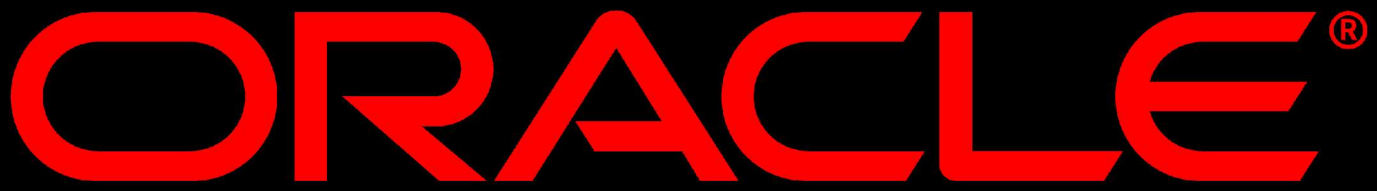

ORACLE

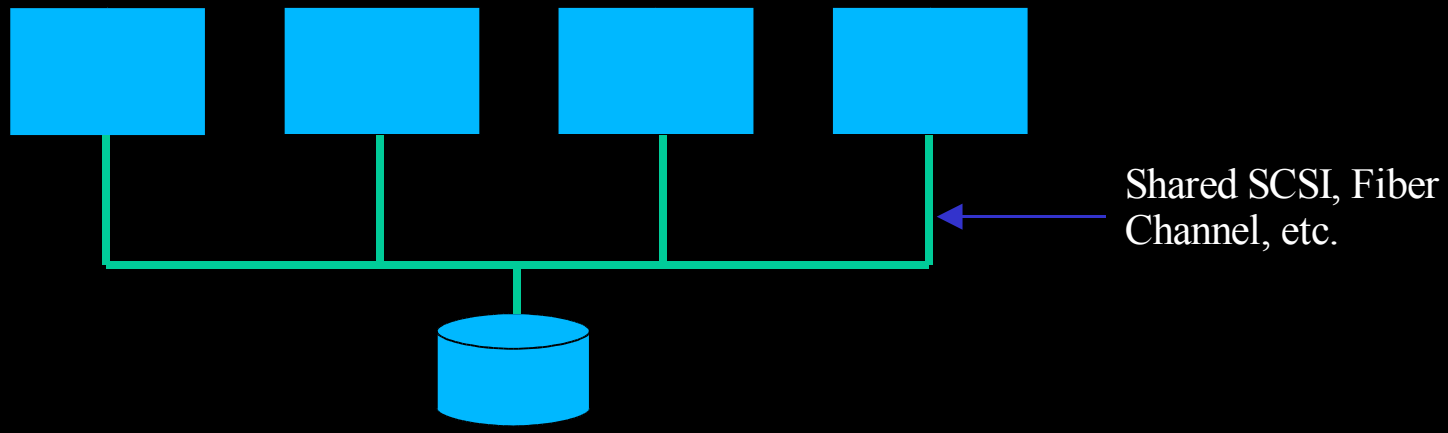
Oracle's Cluster File System on Linux

Mark Fasheh
Software Developer
Oracle Corporation

ORACLE

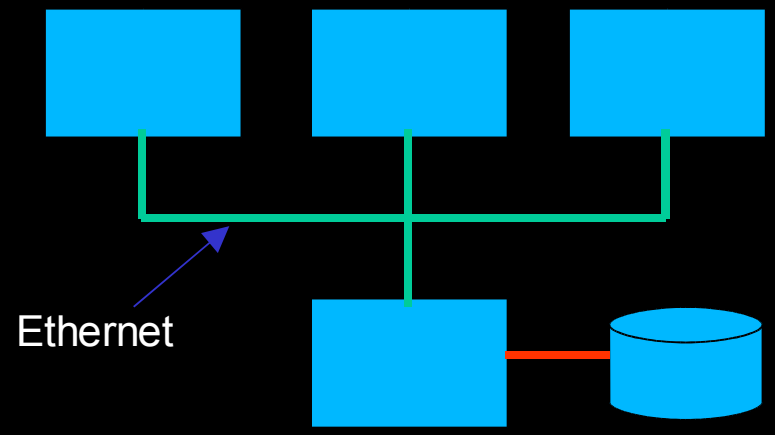
What is OCFS?

