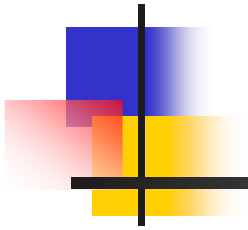
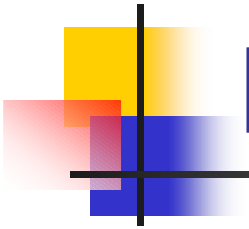


Linux Clustering & Storage Management

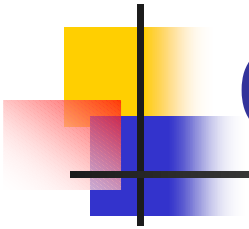
Peter J. Braam

CMU, Stelias Computing, Red Hat



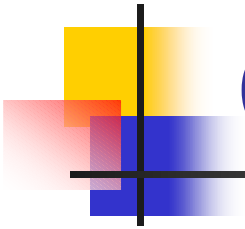
Disclaimer

- Several people are involved:
  - Stephen Tweedie (Red Hat)
  - Michael Callahan (Stelias)
  - Larry McVoy (BitMover)
- Much of it is not new –
  - Digital had it all and documented it!
  - IBM/SGI ... have similar stuff (no docs)

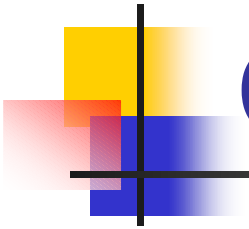


Content

- What is this cluster fuzz about?
- Linux cluster design
- Distributed lock manager
- Linux cluster file systems
- Lustre: the OBSD cluster file system

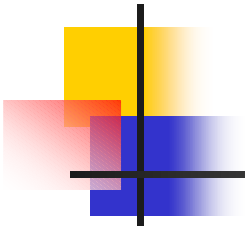


Cluster Fuz



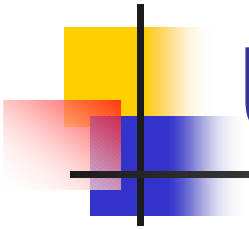
Clusters - purpose

- Assume:
  - Have a limited number of systems
  - On a secure System Area Network
- Require:
  - A scalable almost single system image
  - Fail-over capability
  - Load-balanced redundant services
  - Smooth administration



Precursors – ad hoc solutions

- WWW:
  - Piranha, TurboCluster, Eddie, Understudy:
  - 2 node group membership
  - Fail-over http services
- Database:
  - Oracle Parallel Server
- File service
  - Coda, InterMezzo, IntelliMirror



Ultimate Goal

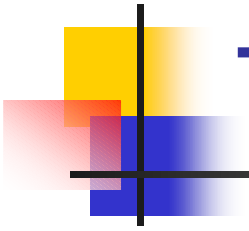
- Do this with generic components
- OPEN SOURCE
- Inspiration: VMS VAX Clusters
- New:
  - Scalable (100,000's nodes)
  - Modular

The Linux "Cluster Cabal":

- **Peter J. Braam –** CMU, Stelias Computing, Red Hat (?)
- **Michael Callahan –** Stelias Computing, PolyServe
- **Larry McVoy –** BitMover
- **Stephen Tweedie –** Red Hat

- **Who is doing what?**

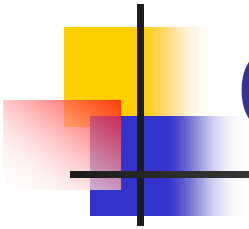
- Tweedie
  - Project leader
  - Core cluster services
- Braam
  - DLM
  - InterMezzo FS
  - Lustre Cluster FS
- McVoy
  - Cluster computing
  - SMP clusters
- Callahan
  - Varia
- Red Hat
  - Cluster apps & admin
- UMN
  - GFS: Shared block FS



Technology Overview

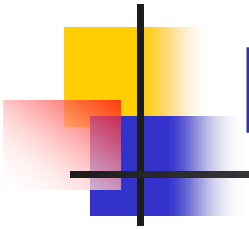
Modularized VAX cluster architecture (Tweedie)

Core	Support	Clients
Transition	Cluster db	Distr. Computing
Integrity	Quorum	Cluster Admin/Apps
Link Layer	Barrier Svc	Cluster FS & LVM
Channel Layer	Event system	DLM



Components

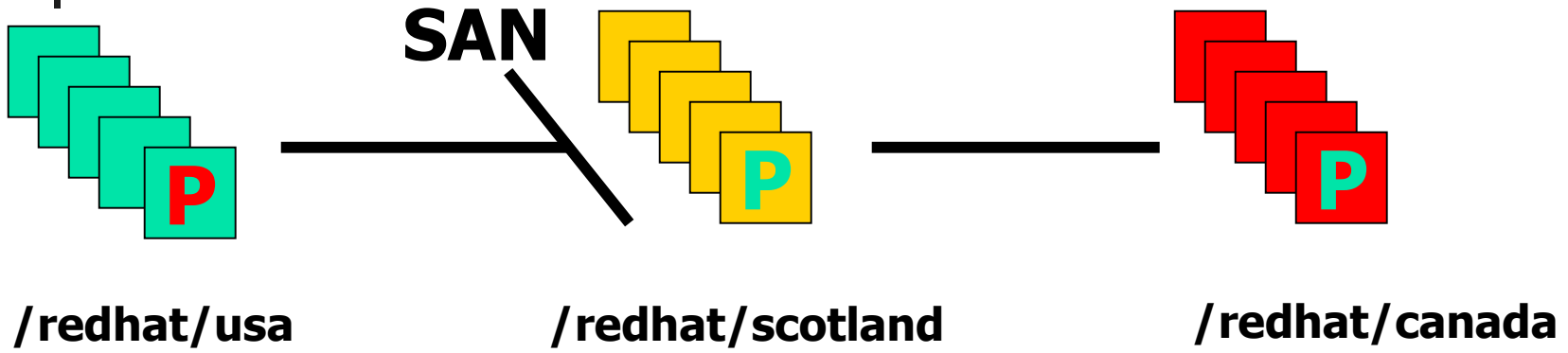
- Channel layer - comms: eth, infiniband
- Link layer - state of the channels
- Integration layer - forms cluster topology
- CDB - persistent cluster internal state (e.g. sysid)
- Transition layer - recovery and controlled startup
- Quorum - who has enoug votes?



