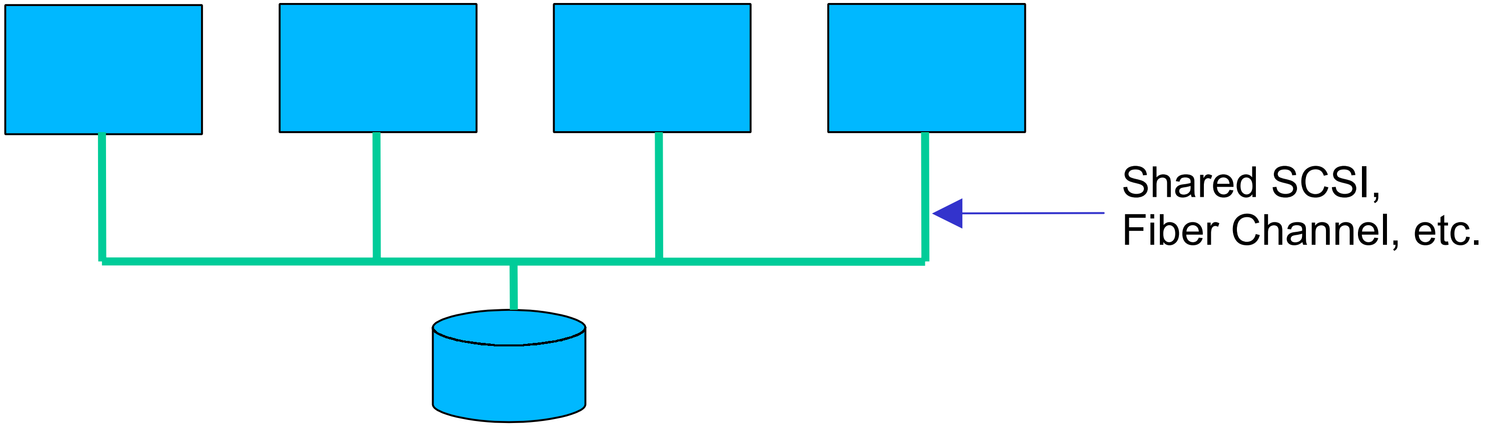
Oracle Cluster File System on Linux

Wim Coekaerts
Oracle
Parag Joshi
HP

HP WORLD 2003
Solutions and Technology Conference & Expo

What is OCFS?

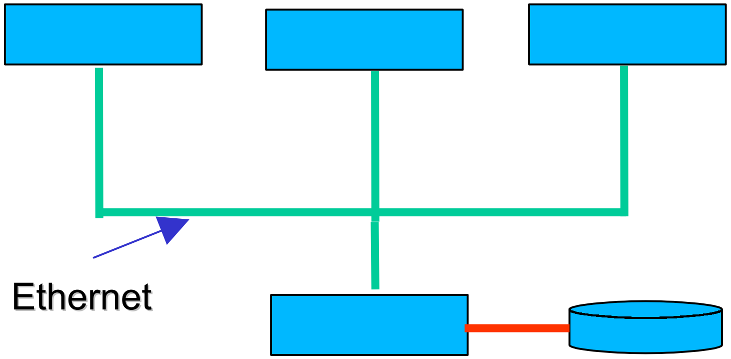
- GPL’d Extent Based Shared Disk Cluster File System
- Allows two or more nodes to access the same file system
- File system is mounted natively on all the nodes
- Supports a maximum of 32 nodes



Shared SCSI,
Fiber Channel, etc.

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
- Requires reliable network



Why does Oracle need it?

- Oracle's Real Application Cluster (RAC) database, uses a shared disk architecture.
- As most Os'es 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
- Availability of cluster while resize / extending

Why does Oracle need it? (cont’)

- OCFS allows for easier management as it looks and feels just like a regular file system
- No limit on the number of files
- Allows for very large files (up to 2TB)
- Maximum volume size is 32GB up to 8TB depending on chosen clustersize of the filesystem
- Oracle database performance on OCFS comparable to raw devices speed

What does the database provide?

- Multi-node data caching (cache fusion)
- Multi-node data locking
- Journal it’s own operations (DB logfiles)
- On Linux and Windows we provide our own cluster management software (oracm)

How do I use it?

