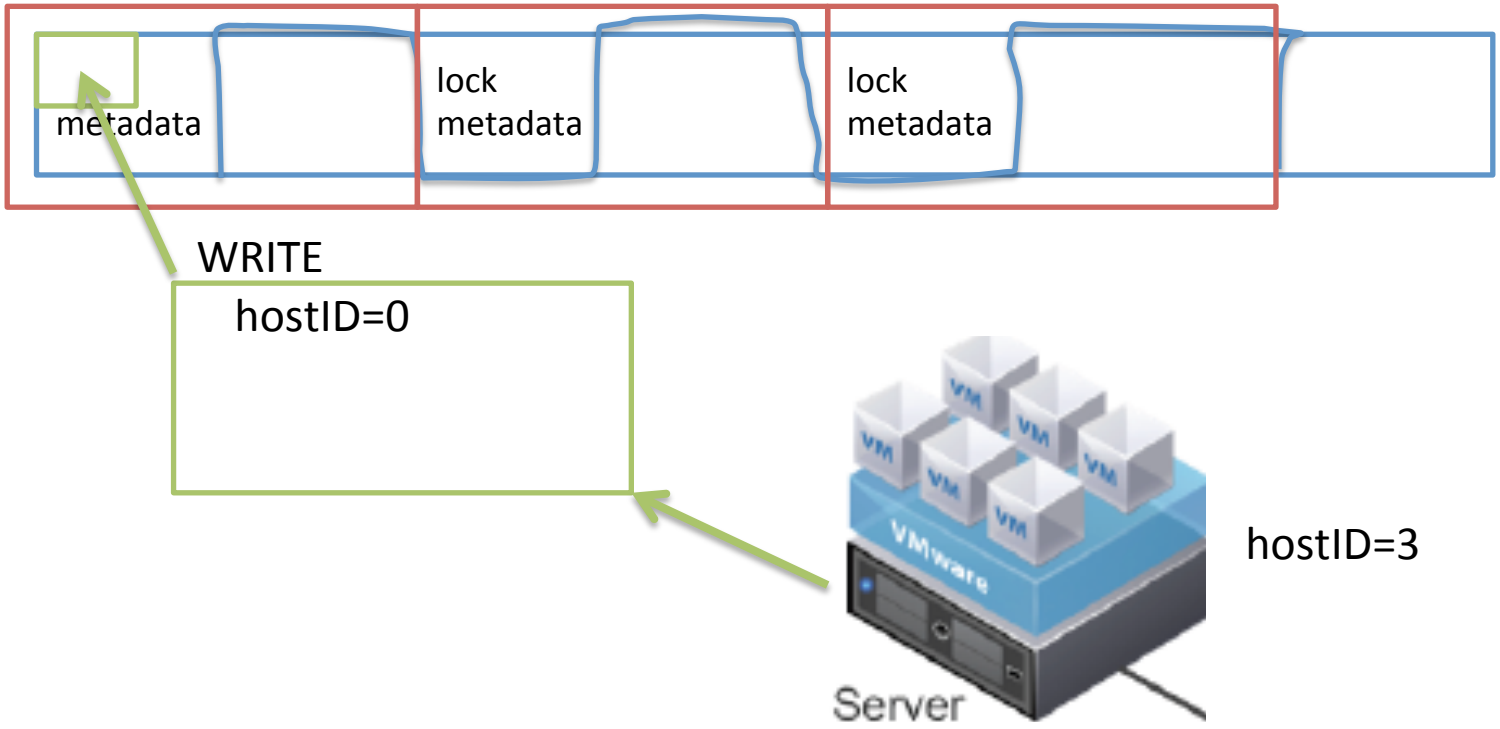
Lock acquiring

- lock acquiring is done by writing <hostID> to the related cluster lock sector



Lock releasing

- lock releasing is done by writing <0> to the related cluster lock sector



SCSI reservations

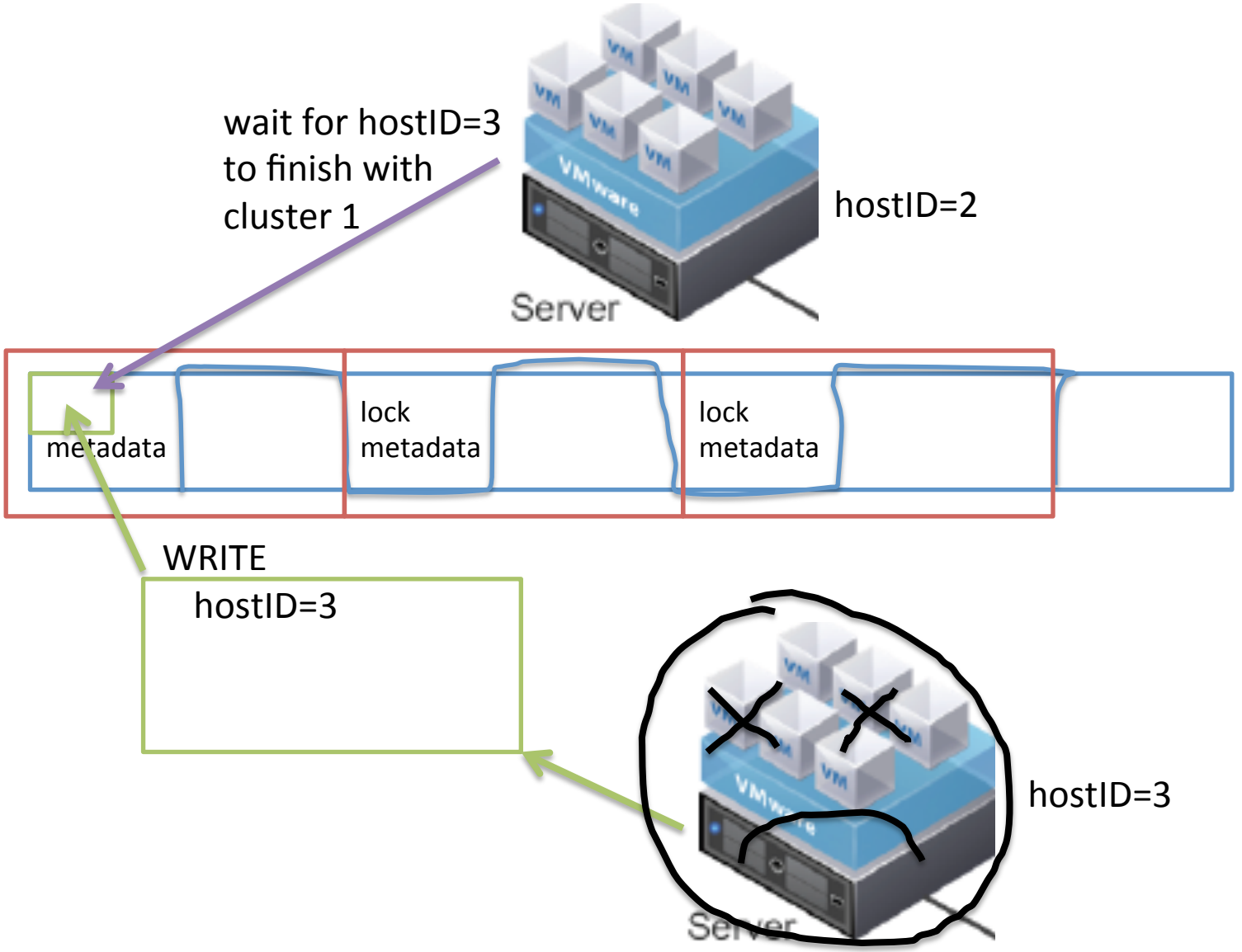
```
if cluster.lock.hostID == 0
    acquire lock
else
    try again
```