- GPL'd Extent Based Cluster File System
- Is a shared disk clustered file system
- Allows two or more nodes to access the same file system
- File system is mounted natively on all nodes
- Supports a maximum of 32 nodes



Is it like NFS?

- No.
- In NFS, the file system is hosted by one node
- Rest of the nodes access the file system via the network
- Single point of failure
- No node recovery
- Slow data throughput



Why does Oracle need it?

- Oracle's Real Application Cluster (RAC) database, uses a shared disk
- As most OSes do not provide a shared disk cluster file system, RAC data files, control files, etc. need to exist on a raw partition
- Raw is hard to manage
- Moreover, Linux 2.4 allows a max of 255 raw partitions
- No auto-extending of partitions

Why does Oracle need it?

- OCFS allows for easier management as it looks and feels just like a regular file system
- No limit on number of files
- Allows for very large files (max 2TB)
- Max volume size 32G (4K block) to 8T (1M block)
- Oracle DB performance is comparable to raw

How do I use it?

- Hardware Setup
  - 2+ node setup with some sort of shared disk
  - Shared disk could be Shared SCSI, Fiber Channel, etc.
  - For testing purposes, recommend using FireWire (very cheap)
  - http://otn.oracle.com/
  - OTN site has README, 2.4.20 kernel with FireWire fixes and the OCFS module