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

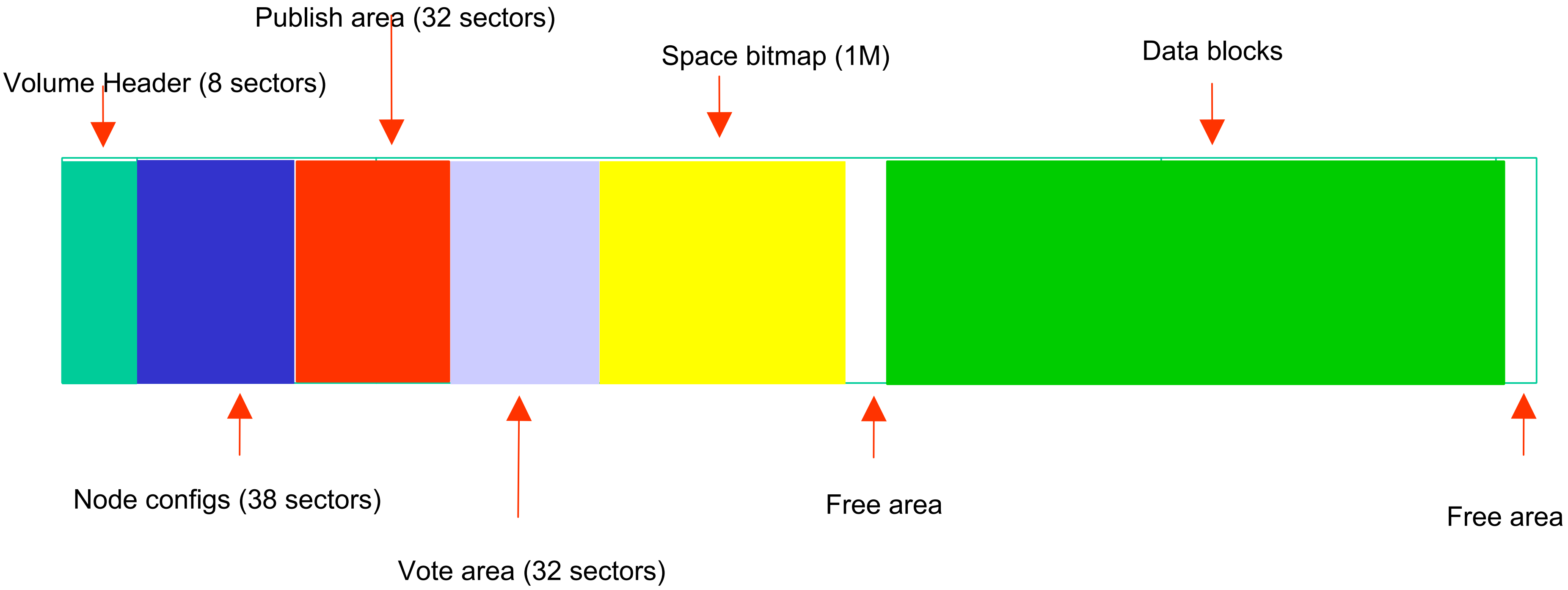
Process Architecture

- 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 but provides a reasonable performance improvement when enabled.
  - After last dismount, [ocfslsnr] exits.

Process Architecture (cont..)

- 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


Publish area (32 sectors)
Volume Header (8 sectors)
Space bitmap (1M)
Data blocks
Node configs (38 sectors)
Free area
Free area
Vote area (32 sectors)

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 1-32
- 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)

Locking

- 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
- Cache coherency for regular data done through the applications

Locking (cont’d)

- OCFS also has a network-based lockmanager
- In it, the node requesting a vote just sends a vote-request packet to all interested nodes
- The nodes in turn 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 locking 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 1MB on disk (so 8m bits)
- 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