atomicTestAndSet

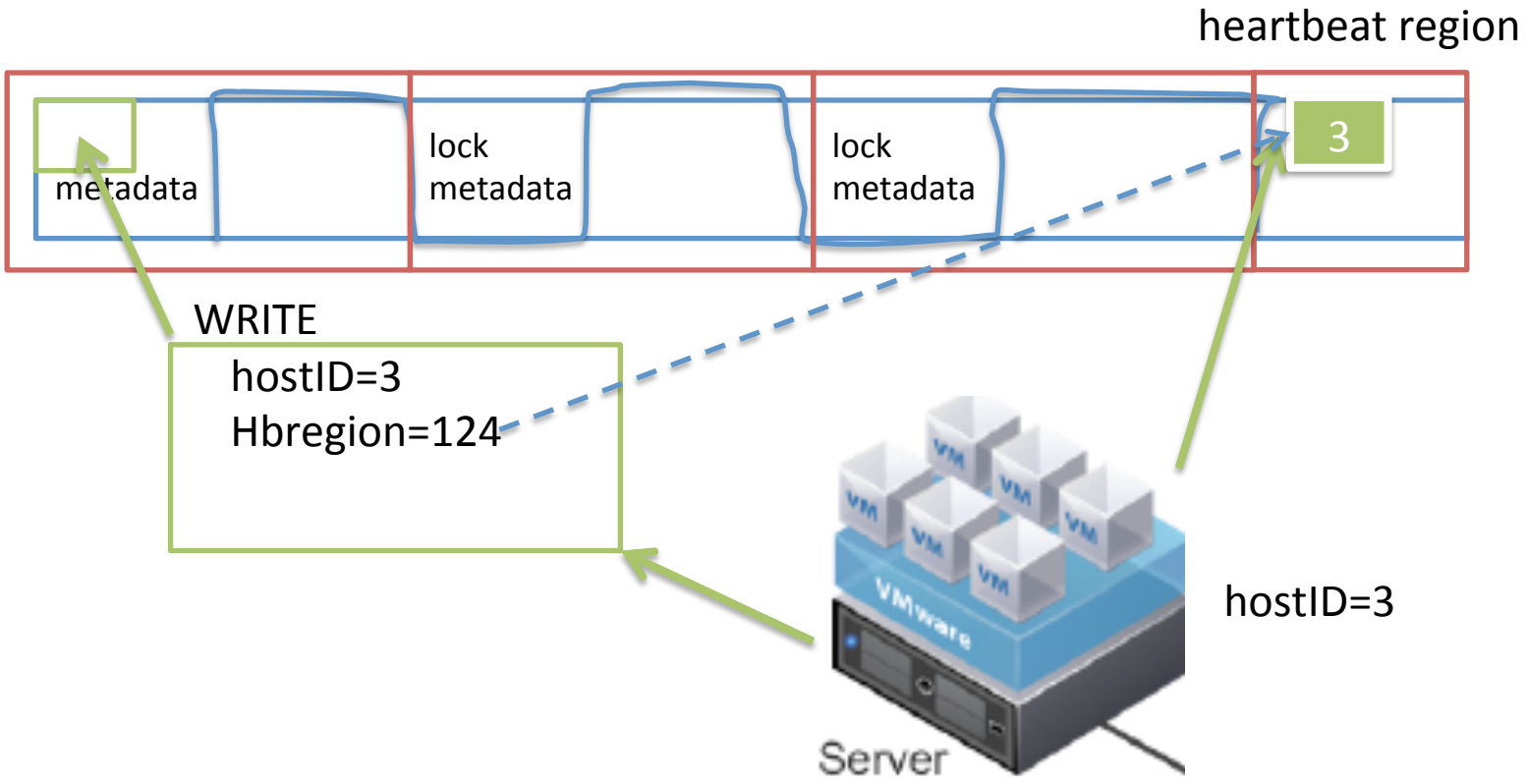
```
    SCSI reservation START command
    // critical region
    SCSI reservation END command
```