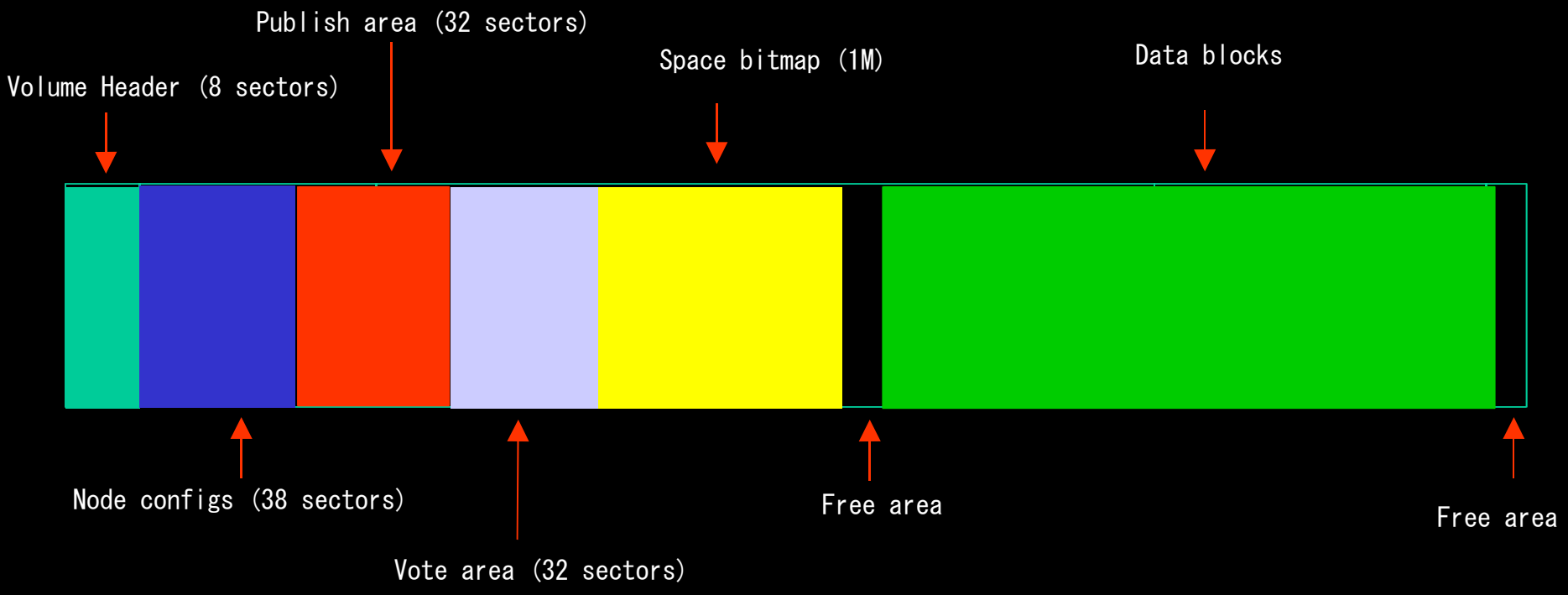
Process Architecture

- OCFS is a kernel module
- On the first mount creates 2 kernel threads
  - [ocfsnm-0] => one for each mounted volume. Thread runs in a loop reading the volume for any lock requests from other nodes.
  - [ocfslsnr] => one on a node. Is a listener for the network dlm. Is activated only when comm_voting is enabled. Currently disabled by default. Will be enabled in a future release.
  - After last dismount, [ocfslsnr] exits.

Process Architecture

- The third important pid is that of the user-space process which is accessing the fs. e.g., cp, mv, dbwr, etc.
- All lock requests on a node are triggered by the user-space process.
- All lock requests by other nodes are serviced by the ocfsnm-x thread.

Volume Layout



Note: Not drawn to scale

Node Configuration

- Node name, ip address, ip port and guid is stored in this area
- Slots 0 to 31 represent node numbers 0 - 31
- Node number is auto-allocated the first time a node mounts a volume
- A node could have different node numbers across multiple ocfs volumes
- /proc/ocfs/<volume_num>/nodenum
- OCFS identifies a node by its guid

Publish Area

- Every node owns one sector for writing, aka, its publish sector
- In it, the nodes write the timestamp at regular intervals to indicate to the other nodes that they are alive
- Nodes also use their publish sector to request locks on a resource
- Resources are structures on disk and its number is its byte offset

Vote Area

- Every node owns one sector for writing, aka, its vote sector
- In it, nodes vote for the resource lock asked to by another node
- Requesting node collects the votes from all the nodes and takes the lock if all vote OK
- The lock state is written on the disk (for files in the file entry, for bitmap in the bitmap lock sector)

Distributed Lock Manager

- OCFS requires locks only for the file system meta-data changes
- Does not protect file data changes
- Expects the application to be cluster-aware
- Oracle RAC is cluster-aware and it performs its own intelligent caching and locking of file data

Distributed Lock Manager

- OCFS also has a network-based dlm
- In it, the node requesting a vote just sends a vote-request packet to all interested nodes
- The nodes inturn reply using the vote-reply packet
- When activated, the publish sector is only used to identify alive nodes whereas the vote sector is unused
- The disk-based dlm gets automatically activated whenever one or more “alive” nodes is not heard of on the network

Space Management - Bitmap

- Each bit in the space bitmap indicates free/alloc state of a data block
- Bitmap size is fixed to 1M
- Size of block size determines max size of volume

```
max_vol_size = block_size * 1M * 8
```

