Lustre

The Inter-Galactic File System

Peter J. Braam

braam@clusterfs.com

http://www.clusterfilesystems.com

Cluster File Systems, Inc

Key requirements

- I/O throughput – 100's GB/sec
- Meta data scalability – 10,000's nodes, ops/sec, trillions of files
- Cluster recovery – simple & fast
- Storage management – snapshots, HSM
- Networking – heterogeneous networks
- Security – strong and global

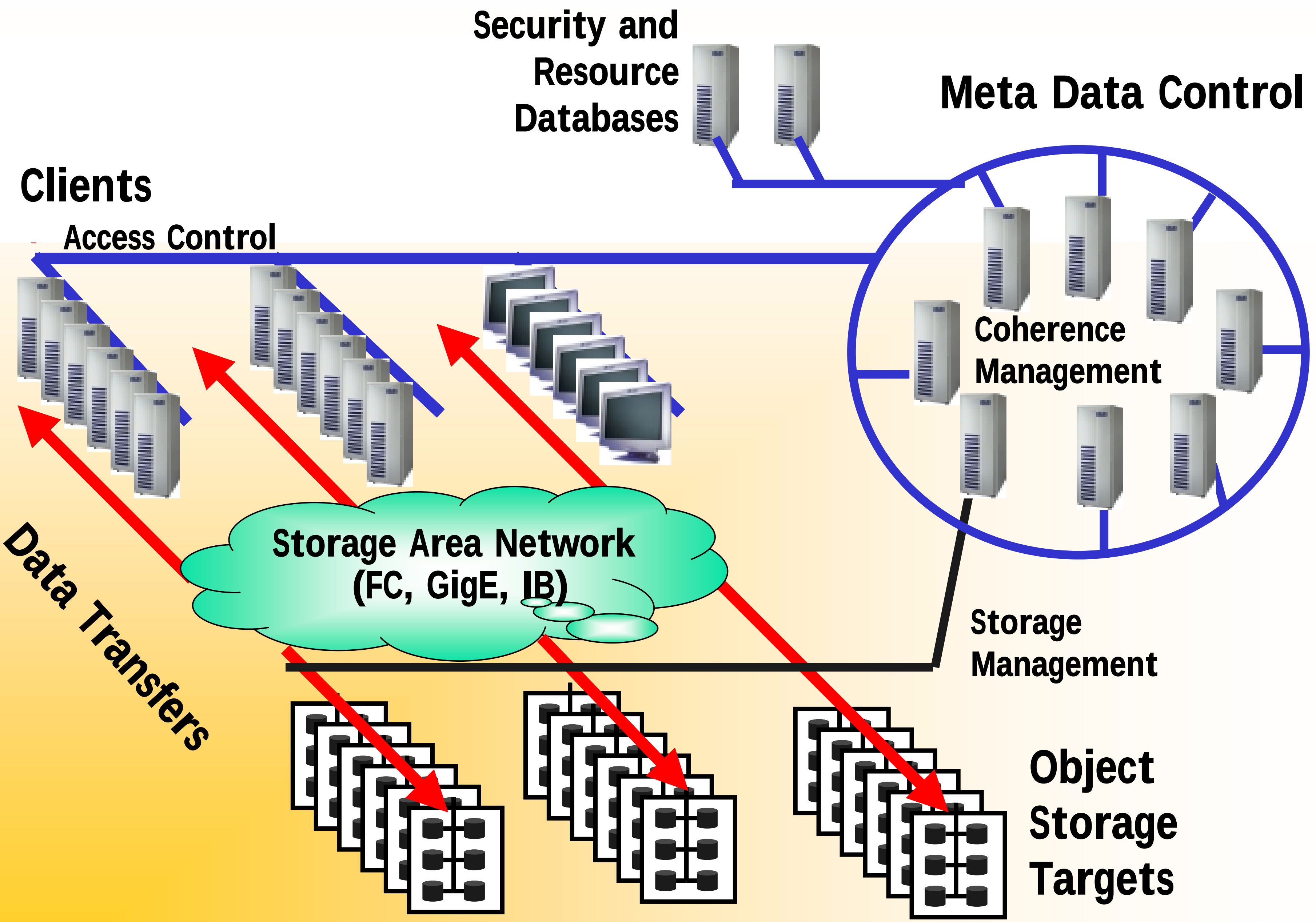
The first 3 years…

- 1999 CMU – Seagate – Stelias Computing
- 2000 Los Alamos, Sandia, Livermore:
  - need new File System
- 2001: Lustre design to meet the SGS-FS requirements?
- 2002: things moving faster
  - Lustre on MCR (1000 node Linux Cluster – bigger ones coming)
  - Lustre Hardware (BlueArc, others coming)
  - Very substantial ASCI pathforward contract (with HP & Intel)