IO not allowed by any other host while in SCSI reservation

Host is dead and can’t release the lock

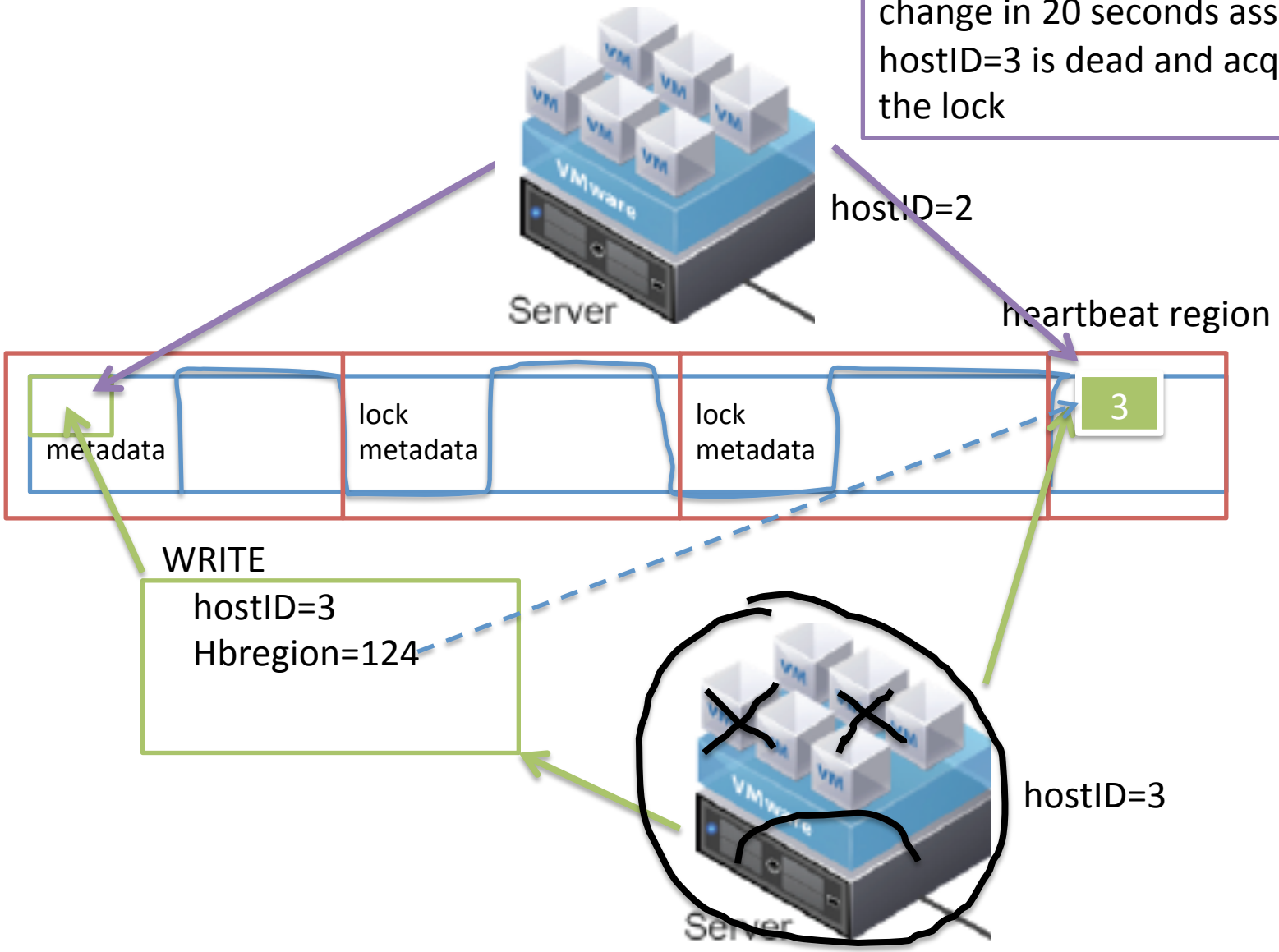


Heartbeats

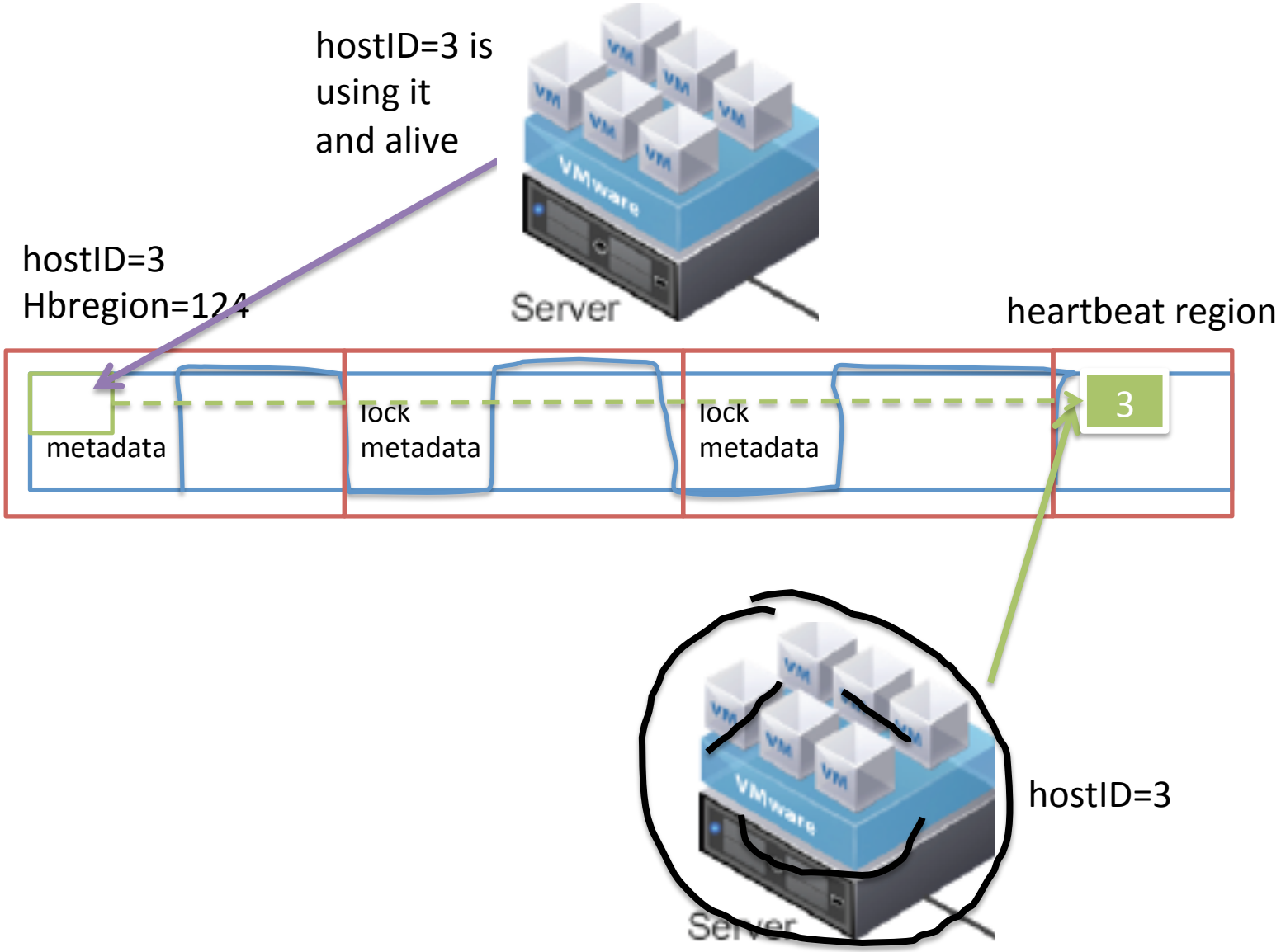

heartbeat region
metadata
lock
metadata
lock
metadata
3
WRITE
hostID=3
Hbregion=124
VM
VM
VM
VM
VM
VM
VMware
Server
hostID=3