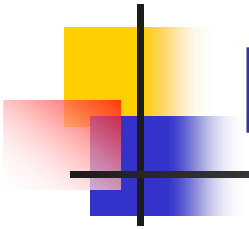
Events

- Cluster transition:
  - Whenever connectivity changes
  - Start by electing “cluster controller”
- Only merge fully connected sub-clusters
- Cluster id: counts “incarnations”
- Barriers:
  - Distributed synchronization points

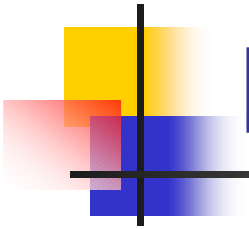
Scalability – e.g. Red Hat cluster



- **P = peer**
  - Proxy for remote core cluster
  - Involved in recovery
- **Communication**
  - Point to point within core clusters
  - Routable within cluster
  - Hierarchical flood fill
- **File Service**
  - Cluster FS within cluster
  - Clustered Samba/Coda etc
- **Other stuff**
  - Membership / recovery
  - DLM / barrier service
  - Cluster admin tools



Distributed Lock Manager

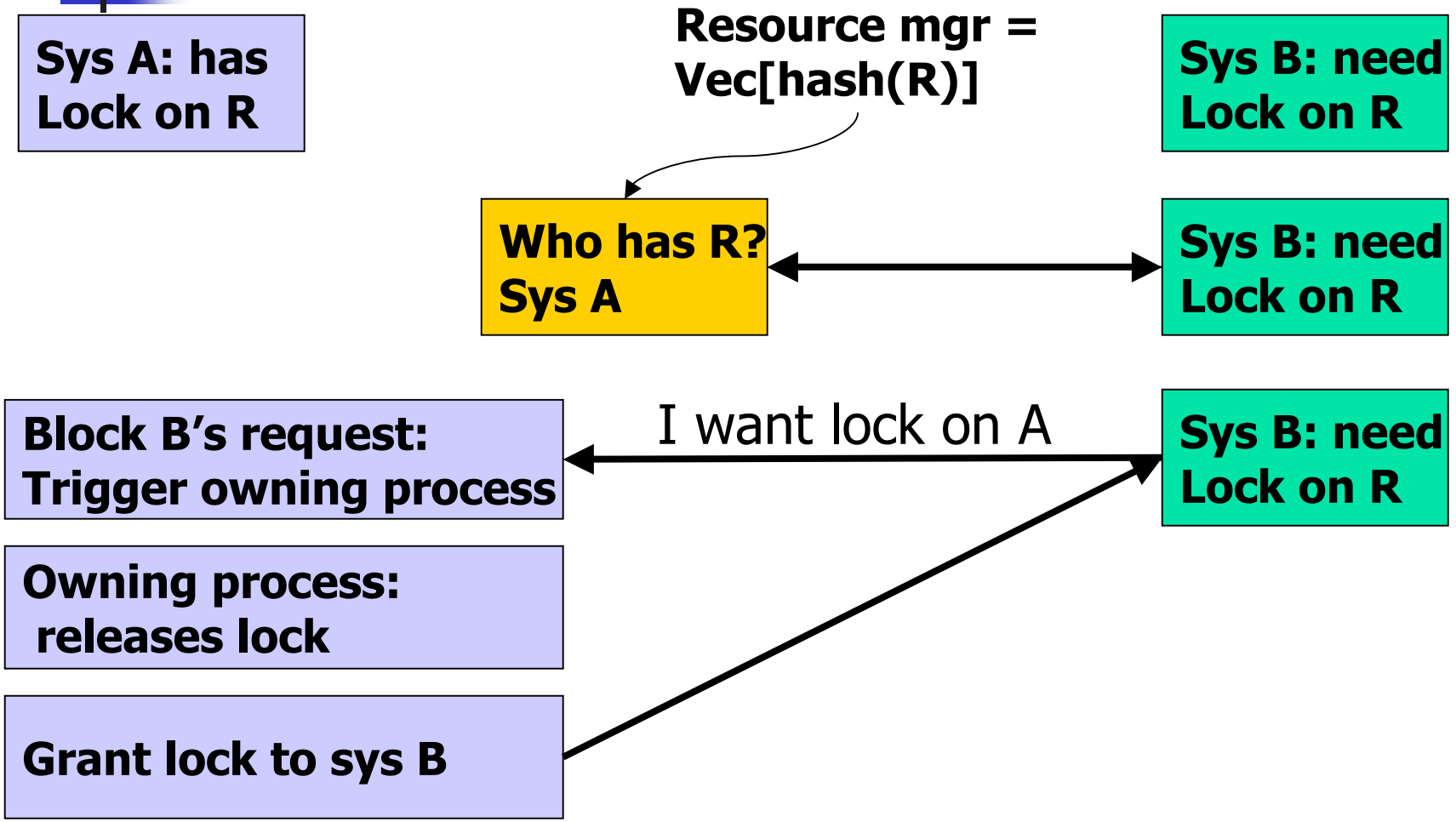


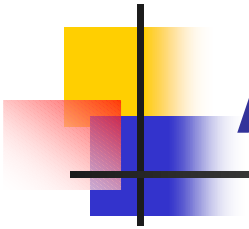
Locks & resources

- Purpose: generic, rich lock service
- Will subsume “callbacks”, “leases” etc.