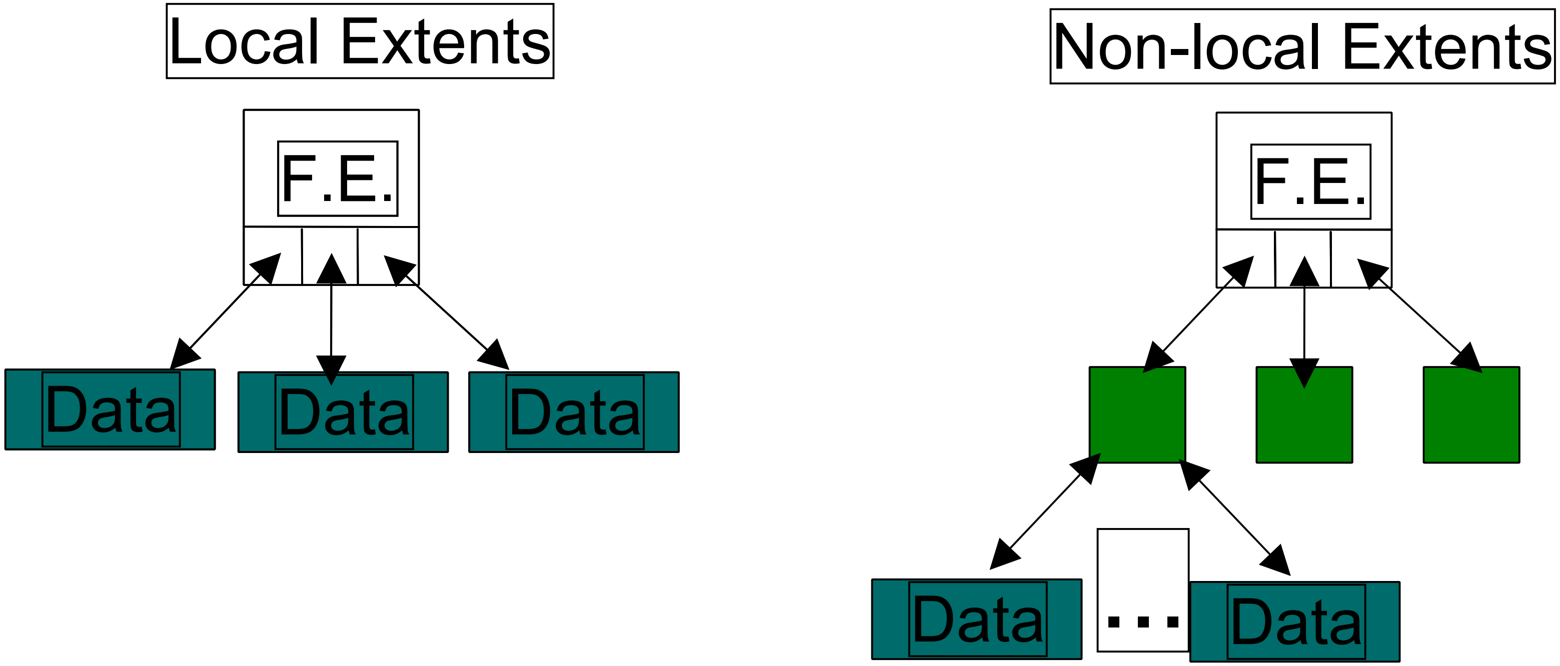
Space Management

- 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 (cont'd)

