Introduction to distributed file systems. OrangeFS experience

Andrew Savchenko
NRNU MEPhI, Moscow, Russia
16 February 2013
Outline

1. Introduction
2. Species of distributed file systems
3. Behind the curtain
4. OrangeFS
5. Summary
Introduction

Why one needs a non-local file system?

• a *large* data storage
• a *high performance* data storage
• *redundant* and highly available solutions

There are dozens of them: 72 only on wiki[1], more IRL. Focus on *free software* solutions.
Introduction

Why one needs a non-local file system?

- a large data storage
- a high performance data storage
- redundant and highly available solutions

There are dozens of them: 72 only on wiki[1], more IRL. Focus on free software solutions.
Species of distributed file systems

Terminology is ambiguous!
Network file systems

A single server (or at least an appearance) and multiple network clients.

Examples: NFS, CIFS.
Clustered file systems

Servers sharing the same local storage (usually SAN[2] at block level).

*shared storage* architecture.

Examples: GFS2[3], OCFS2[4].
“Shared nothing” model, independent servers.

*intelligent server* architecture.

Examples: pNFS[5], AFS[6].
Parallel file systems

- Parallel access from clients to (all) servers
- Parallel R/W to the same data file
- Mitigate bandwidth and latency bottlenecks
- Fields of use: HPC and high-end business applications

Examples: Lustre[7], OrangeFS[8], Ceph[9].

Fully parallel file systems:
- Parallel data and metadata access
- Very important for large directories

Examples: OrangeFS[8], Ceph[9], FhGFS[10].
Parallel file systems

- Parallel access from clients to (all) servers
- Parallel R/W to the same data file
- Mitigate bandwidth and latency bottlenecks
- Fields of use: HPC and high-end business applications

Examples: Lustre[7], OrangeFS[8], Ceph[9].

**Fully parallel** file systems:

- Parallel data *and* metadata access
- Very important for large directories

Examples: OrangeFS[8], Ceph[9], FhGFS[10].
Parallel file systems
High Availability

Do not confuse High Availability and Fault Tolerance:

- FT: zero downtime
- HA: small downtime (∼ min)

Data FT approaches:

- data replication (e.g. in Ceph[9])
- disk level redundancy (usually RAID 5/6)

Service HA:

- heartbeat
- pacemaker

High Availability

Do not confuse High Availability and Fault Tolerance:
- FT: zero downtime
- HA: small downtime (∼ min)

Data FT approaches:
- data replication (e.g. in Ceph[9])
- disk level redundancy (usually RAID 5/6)

Service HA:
- heartbeat
- pacemaker

High Availability

Do not confuse High Availability and Fault Tolerance:
  • FT: zero downtime
  • HA: small downtime (\(\sim\) min)

Data FT approaches:
  • data replication (e.g. in Ceph[9])
  • disk level redundancy (usually RAID 5/6)

Service HA:
  • heartbeat
  • pacemaker

HPC stuff

• Parallel solutions are highly preferred

• Infiniband[12] support
  - Lustre[7], OrangeFS[8], FhGFS[10]
  - Do not use IP over IB!

• MPI[13] I/O support
  - Usually ROMIO[14] interface
  - Lustre[7], OrangeFS[8], NFS

• Tasks optimization
POSIX compliance and FS features

POSIX was designed for local FS with serial I/O interfaces, thus it hinders parallel access.