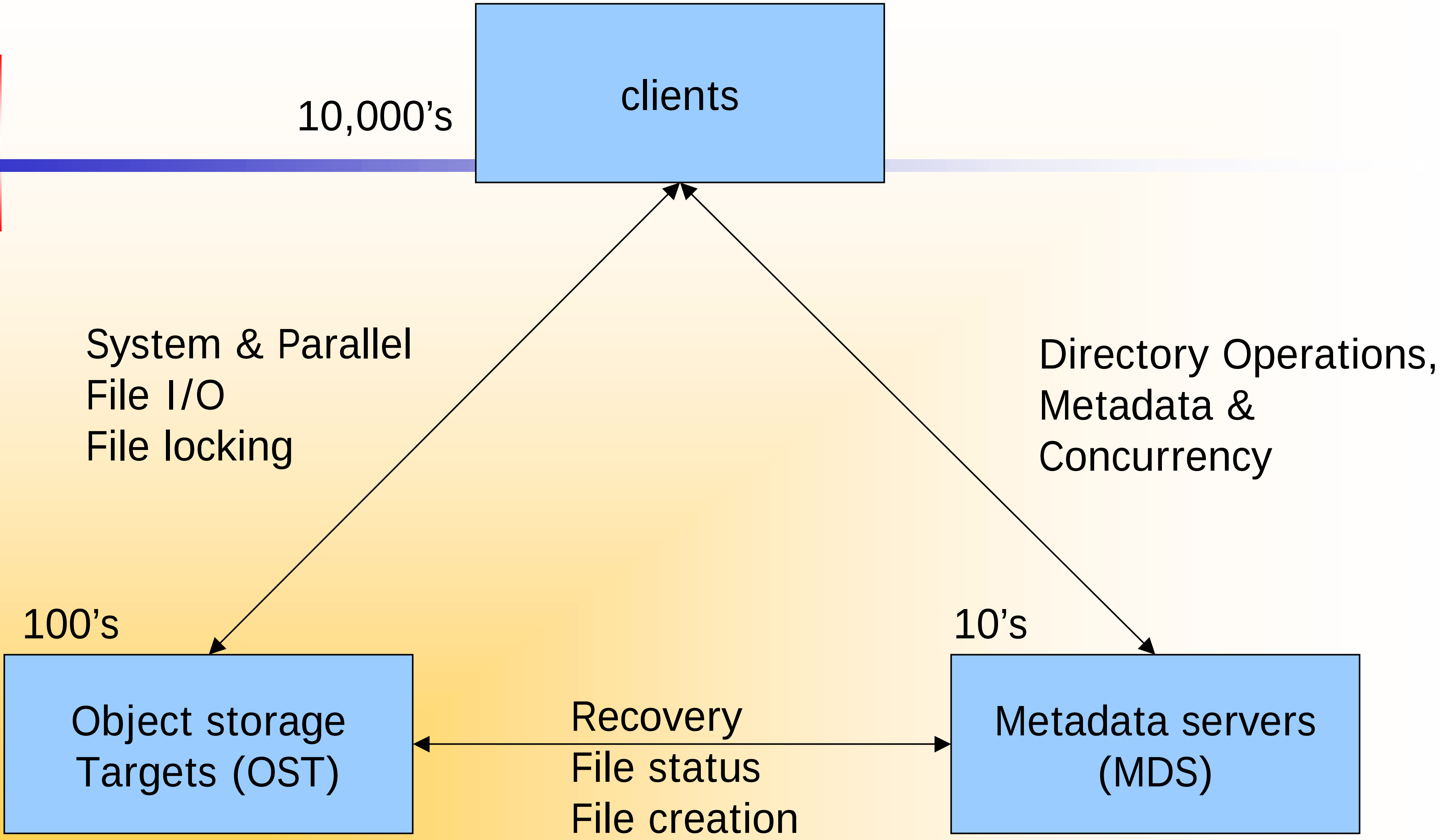
Approach

- Initially Linux focused
- Was given blank sheet
- Learn from successes
  - GPFS on ASCI White
  - TUX web server, DAFS protocol
  - Sandia Portals Networking
  - Use existing disk file systems: ext3, XFS, JFS
- New protocols
  - InterMezzo, Coda