Heartbeats

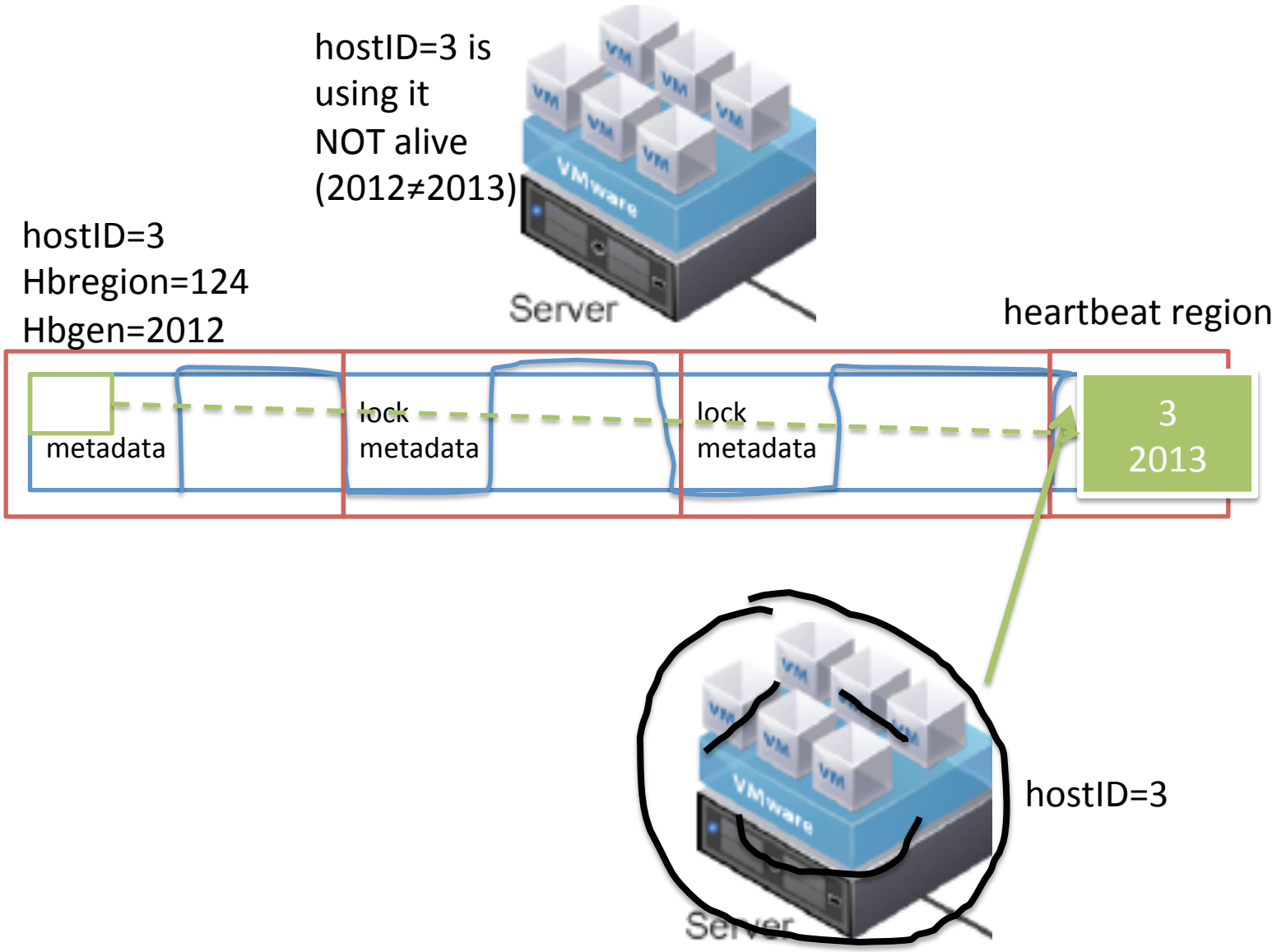
if Hbregion[124] does not change in 20 seconds assume hostID=3 is dead and acquire the lock