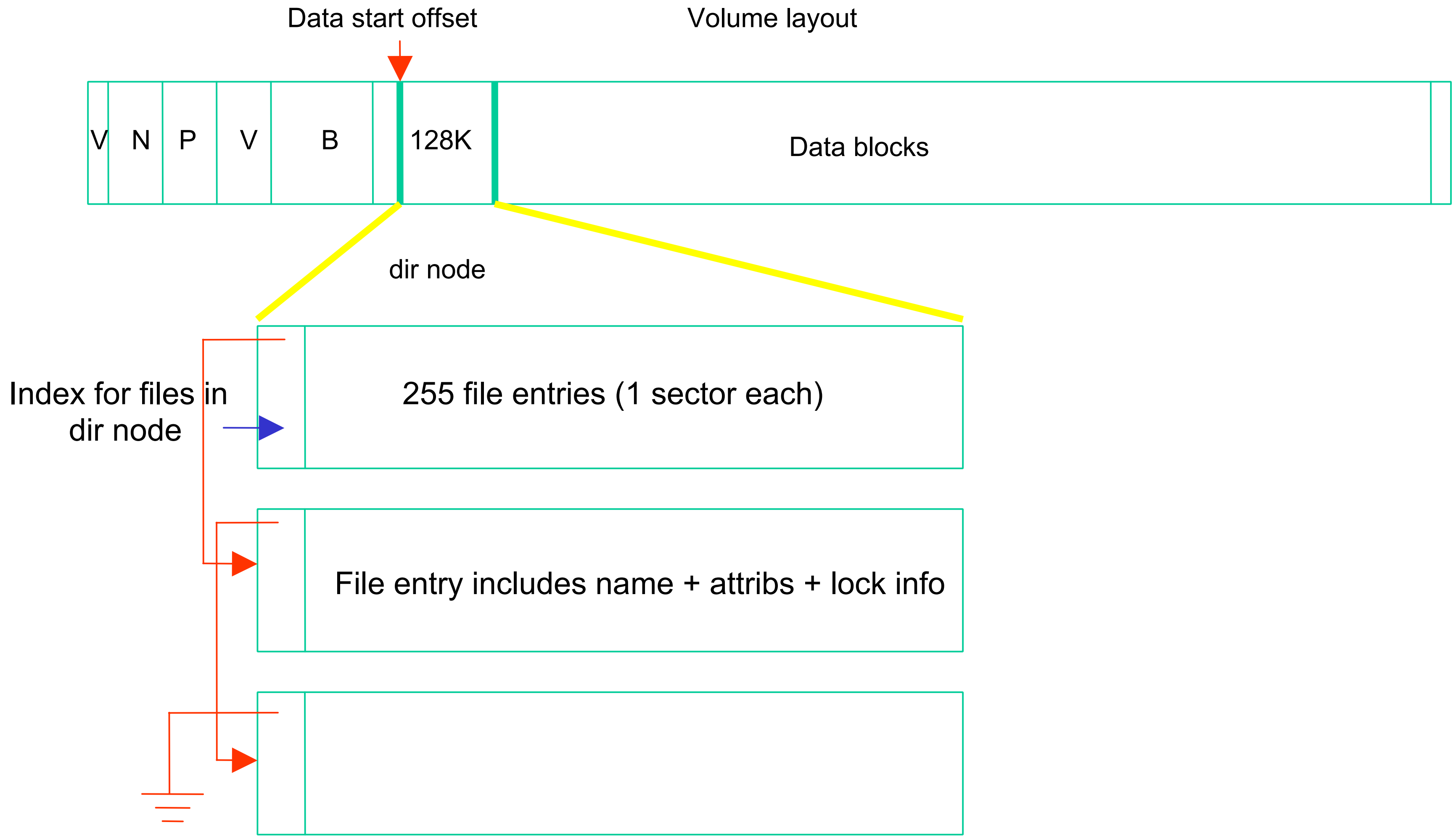


- Green squares are indirect blocks which hold 18 extent pointers each.
- Can have up to three levels of indirect pointers before you've run out of theoretical space.

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 (cont’d)



Data start offset

Volume layout

Index for files in dir node

255 file entries (1 sector each)

File entry includes name + attribs + lock info

Journaling

- Two journal files per node
- Journals are pre-allocated to 1 megabyte
- Contain transactions made up of one or more log records specifying a specific operation
- Each process commits or aborts it's own journaled operations.
- We only do metadata journalling

Journaling (cont'd)

- Recover logfile contains operations to be done in case of a transaction abort
- Cleanup logfile contains operations to be done during a transaction commit
- NM thread handles node recovery
- Recovery is also journaled to prevent multiple concurrent recoveries

OCFS Version 2 Changes

- Main Goal is to work towards general purpose
- Improve performance of meta data operations
  - Local space allocation algorithm
  - Remove the bottleneck on space allocation
- Activate the network-based locking by default
- Caching of regular data moving towards general purpose file system (allow for oracle binaries, shared ORACLE_HOME)

OCFS Version 2 Changes (cont'd)

- Kernel Changes
  - Remove recursive locking
  - Remove the current OCFS Journaling code and use the default kernel, JDB code to increase code re-use. Same ability to do online resizing of the filesystem
  - This depends on a cluster-aware volume manager

Interoperability between IA32 and IPF

- OCFS V1 and V2 both are able to have a mixed configuration of nodes mounting the same disk volume together
- Meta data on disk is compatible
- Binary compatibility of course depends on the operating system itself. OCFS does not change the data on disk
- For Oracle, Data file format on Linux for IA32 and IPF is identical (controlfiles, datafiles, logfiles…)

Interoperability between IA32 and IPF (cont’d)

- To move between IA32 and IPF, simply shut down oracle on one architecture and restart on the other. Change-over in a matter of seconds.
- No need to copy database files
- No need to change controlfiles if the ocfs volume is mounted at the same mountpoint (see demo)

Shared Oracle Home

- OCFS V1 addresses most of the management issues as it is related to raw devices.
- OCFS V2 allows for the concept of a shared ORACLE_HOME installation
  - Only one copy of the database software
  - Applying patches in one place (reduced risk of having software out of sync on other nodes)
  - Reduces disk space requirements
- One filesystem with the software installed mounted on every node

Shared Oracle Home (cont'd)

- Concept of Local Files through the use of symbolic links.
- Next step in ease of management for a cluster setup on Linux.
  - Single volume for all log/trace files
  - Single volume for configuration files

OCFS Performance

- Performance benchmarks show that OCFS is similar to Raw.
- O_DIRECT implementation used by Oracle
- CPU overhead is minimal
- IO throughput is equivalent to RAW
- Scalability with large number of users is equivalent

Online backup’s with OCFS

- RMAN is preferred tool for making backups of Oracle on OCFS as well
- We provide tools (dd, cp, tar) with OCFS to allow for online backups to be done

HP WORLD 2003

Solutions and Technology Conference & Expo

Interex, Encompass and HP bring you a powerful new HP World.

interex
shared knowledge • shared power

encompass
AN HP USER GROUP

hp
invent