Big Lustre picture



Cluster File Systems, Inc



Lustre System

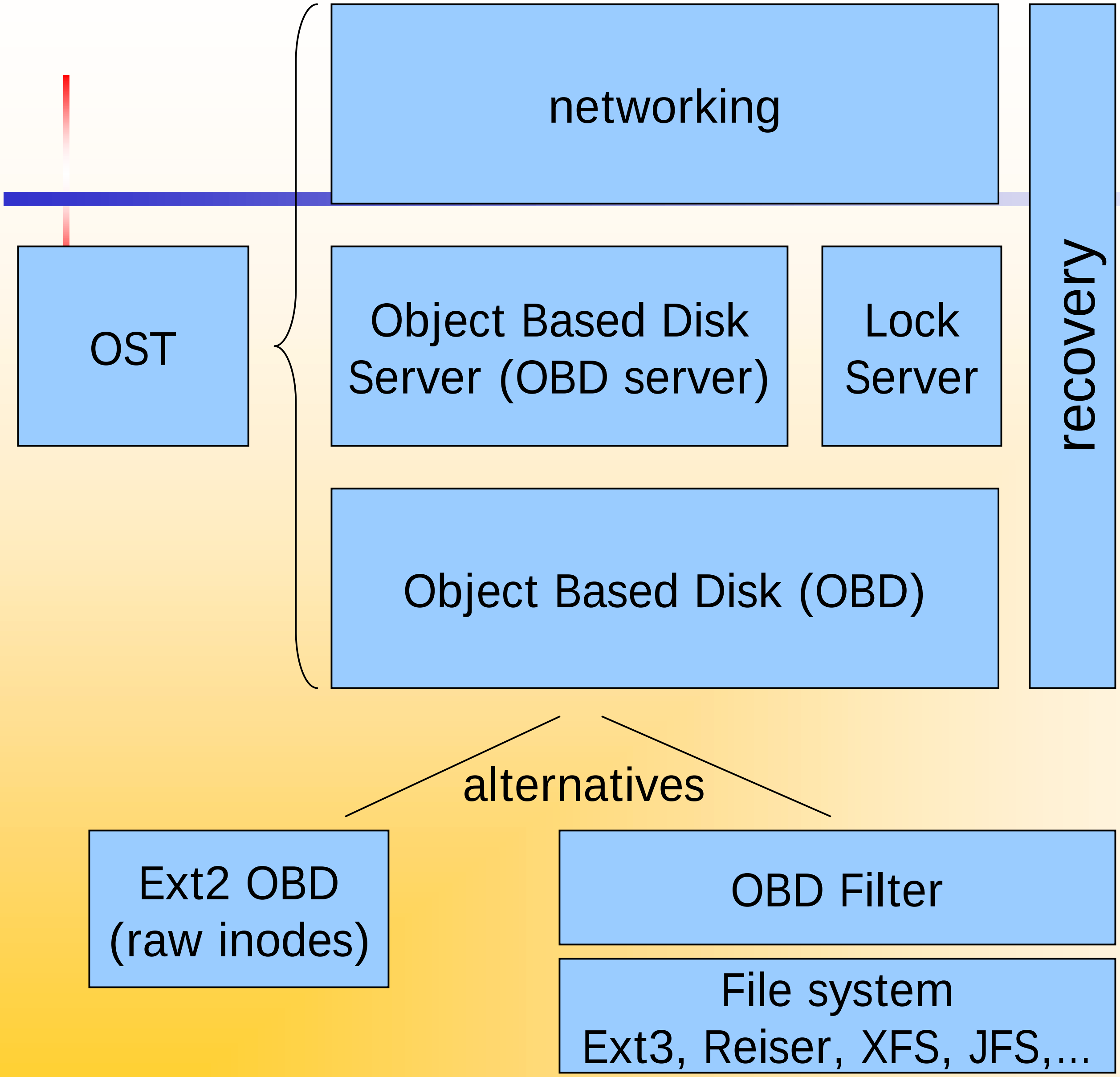
Cluster File Systems, Inc

Ingredient 1: object storage

What is Object Based Storage?

- Object Based Storage Device
  - More intelligent than block device
- Speak storage at "inode level"
  - create, unlink, read, write, getattr, setattr
  - iterators, security, almost arbitrary processing
- So…
  - Protocol allocates physical blocks, no names for files
- Requires
  - Management & security infrastructure

Object Storage Target



Cluster File Systems, Inc

How does object storage help?