Most common issues:

- file locks
- special files
- quota support
- acl support
- hardlinks
- mmap
- I/O usually do not follow POSIX (strictly)
Setup considerations

• Know your workload
• What POSIX features do you need?
• Is MPI needed?
• Is HA needed?
• Choose locality type
• Choose security level
Network performance

![Network performance graph](image-url)
Network performance

![Network performance graph](image-url)
OrangeFS

Procs:
- Scalable parallel FS
- Good MPI I/O support
- HA support
- Reasonable performance on large directories
- Low CPU load with high network I/O
- Configurable data distributions
- Native IB[12] support
- pNFS[5] support

Cons:
- No hardlinks or special files
- No unlink(), locks
- No quota
- Not suitable for $HOME
- Support for kernels $\geq$ 3.4 is on the way
OrangeFS

Procs:
- Scalable parallel FS
- Good MPI I/O support
- HA support
- Reasonable performance on large directories
- low CPU load with high network I/O
- configurable data distributions
- native IB[12] support
- pNFS[5] support

Cons:
- no hardlinks or special files
- no unlink(), locks
- no quota
- is not suitable for $HOME
- support for kernels $\geq$ 3.4 is on the way
OrangeFS HA support

10.0.67.104 (cluster IP) 10.67.107 (cluster IP)

GigE

pvfs2-ha1  pvfs2-ha2

crossover cable

SCSI

/dev/sdb1 /dev/sdb2

shared storage

GigE

pvfs2-ha1  pvfs2-ha2

crossover cable

SCSI

/dev/sdb1 /dev/sdb2

GigE
OrangeFS Benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Server CPU</th>
<th>Client CPU</th>
<th>I/O, MB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlusterFS</td>
<td>1.23</td>
<td>4.35</td>
<td>30</td>
</tr>
<tr>
<td>OrangeFS</td>
<td>0.11</td>
<td>0.48</td>
<td>95</td>
</tr>
</tbody>
</table>

- 15 nodes, 1 Gbit/s
- 1 : 15 servers setup
- Node: 2 x Xeon5450, 32 GB RAM, 54 MB/s HDD
Summary

• There is no universal solutions
• Understand your workload
• You’ll have very peculiar issues with any FS
• But these problems are usually solvable

• Good thing to look at for:
  • HPC: Lustre[7], OrangeFS[8], pNFS[5]
  • Data storage: Ceph[9], Lustre[7]

• Always send your patches!

Thank you for your attention!
NFS vs GFS2 vs OCFS2

Disclaimer:

Graphs aren’t mine! But they correlate well with our general experience. Our systems are in production now and old data were not saved.

Figures are taken from Giuseppe Paternò’s “Filesystem comparision: NFS, GFS2, OCFS2”[15]

Note: GFS2 is deprecated now, because only:
• up to 16 nodes are supported[16]
• up to 25 TB storage[16]
# NFS vs GFS2

## NFS vs GFS2 (generic load)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>I/O rate NFS (MB/s)</th>
<th>NFS avg transfer rate (MB/s)</th>
<th>I/O rate GFS (MB/s)</th>
<th>GFS avg transfer rate (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>21</td>
<td>2</td>
<td>43</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>6</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>6</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>0.5</td>
<td>0.1</td>
<td>41</td>
<td>8</td>
</tr>
</tbody>
</table>
GFS2 vs OCFS2

Standard tree generation

(operation timings in Seconds)
GFS2 vs OCFS2

Graph structure generation

(operation timings in Seconds)
GFS2 vs OCFS2

Change group (chgrp)

42 mins Vs 37 secs

(operation timings in Seconds)

Operation needed to share data across the working group
Bibliography

- List of file systems.
- SAN.
- GFS2.
- OCFS2.
  URL: https://oss.oracle.com/projects/ocfs2/.
- pNFS.
  URL: http://www.pnfs.com/.
- AFS.
  URL: http://www.openafs.org/.
- Lustre.
  URL: http://lustre.org/.
- OrangeFS.
  URL: http://www.orangefs.org/.
Bibliography II

CEPH. —
URL: http://ceph.com/.

FhGFS. —
URL: http://fhgfs.com/.

Top 500 list. —
URL: http://www.top500.org/.

Infiniband. —

Message Passing Interface. —

ROMIO. —

Paternò Giuseppe. —
Filesystem comparison: NFS, GFS2, OCFS2. —

Red Hat Documentation. GFS2 Overview. —