- Block sizes can be 4K, 8K, 32K, 64K, 128K, 256K, 512K or 1M

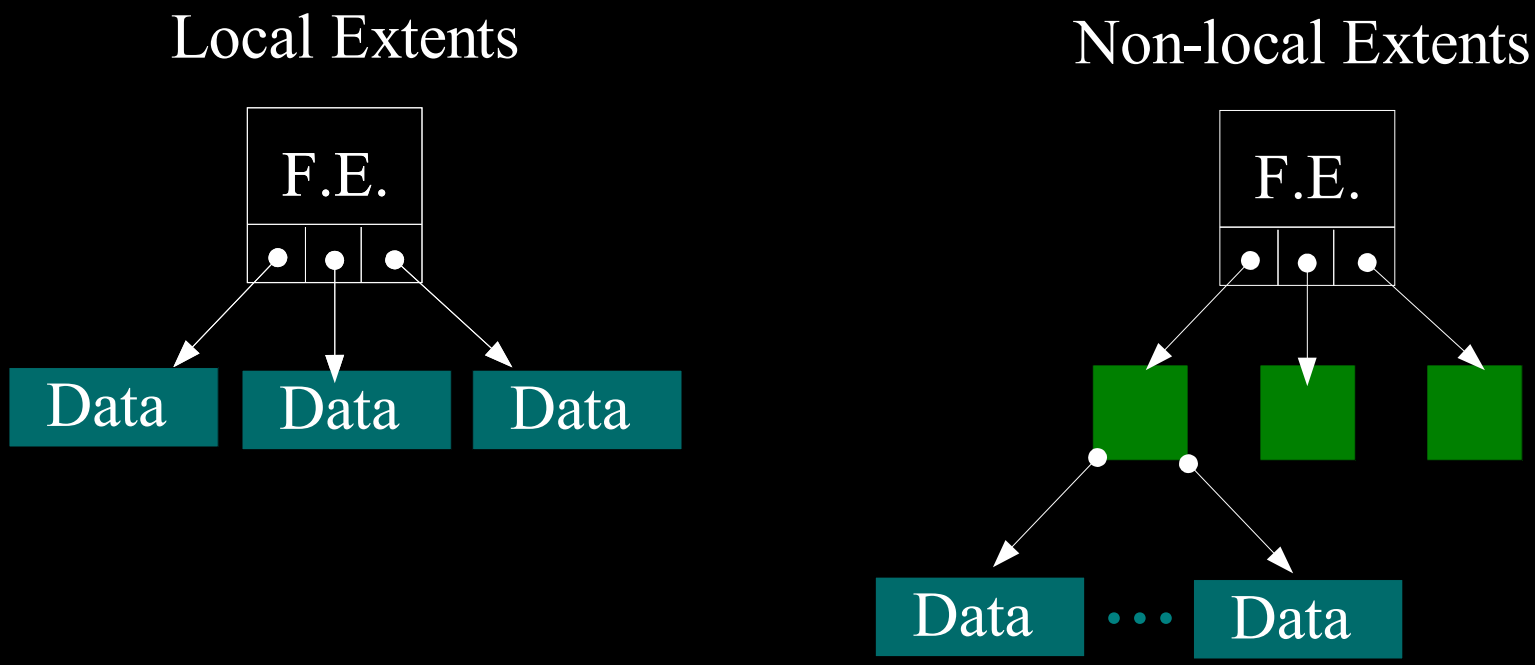
Space Management

- Meta-data and file data allocated space from the same bitmap
- Each meta-data on disk has a lock structure which holds the lock state
- System files allocated using the same scheme
- System files are used for log data, etc.
- Are hidden for regular file system calls

Space Management - File

- Uses extent based space allocation for files rather than the block based (ext2)
- Requires less accounting for very large files
- File entry initially has 3 direct extent pointers
- When file has >3 extents, the extent pointers become indirects
- When file has >54 extents, the extent pointers become double indirects

Space Management - File



- Green squares are indirect blocks which hold 18 extent pointers each.
- Can have up to three levels of indirect pointers before you've run out of theoretical space.

Space Management - Directory

- Directory is a 128K block
- It includes 255 (512 byte) file entries
- Each file entry represents a file, sub-dir or link
- File entry houses the name of the file/sub-dir/link, attributes, locking info
- When the number of file in a dir > 255, another 128K block is linked

Space Management - Directory

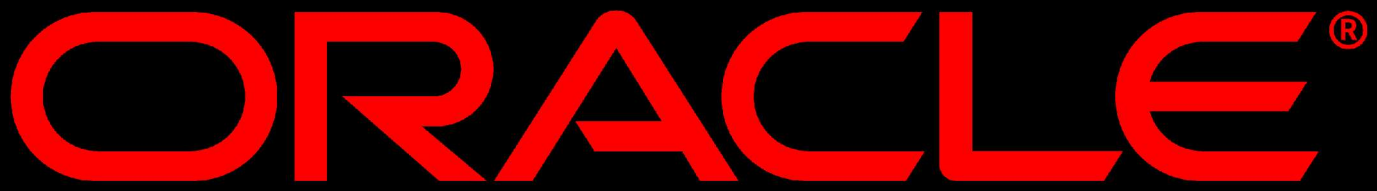
Future Releases

- Improve performance when dealing with large number of files
- Integrate with RH's Cluster Manager
- Activate the network-based dlm

Improvements in Linux

- Make VFS cluster-aware
- Extend locks in VFS to cluster-wide locks
- Generic DLM services
- Cluster Manager
- IO Fencing

ORACLE


ORACLE