File – I/O

- Open file on metadata system
- Get information
  - What objects
  - What storage controllers
  - What part of the file
  - Striping pattern
- Use connection to storage controllers you need
  - Do logical object writes to OST
  - From time to time OST updates MDS with new file sizes

I/O bandwidth requirements

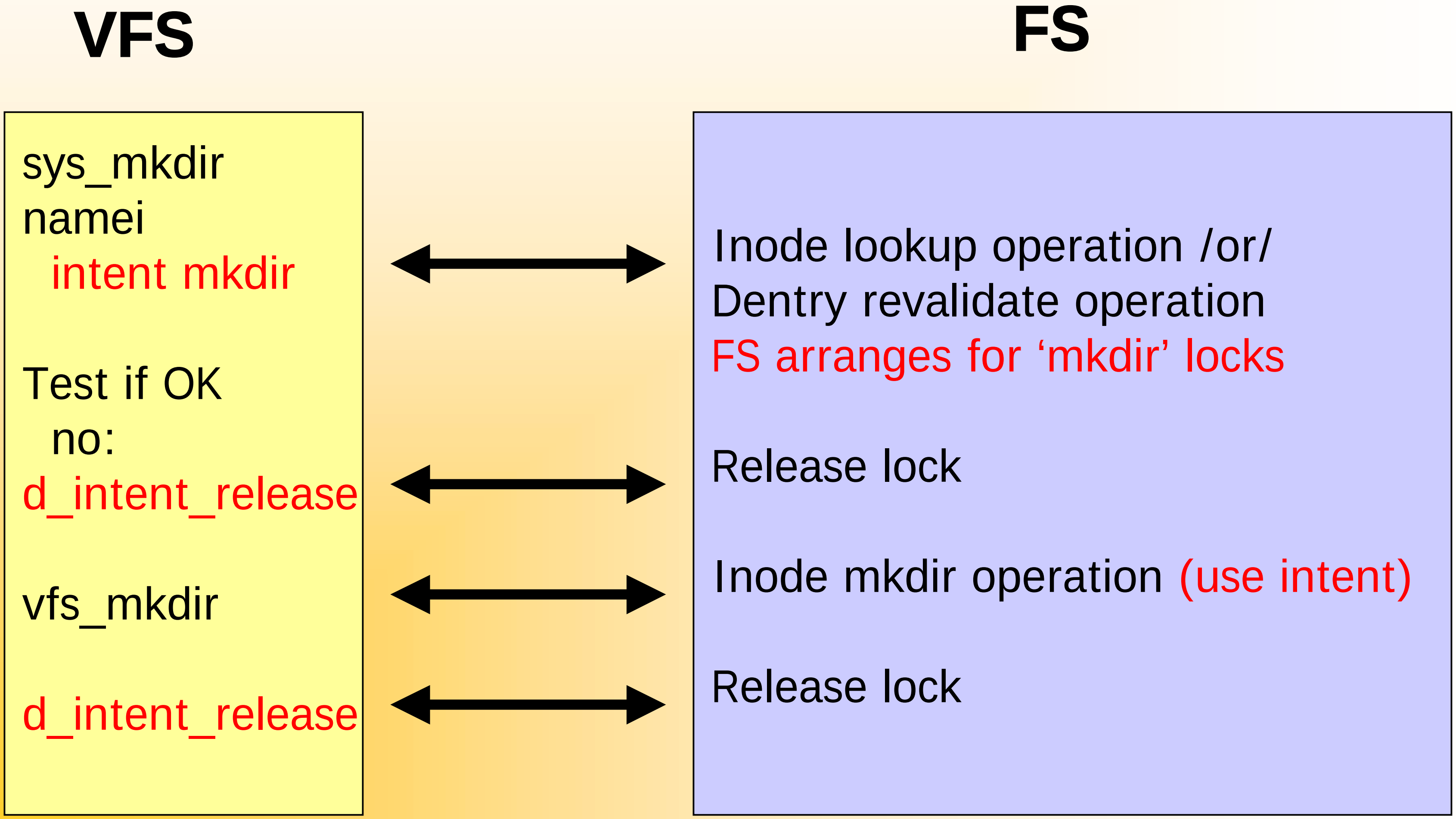
- Required: 100’s GB/sec
- Consequences:
  - Saturate 100’s – 1000’s of storage controllers
  - Block allocation must be spread over cluster
  - Lock management must be spread over cluster
- This almost forces object storage controller approach

