zFS - A Scalable Distributed File System Using OSD

Ohad Rodeh
Uri Schonfeld
Avi Teperman

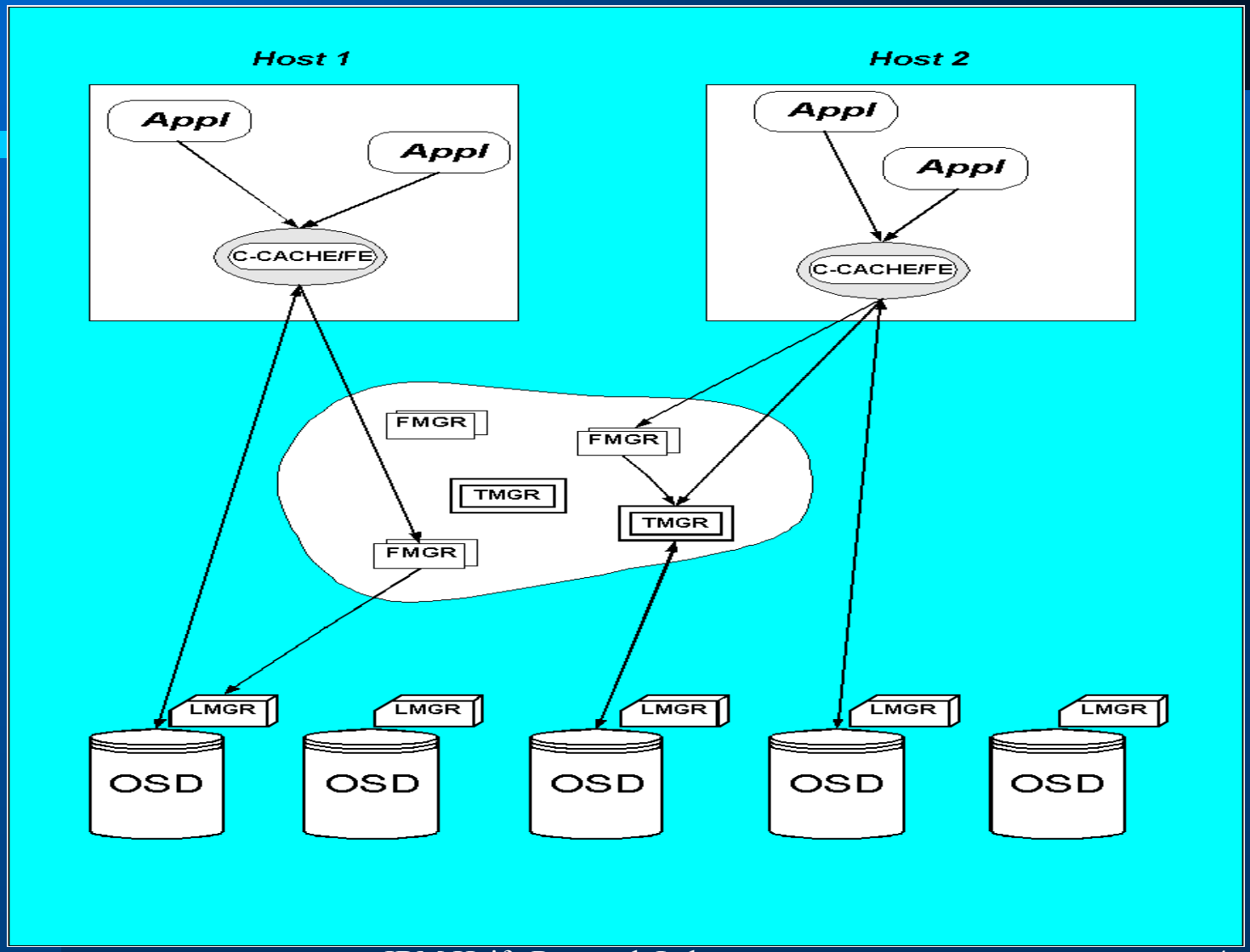
zFS Background

- zFS is part of continued research on storage
  - Started with Distributed Sharing Facility
  - Continued with Antara Object Store Device
- zFS is an attempt to explore completely distributed File System based on Object Store Devices

zFS Goals

- A File System that operates well on few or thousands of machines
- Built from off-the-shelf components with Object Store Devices (OSDs)
- Use the memory of all machines as a global cache
- Achieve almost linear scalability

zFS Architecture



zFS Components

- Object Store Device (OSD)
- Lease Manager (LMGR)
- File Manager (FMGR)
- Transaction Manager (TSVR)
- Front-End/Cache (FE/Cache)

- *No Single Point of Failure*

zFS Components
Object Store Device (OSD)

- **OSD enables:**
  - **Creation/deletion of objects**
  - **Read/write byte ranges from/to objects**
- **OSD provides:**
  - **File abstraction**
  - **Security**
  - **Safe writes**
- **Using OSD allows zFS to focus on File Management and Scalability**

zFS Components
Lease Manager (LMGR)

- **The need for lease manager stems from the following facts:**
  - **Locking mechanism is required to control access to disks**
  - **In SAN file systems clients can write directly to OSDs. Therefore:**

To work in SAN file systems the OSDs themselves have to support locking

zFS Components
Lease Manager (LMGR)

- **To reduce OSD's overhead the following mechanism is used:**
  - **Each OSD has *one* lease manager**
  - **OSD maintains and grants to its LMGR one *major lease***
  - **LMGR grants *object leases* to the FMGRs requesting it**
  - **FMGR grants *range leases* to the FEs requesting it**

zFS Components
Lease Manager (LMGR)

- We prefer leases over locks to avoid the mechanism for detecting failed machine holding a lock
  - In case of lease, we can reclaim it after its lease period expires
- Leases incur the overhead of leases renewal

zFS Components
File Manager (FMGR)

- When a file is first opened a file manager is assigned to it.
- Each lease request on the file is mediated by the FMGR.
- The FMGR interacts with the proper LMGR to get the object lease and grants range leases to the FE.

zFS Components
File Manager (FMGR)

- **The FMGR keeps track of:**
  - **Where each file's extents reside**
  - **Each file's lease.**

  If client *X* requests block and lease which resides on client *Y* the FMGR will direct FE/Cache on *Y* to send the requested data to *X*.