- Lock resources: resource database
  - Organize resources in trees
- High performance
  - node that acquires resource manages tree

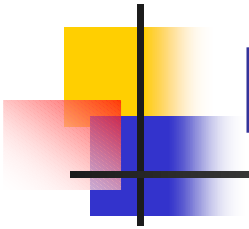
Typical simple lock sequence





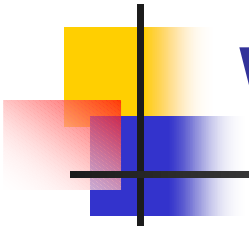
A few details…

- Six lock modes
  - Acquisition of locks
  - Promotion of locks
  - Compatibility of locks
- First lock acquisition
  - Holder will manage resource tree
- Remotely managed
  - Keep copy at owner
- Notifications:
  - On blocked requests
  - On release
- Recovery (simplified):
  - Dead node was:
    - Mastering resources
    - Owning locks
  - Re-master rsrc
  - Drop zombie locks



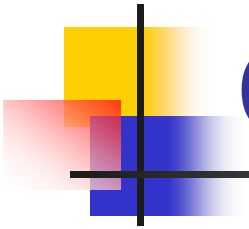
Lustre file system

- Based on object storage
- Exploits cluster infrastructure and DLM
- Cluster wide Unix semantics



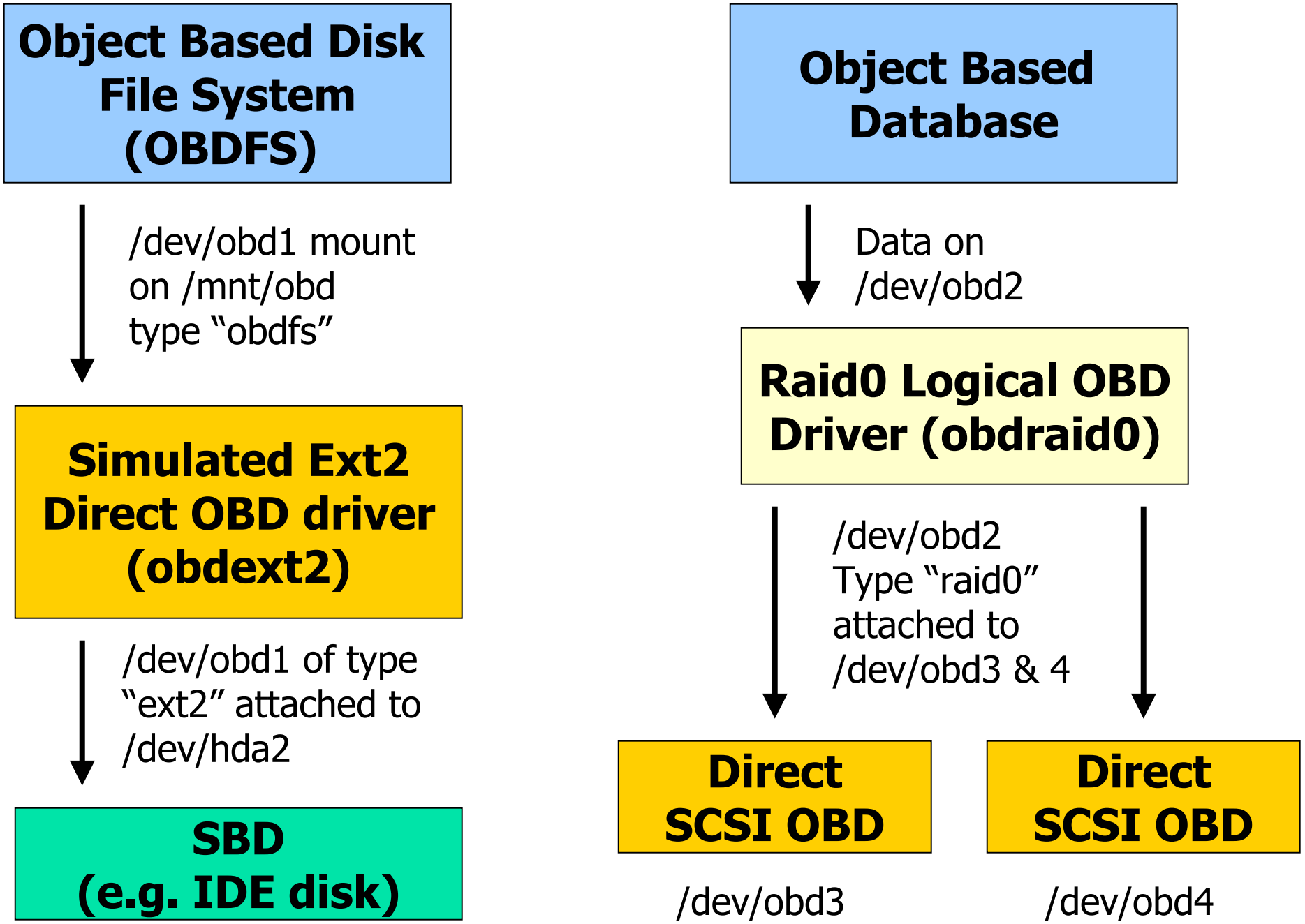
What Is an OBSD ?