Ingredient 3: metadata handling

Intent based locks & Write Back caching

- Protocol adaptation between clients and MDS
- Low concurrency - write back caching
  - On client in memory updates with delayed replay on MDS
- High concurrency
  - Want single network request per transaction, no lock revocations
  - Intent based locks – lock includes all info to complete transaction

Linux VFS changes: intent lookups



VFS

sys_mkdir
namei
  intent mkdir

Test if OK
  no:
d_intent_release

vfs_mkdir

d_intent_release

FS

Inode lookup operation /or/
Dentry revalidate operation
FS arranges for 'mkdir' locks

Release lock

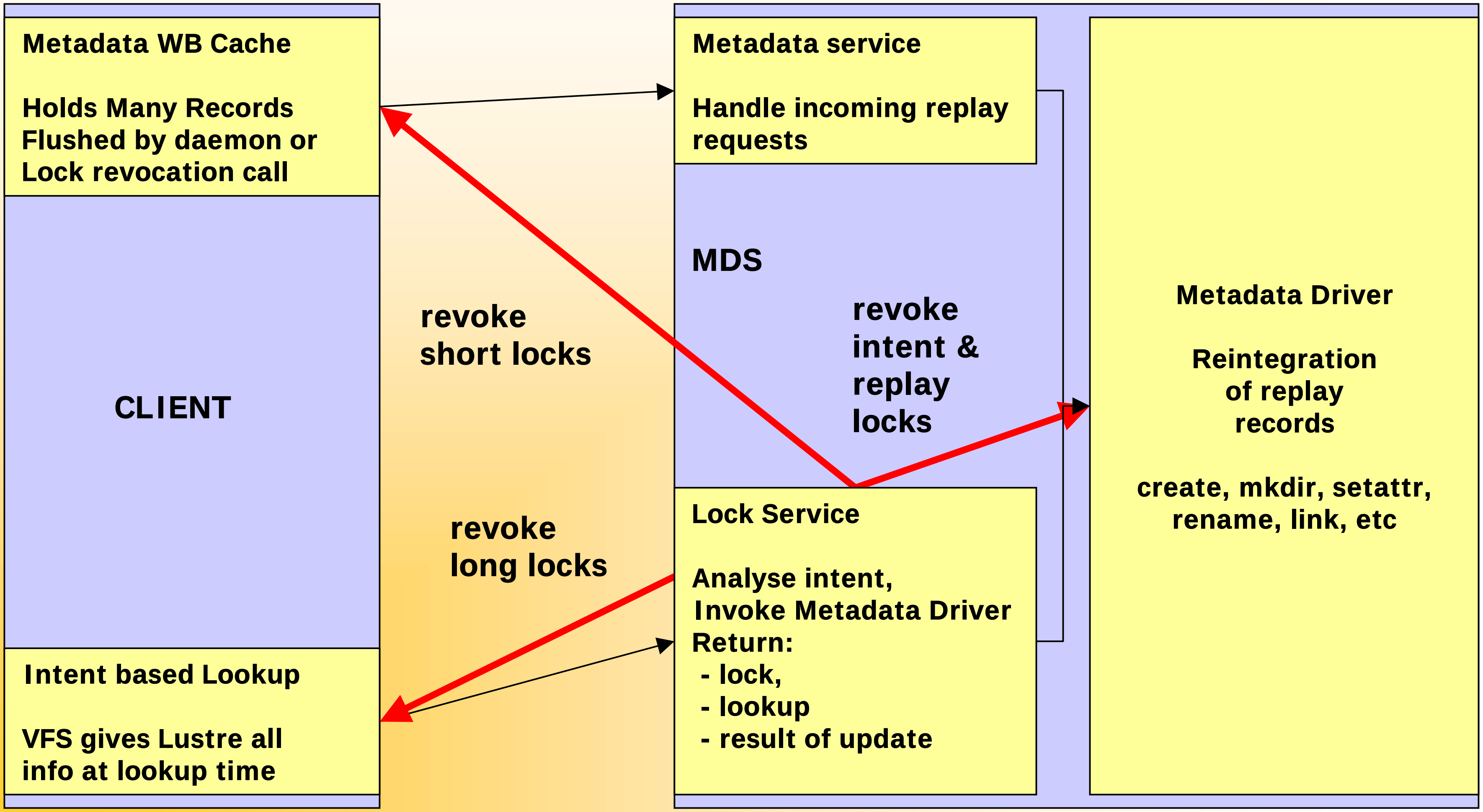
Inode mkdir operation (use intent)

Release lock

Two types of metadata locks:

- Long locks –
  - Lock whole pathname, help with concurrency
  - e.g. locking the root directory is BAD
    - so lock /home/peter & /home/phil separately
- Short Locks
  - Lock a directory subtree -help for delegation
  - e.g. a single lock on /home/phil is GOOD

Metadata updates



Metadata WB Cache

Holds Many Records
Flushed by daemon or
Lock revocation call

CLIENT

Intent based Lookup

VFS gives Lustre all
info at lookup time

revoke
short locks

revoke
long locks

Metadata service

Handle incoming replay
requests

MDS

revoke
intent &
replay
locks

Lock Service

Analyse intent,
Invoke Metadata Driver
Return:
- lock,
- lookup
- result of update