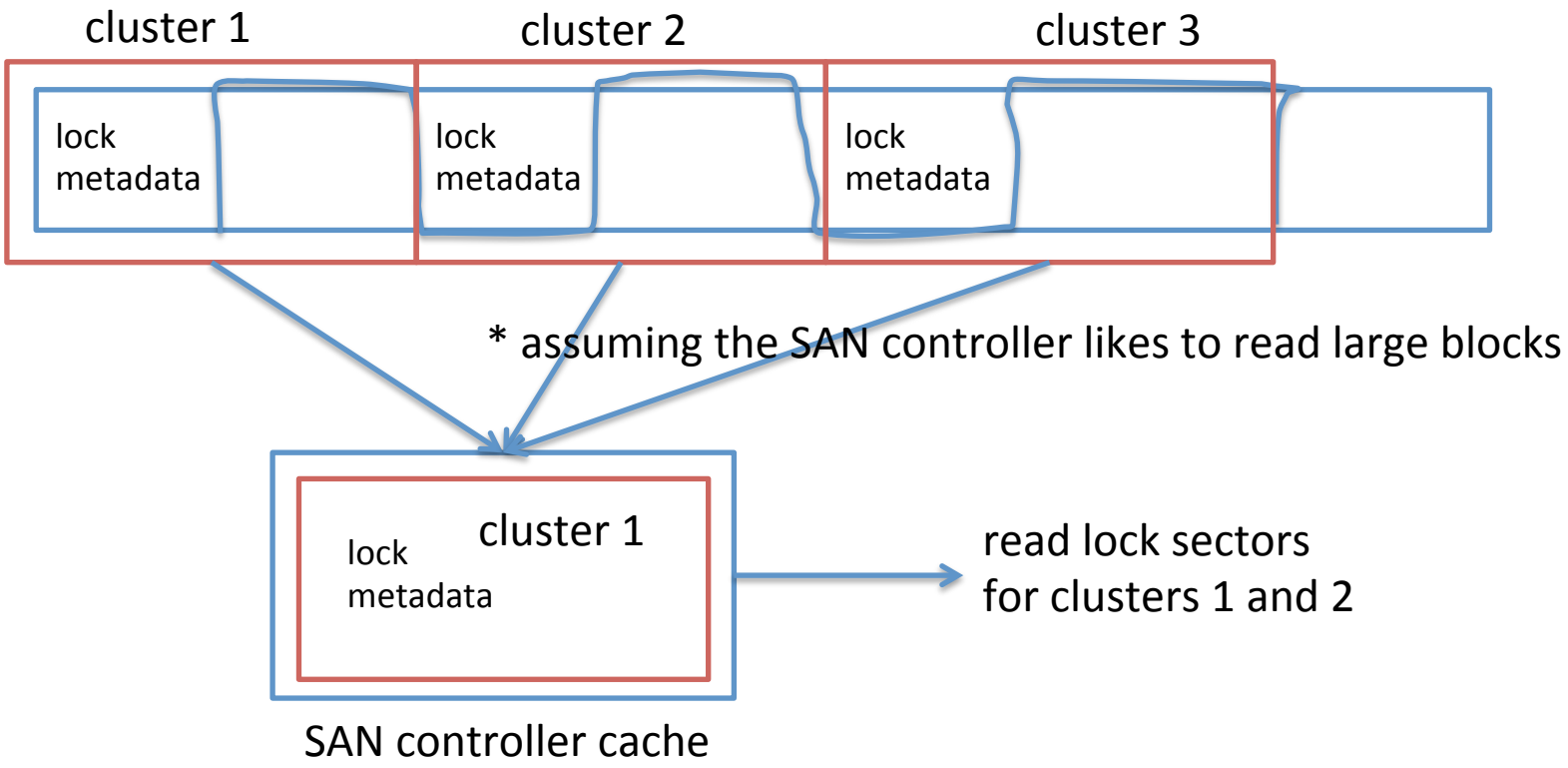
Stale locks after a crash



Generation numbers



Poor lock/metadata locality of reference



Cluster groups

Figure 3: VMFS layout on disk

Cluster groups

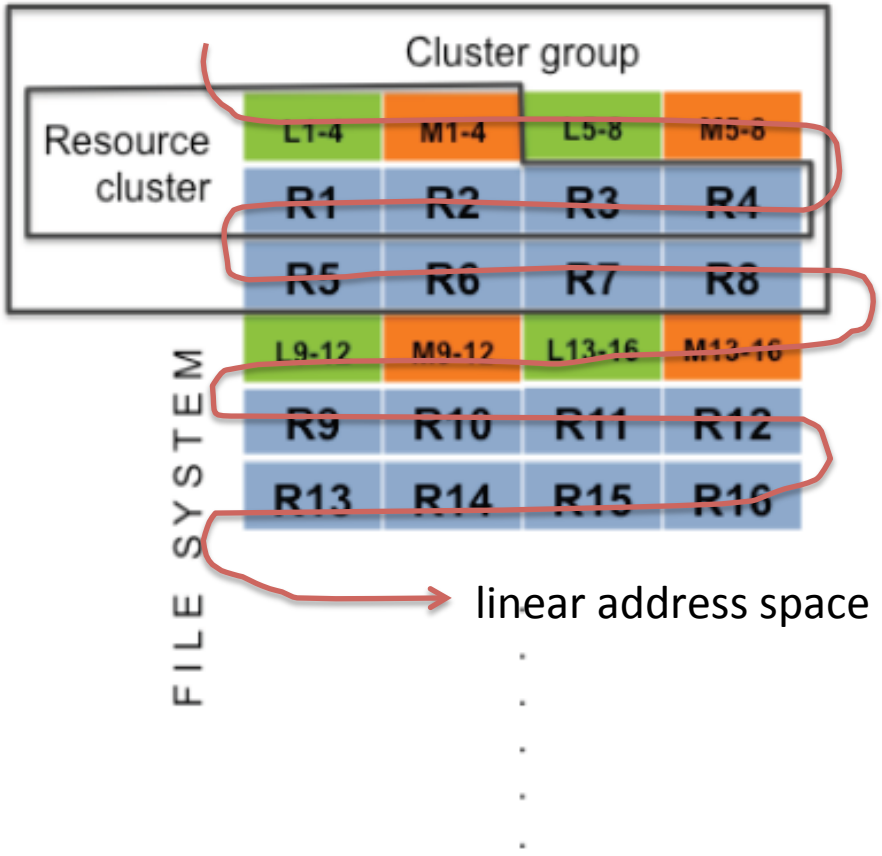


Figure 3: VMFS layout on disk

Cluster groups (locality of reference)