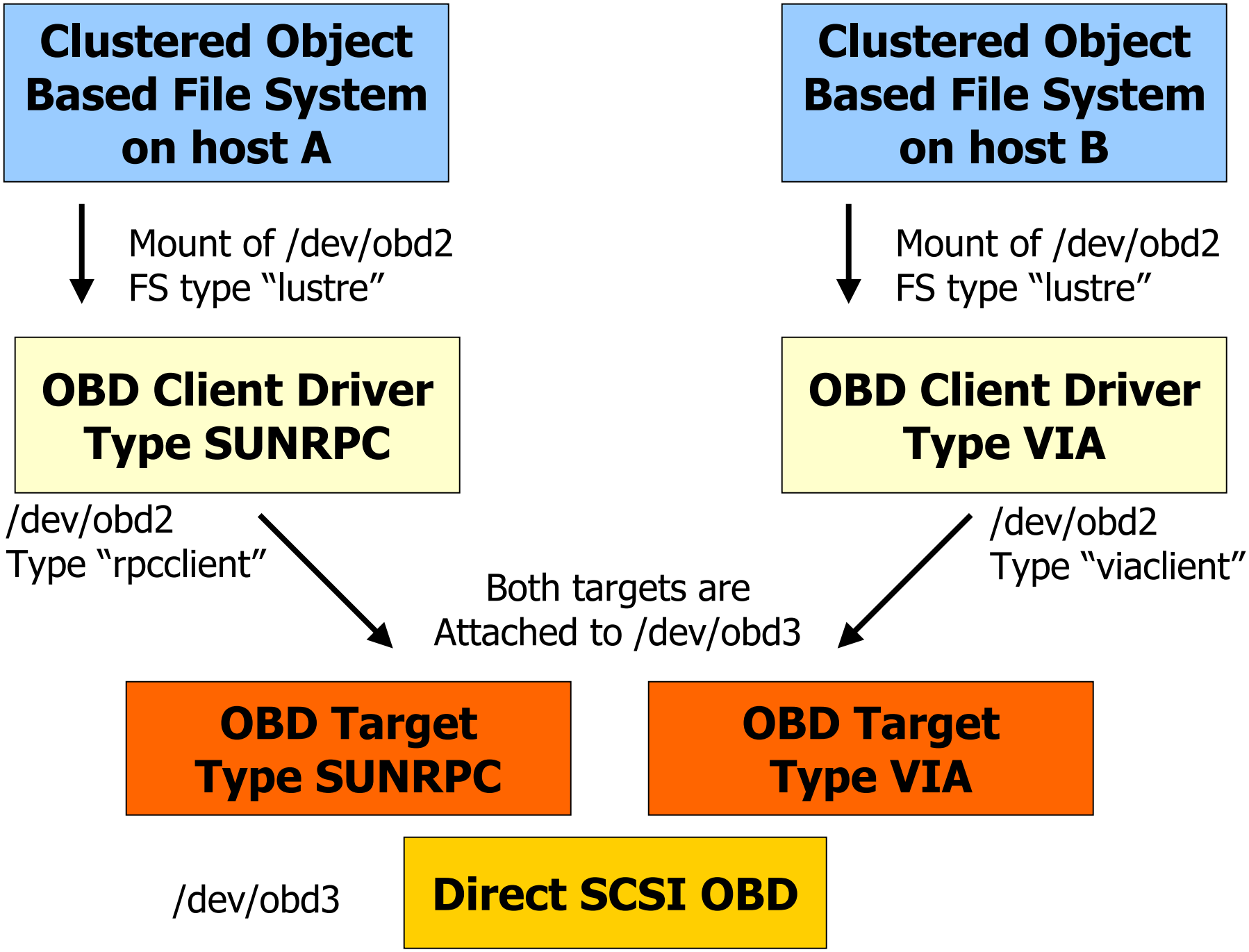
- Object Based Storage Device
  - More intelligent than block device
- Speak storage at “inode level”
  - create, unlink, read, write, getattr, setattr
- Variety of OBSD types:
  - PDL style OBD’s – not rich enough for Lustre
  - Simulated, e.g. in Linux: lower half of an fs
  - “Real obds” – ask disk vendors