Metadata Driver

Reintegration
of replay
records

create, mkdir, setattr,
rename, link, etc

Ingredient 3: Storage Networking

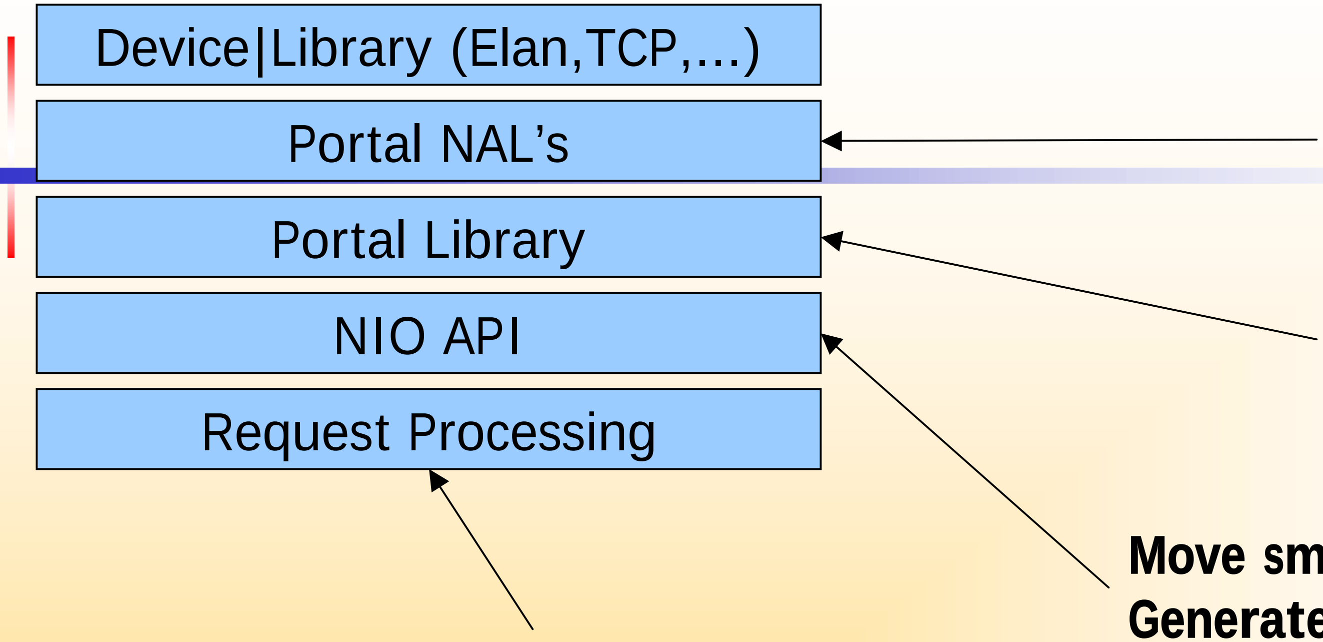
Lustre networking

- Currently runs over
  - TCP,
  - Quadrics
  - Myrinet (almost)
- Other networks we are looking at:
  - SAN's
  - I/B
  - NUMA interconnects (@ GB/sec)
  - SCTP

Portals

- Sandia Portals message passing
  - simple message passing API
  - support for remote DMA
  - support for plugging in device support
  - Network Abstraction Layers

Lustre Network Stack



Device|Library (Elan,TCP,...)

Portal NAL's

Portal Library

NIO API

Request Processing

now: Elan & IP
soon: Sandia, GM

Sandia's API
CFS improved impl.

Move small & large buffers
Generate events

0-copy marshalling libraries
service framework
client request dispatch
connection & address naming
generic recovery infrastructure