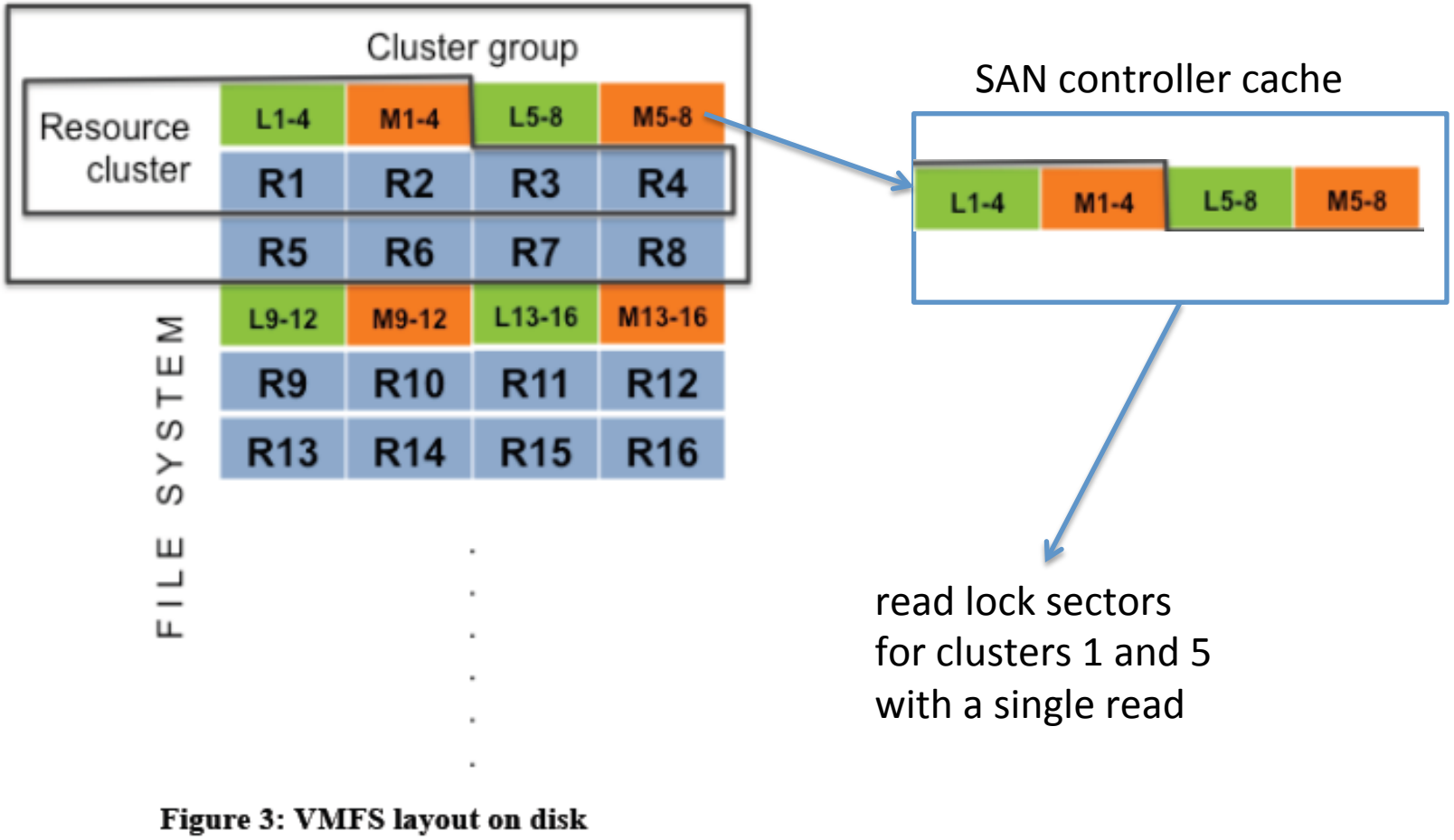


Figure 3: VMFS layout on disk

Types of resources

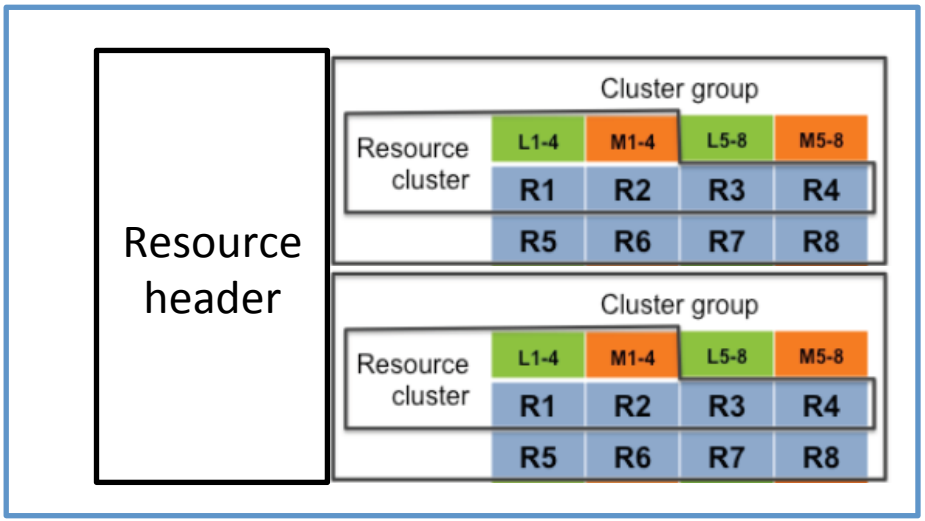
- There are 4 types of resources:
  - file block resources (ex. 1MB)
  - sub-block resources (ex. 8KB)
  - pointer-block resources (ex. 4KB)
  - file descriptor resources (ex. 2KB)

Types of resources

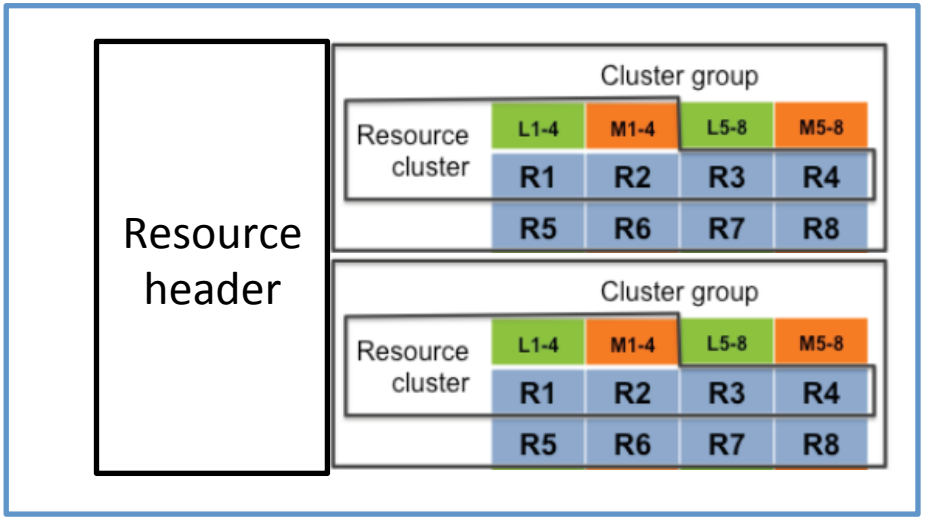
- There are 4 types of resources:
  - file block resources (ex. 1MB)
    - actual file data
  - ~~sub-block resources (ex. 8KB)~~
  - ~~pointer-block resources (ex. 4KB)~~
  - file descriptor resources (ex. 2KB)
    - directory structure