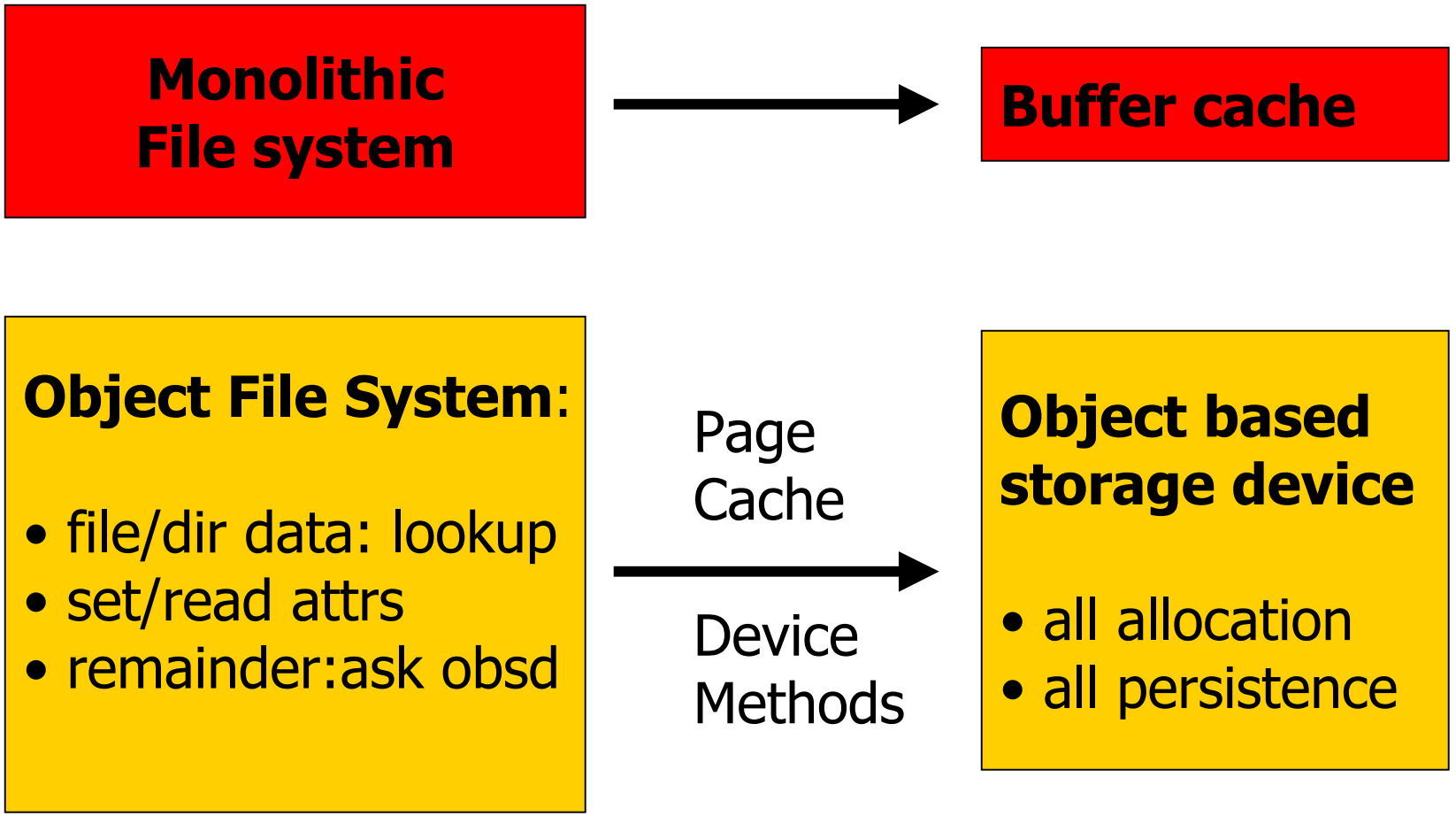
Components of OB Storage

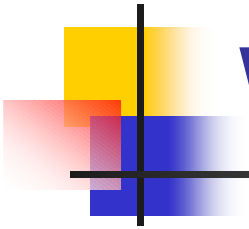
- Storage Object Device Drivers
  - class drivers – attach driver to interface
    - **Targets, clients** – remote access
    - **Direct drivers** – to manage physical storage
    - **Logical drivers** – for storage management
- object storage applications:
  - Object (cluster) file system: blockless
  - Specialized apps: caches, db's, filesrv


Object Based Disk File System (OBDFS)
/dev/obd1 mount on /mnt/obd type "obdfs"
Simulated Ext2 Direct OBD driver (obdext2)
/dev/obd1 of type "ext2" attached to /dev/hda2
SBD (e.g. IDE disk)
Object Based Database
Data on /dev/obd2
Raid0 Logical OBD Driver (obdraid0)
/dev/obd2 Type "raid0" attached to /dev/obd3 & 4
Direct SCSI OBD
Direct SCSI OBD
/dev/obd3
/dev/obd4


Clustered Object Based File System on host A
Clustered Object Based File System on host B
Mount of /dev/obd2
FS type "lustre"
Mount of /dev/obd2
FS type "lustre"
OBD Client Driver Type SUNRPC
OBD Client Driver Type VIA
/dev/obd2
Type "rpcclient"
/dev/obd2
Type "viaclient"
Both targets are
Attached to /dev/obd3
OBD Target Type SUNRPC
OBD Target Type VIA
/dev/obd3
Direct SCSI OBD

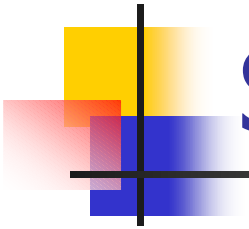
OBDFS


Monolithic
File system
Buffer cache
Object File System:
• file/dir data: lookup
• set/read attrs
• remainder:ask obsd
Page
Cache
Device
Methods
Object based
storage device
• all allocation
• all persistence