- **File manager assignment is dynamic.**

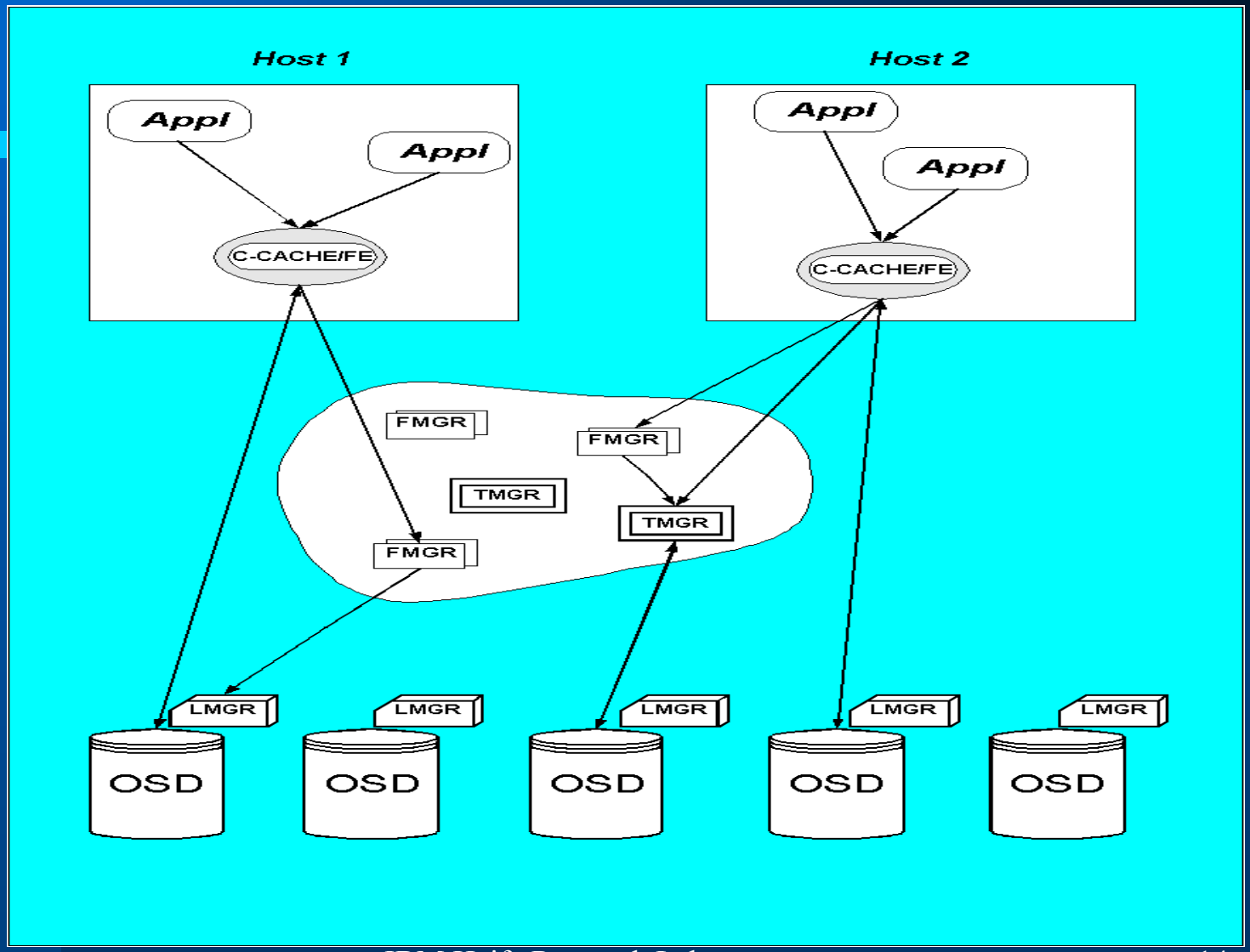
zFS Components
Transaction Manager (TMGR)

- Meta data operations handle several objects
- To ensure file system consistency zFS implements them as distributed transactions
- All meta data operations are handled by the TSVR

zFS Components
Front-End / Cache

- **FE**
  - **Runs on every client machine**
  - **Presents to the application/user the standard file system API**
  - **Provides access to zFS files and directories**
- **Cache**
  - **Provides access to zFS data and metadata in local memory to other machines**

zFS Architecture



Current Status

- zFS implemented on Linux
  - Kernel 2.4.19
- Currently
  - Most components implemented, started integration

Related Documents

- **zFS Web Site**
  - **http://www.haifa.il.ibm.com/projects/storage/zFS/index.html**
- **DSF Web Site**
  - **http://www.haifa.il.ibm.com/projects/storage/dsf/index.html**

Backup Slides

zFS Failure Handling

$LMGR_i$ failed

- Detected by all FMGRs that hold leases for objects of OSD_i
- Each FMGR informs all FEs holding files on OSD_i to flush their dirty data and release files
- FMGRs instantiate new $LMGR_i$ which tries to get the OSD_i major lease
- Once the previous major lease expires, one $LMGR_i$ gets the major lease and all others are terminated
- Operation on OSD_i resumes

Write Scenario