System files

file-block system file



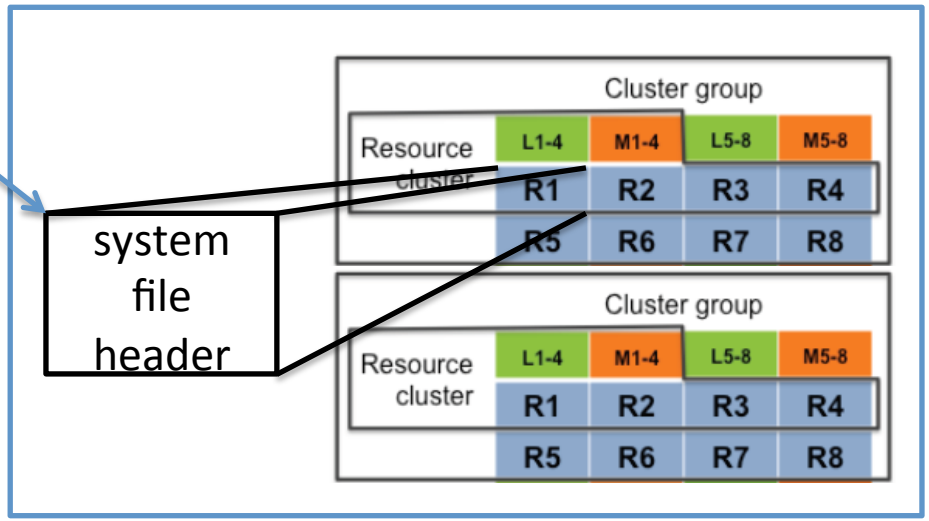
file-descriptor system file



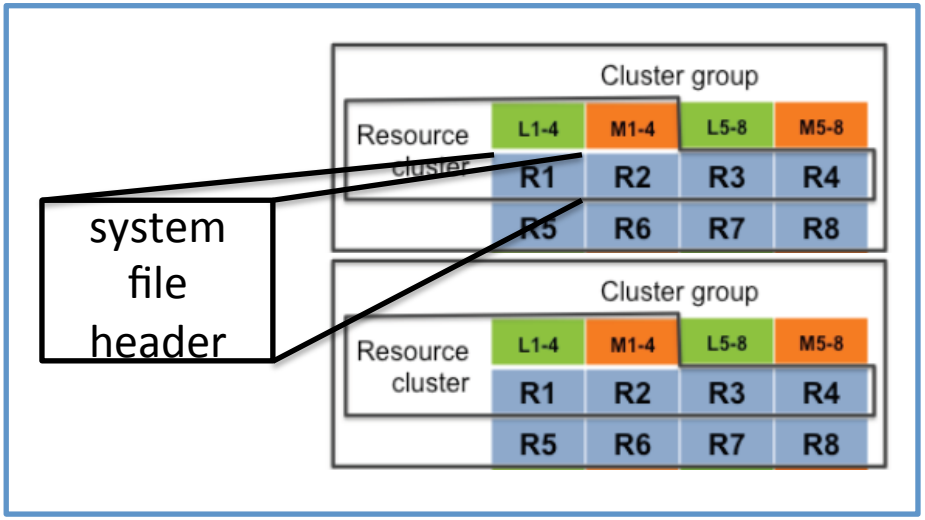
System files are actual files

* hosts boot with this offset hardcoded

file-block system file



file-descriptor system file



Too much cost involved in locking

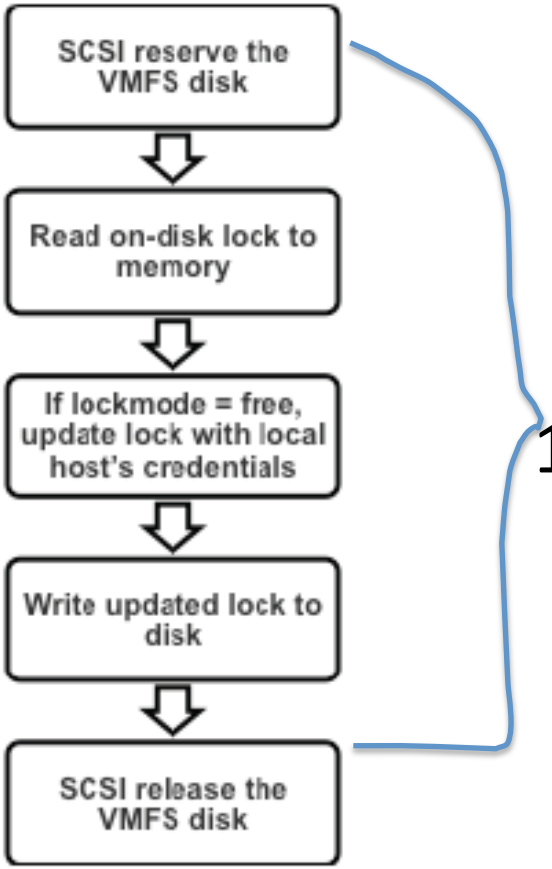


Figure 5: Algorithm to acquire a disk lock (simplified)

VMFS transactions

- VMFS rolls forward the state of the file system on a journal replay
- The journal of each host is pointed by the heartbeat region of the host