Why This Is Better…

- Clustering
- Storage management

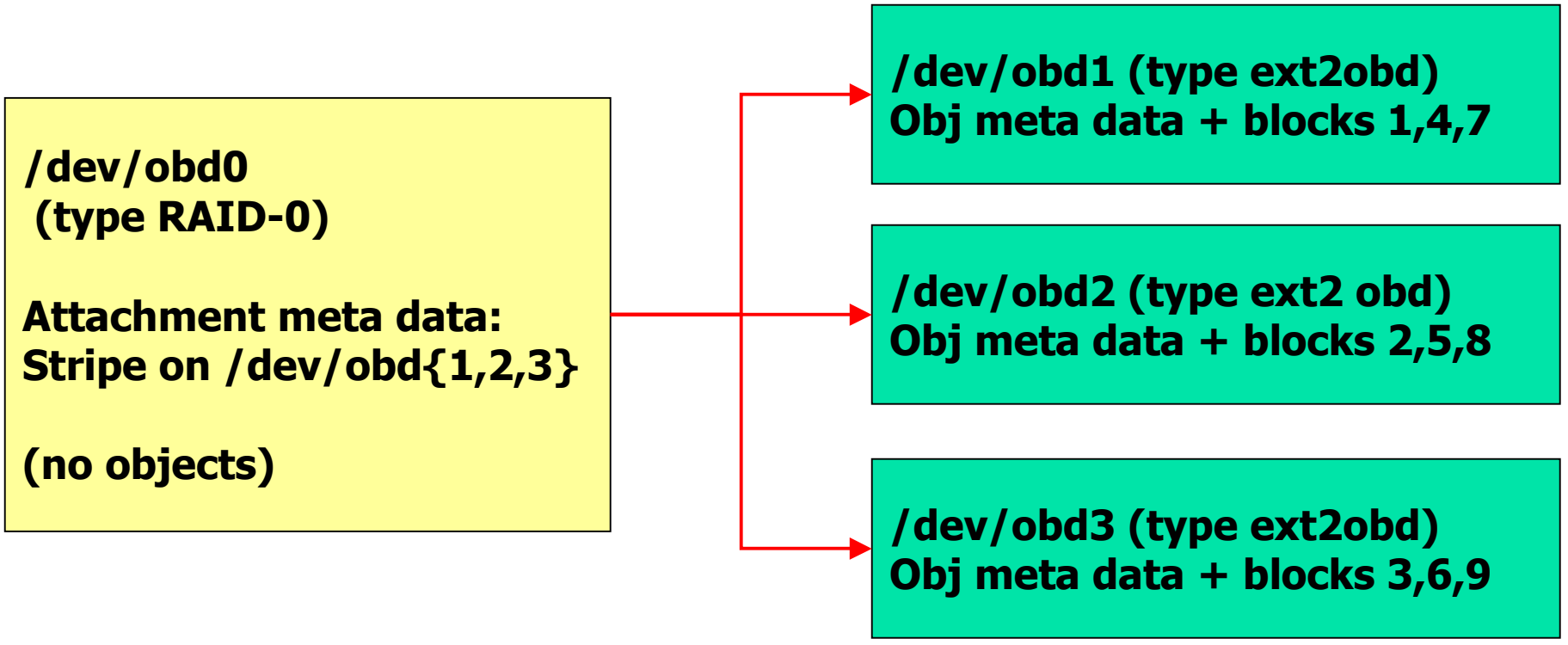


Storage Management

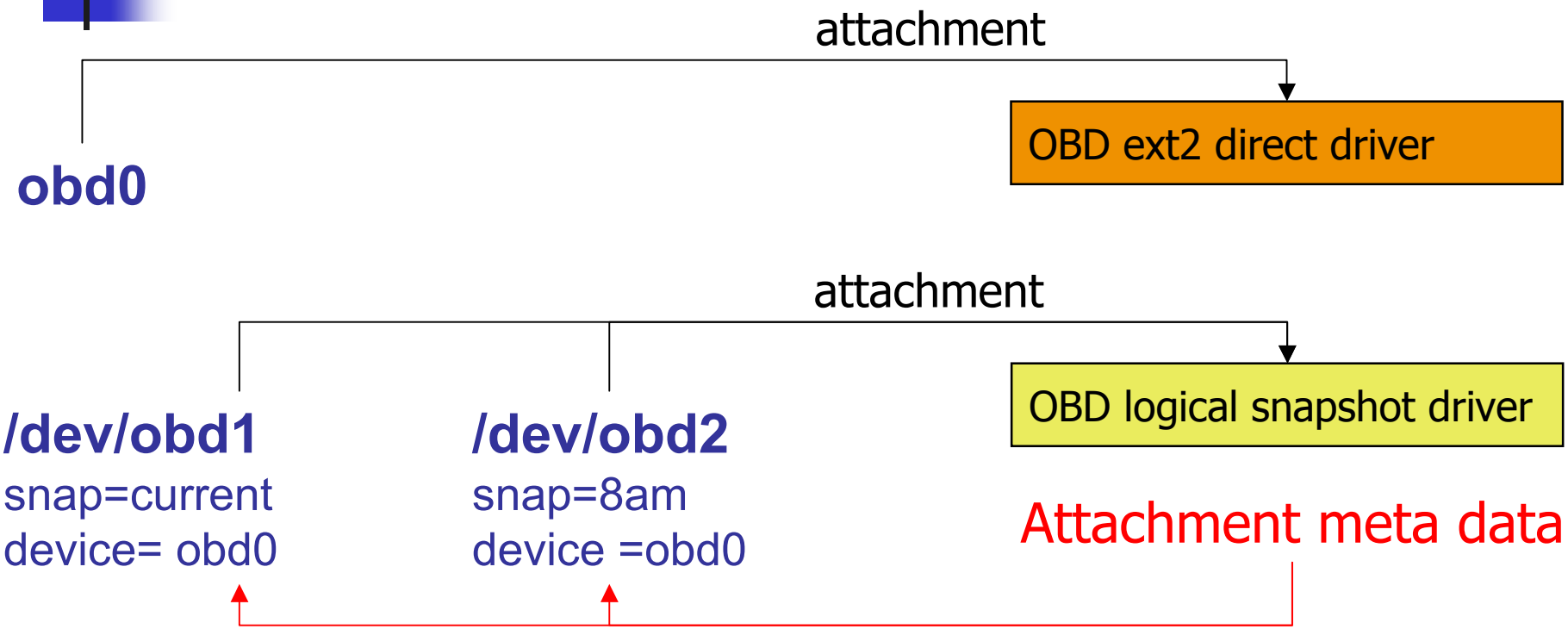
- Many problems become easier:
  - File system snapshots
  - Hot file migration
  - Hot resizing
  - Raid
  - Backup

LOVM: can do it all - Raid

Logical Object Volume Management:



Snapshot setup



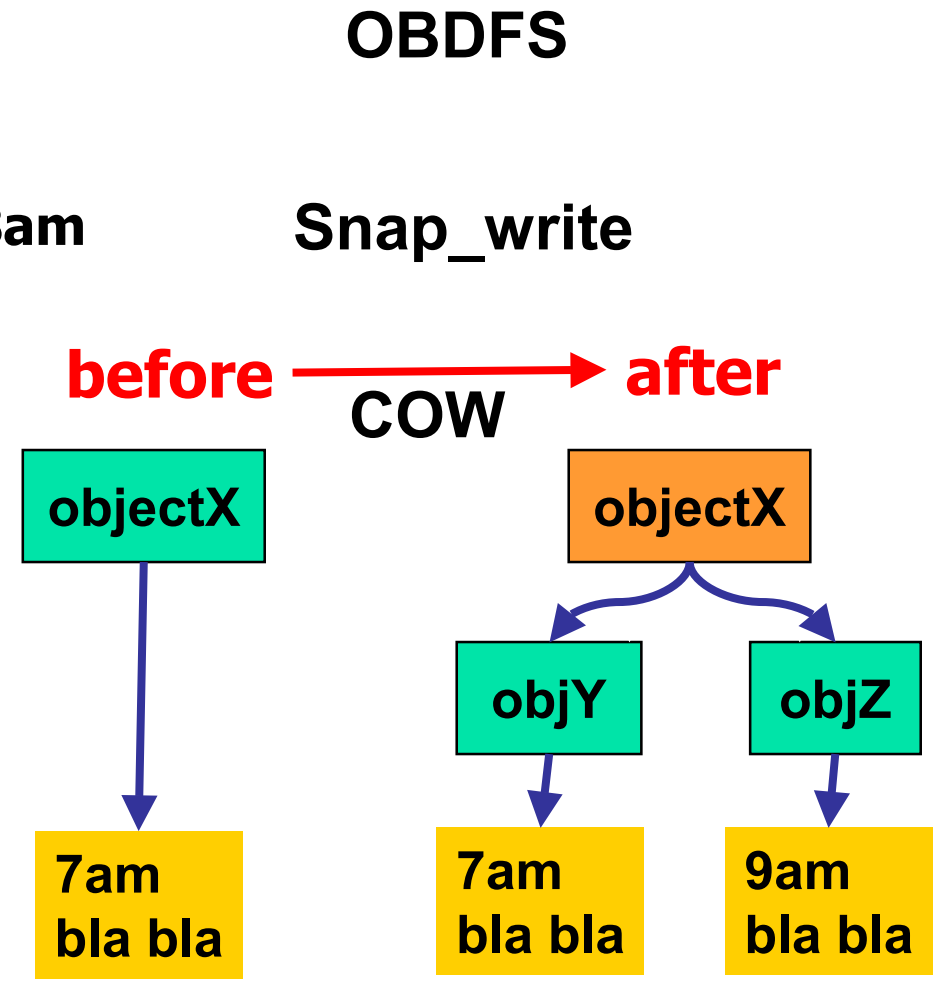
- **Result:**
  - /dev/obd2 is read only clone
  - /dev/obd1 is copy on write (COW) for 8am

Snapshots in action

- **mount /dev/obd1 /mnt/obd**
- **mount /dev/obd2 /mnt/obd/8am**

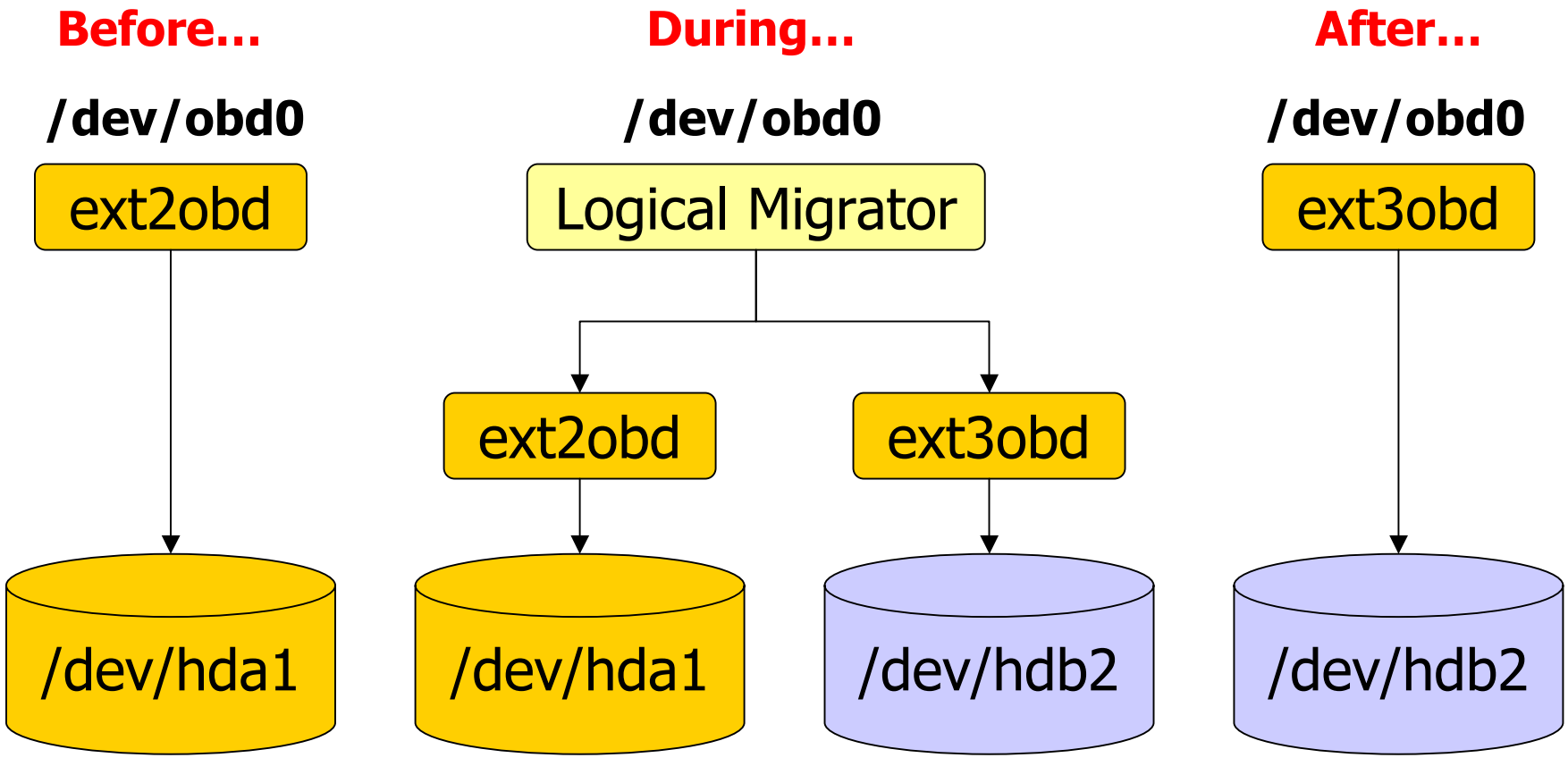
- **Modify /mnt/obd/files**

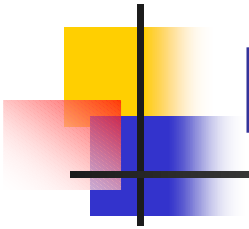
- **Result:**
  - **new copy** in /mnt/obd/files
  - **old copy** in /mnt/obd/8am



Hot data migration:

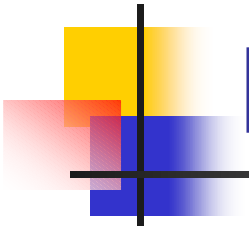
Key principle: dynamically switch device types





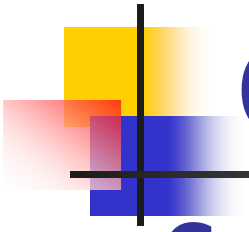
Lustre File System

- Lustre ~ Linux Cluster
- Object Based Cluster File System
  - Based on OBSD's
- Symmetric - no file manager
- Cluster wide Unix semantics: DLM
- Journal recovery etc.