Ingredient 4: Storage Management

Components of OB Storage

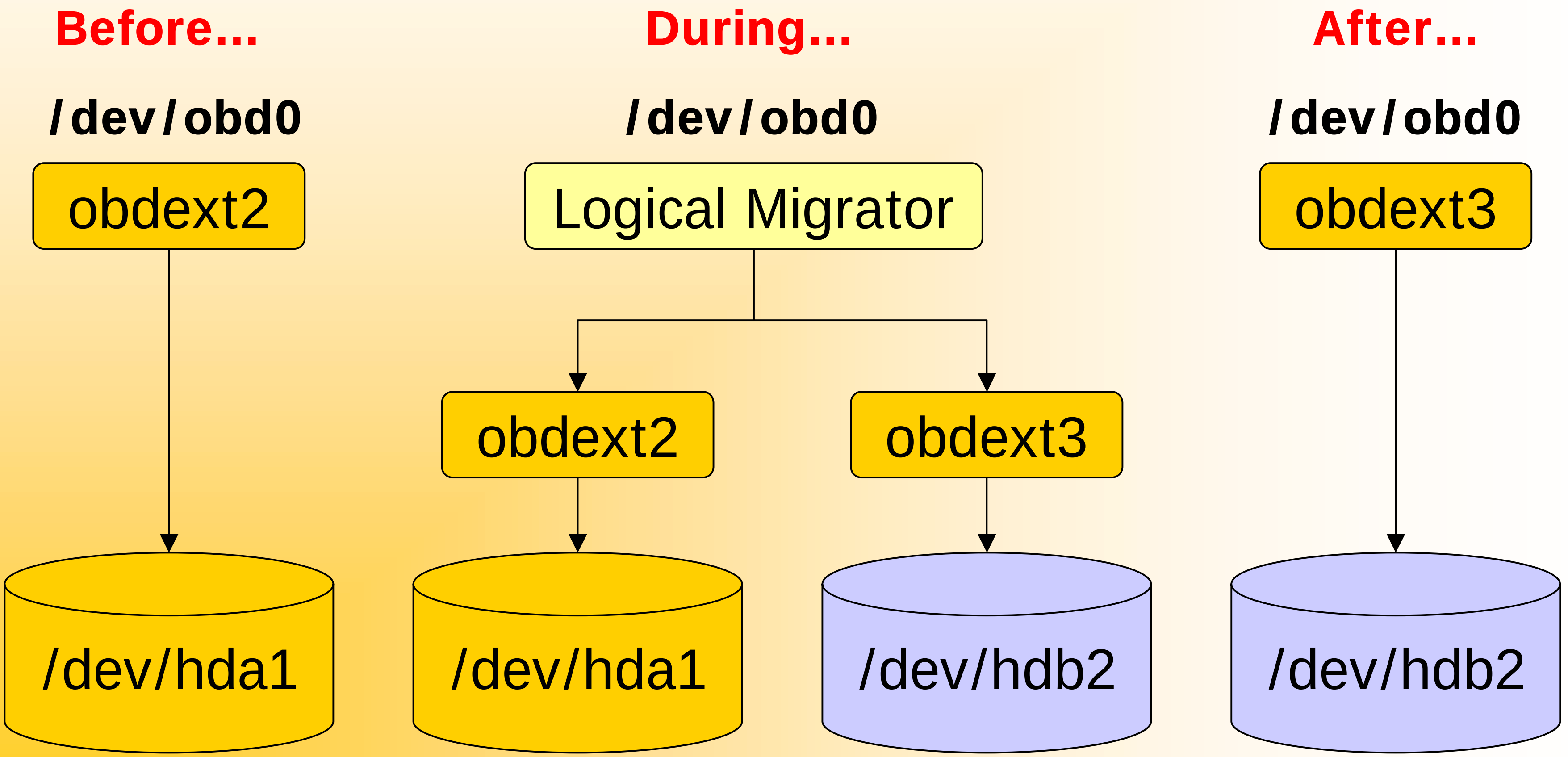
- Storage Object Device Drivers
  - **Class driver** – attach driver to interface
  - **Targets, clients** – remote access
  - **Direct drivers** – to manage physical storage
  - **Logical drivers** – for intelligence & storage management
  - **Object storage "applications"** – eg. the file system

Examples of logical modules

- Storage management:
  - System software, trusted
  - Often inside the standard data path,
  - Often involves iterators
  - Eg: security, snapshots, versioning data migration, raid
- Lustre offers active disks
  - almost arbitrary intelligence can be loaded into OST driver stack

Example of management: hot data migration:

Key principle: dynamically switch object device types



Conclusions

- We think Lustre can run well on BlueGene

Cluster File Systems

- Small scale service company
  - Open Source development
  - contract work for Government labs
  - some consulting and collaboration with industry
- Extremely specialized and extreme expertise
  - we only do file systems and storage
- Investments etc
  - Please visit “Save the Children”
  - no thank you – it’s perfectly possible to go forward without