VMFS transations with regular locks

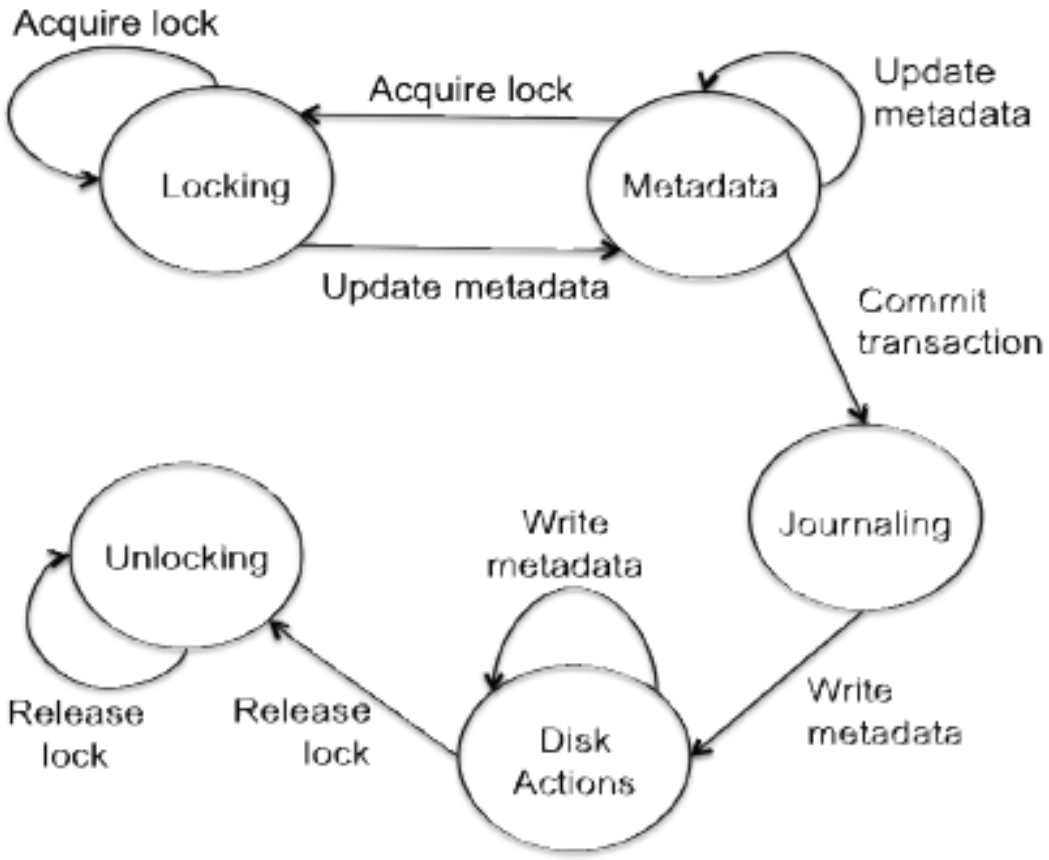


Figure 6: Transaction state machine

Optimistic locking

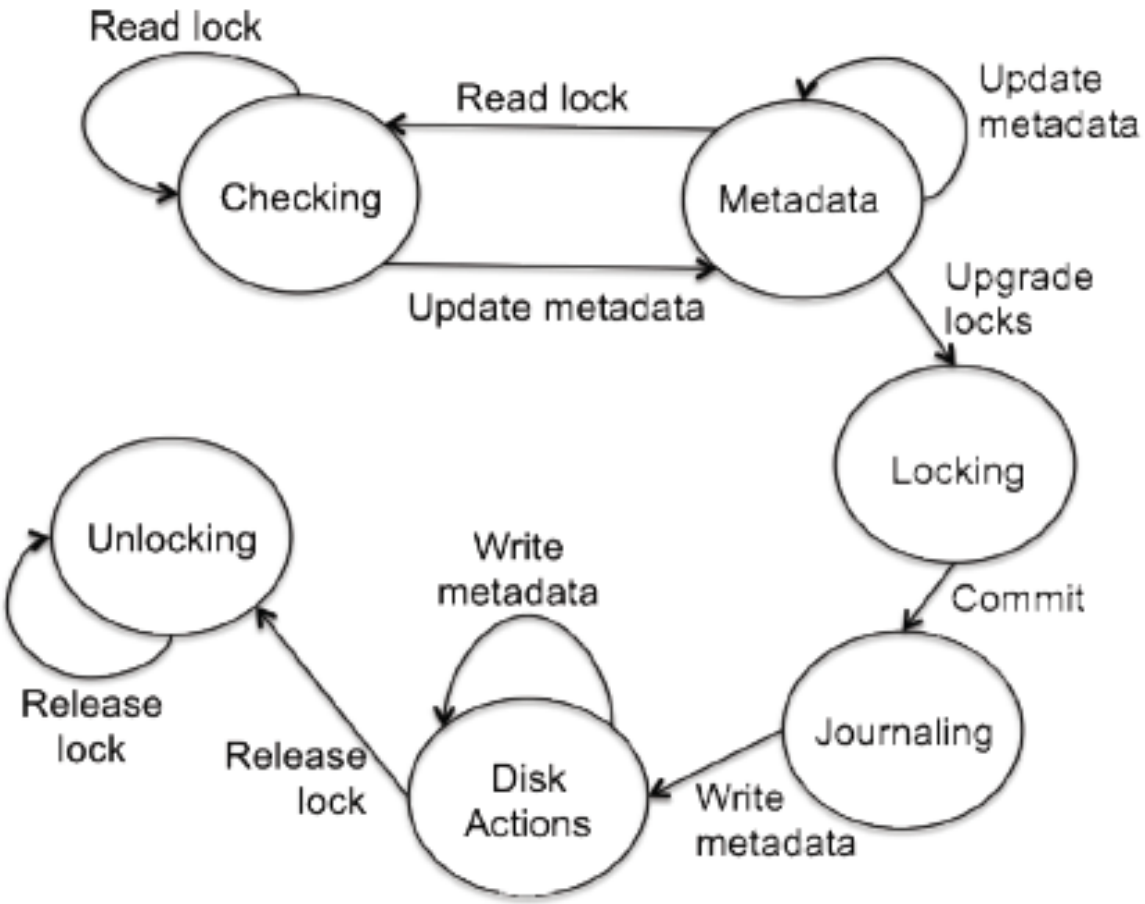
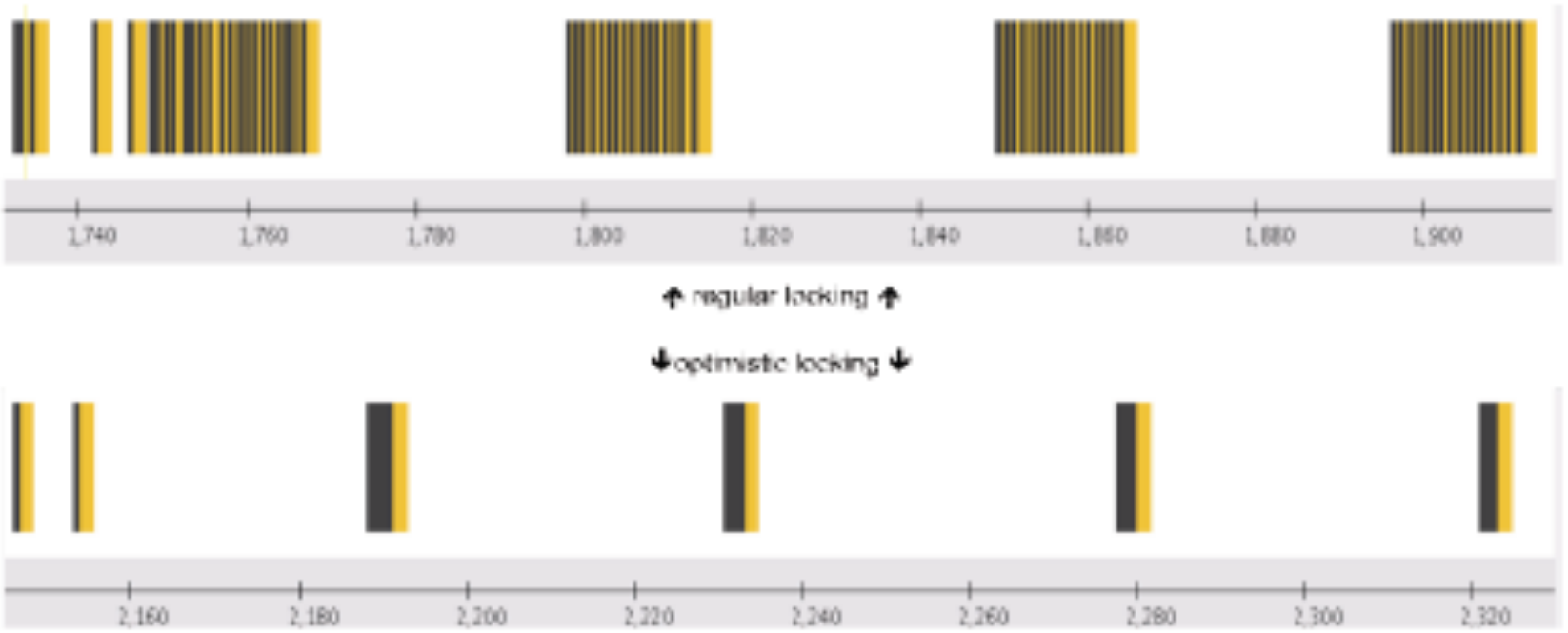


Figure 7 shows how optimistic locking changes the transaction state machine. In particular, the host no longer eagerly acquires locks to all the metadata regions it needs to read and modify in the transaction. Instead, it reads all required locks and if the locks are free, it reads and updates metadata in memory. By the time the transaction is ready to be committed to the journal, the host has built up a list of all locks that the transaction depends on. It then proceeds to acquire all locks using a single SCSI reservation as shown in figure 8 below:

Optimistic locking results



Black: reserve, yellow: R-M-W release
Width: duration of operation, #stripes: operation frequency

Data mover left for next talk