Benefits of Lustre design

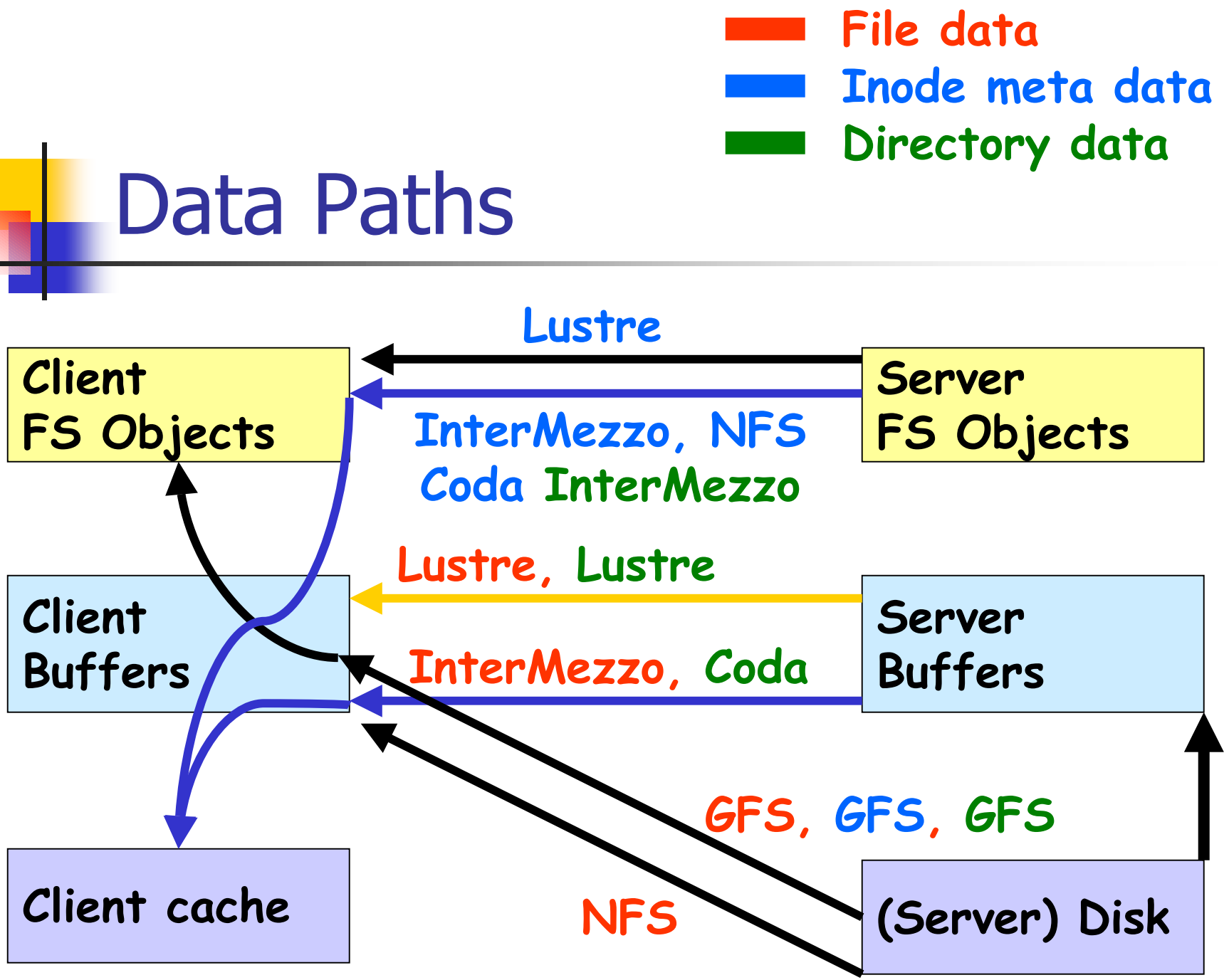
- space & object allocation
  - Managed where it is needed !!
- consequences
  - IBM (Devarakonda etc): less traffic
  - **Much** simpler locking

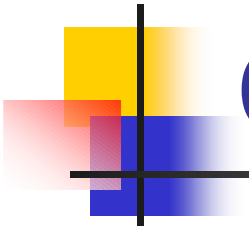


Others…

- **Coda:**
  - mobile use, server replication, security
- **GFS:**
  - shared storage file system, logical volumes
- **InterMezzo:**
  - Smart “replicator”. Exploits disk fs.
- **Lustre**
  - shared storage file system
  - likely best with smarter storage devices
- **NFS**

Data Paths


File data
Inode meta data
Directory data
Lustre
Client
FS Objects
Server
FS Objects
InterMezzo, NFS
Coda InterMezzo
Lustre, Lustre
Client
Buffers
Server
Buffers
InterMezzo, Coda
GFS, GFS, GFS
Client cache
NFS
(Server) Disk



Conclusions

- Linux needs this stuff
  - Badly
- Relatively little literature
  - cluster file systems
  - DLMs
- Good opportunity to innovate