Evaluating selected cluster file systems with *Parabench*

Internship report
Authors: Marcel Krause; Jens Schlager, B.A.
Tutors: Olga Mordvinova, Julian M. Kunkel
Winter term 2009–2010

Overview

- Test Scenarios ←
- Test Patterns
- Cluster Hardware
- Benchmark Results
- Workflow Optimization
- Conclusion

Test Scenarios

- Original plan:
  Test all combinations with up to 8 nodes.

  S = {2, 4 servers} × {1, 2, 4 clients}
  × {100, 1k, 10k, 100k iterations}
  × {4 patterns} |S| = 96 tests

- Reality: availability and stability problems
  - Most of the time only 7 nodes at most
  - Test duration limits for slow file systems

Test Scenarios

- { server nodes } ∩ { client nodes } = Ø
- Configuration files:
  - OCFS: software RAID (NBD) as “servers”;
    not really servers because all logic in clients
  - GlusterFS: defaults + server/client count
  - Ceph: example + server/client count,
    all servers had data and meta data

Overview

- Test Scenarios
- Test Patterns ←
- Cluster Hardware
- Benchmark Results
- Workflow Optimization
- Conclusion

Test Patterns

- 3 patterns reflecting an OLAP (OnLine Analytic Processing) engine
  - Business Intelligence: not actually HPC, but also data intensive
- 1 synthetic pattern: Basic Operations Test (includes sequential read/write)
- All patterns written in *Parabench* language

Test Patterns

- Create index:
  - generates initial index directory structure and builds index configuration files
  - Operations: mkdir, write, rename, delete
- Delete index:
  - delete data directories, update meta data
  - Operations: delete, rmdir, write

Test Patterns

- Index index:
  - fills the created index with data
  - Operations: read, write
  - Solo part: repeat all formerly distributed operations on one single client
- Basic Operations Test:
  - write, read, append, rename, delete

Overview

- Test Scenarios
- Test Patterns
- Cluster Hardware ←
- Benchmark Results
- Workflow Optimization
- Conclusion

Cluster Hardware

Highlights:

- 2x Intel Xeon 2 GHz
- 1 GB DDR-RAM
- 80 GB IDE HDD
- 2x Gigabit Ethernet ports (one in use)
- Intel 82545EM Gigabit Ethernet controller

Special hardware on nodes 01 – 05:

- RAID controller Promise FastTrack TX2300
- RAID0 (Striping) of two 160 GB SATA-II HDDs

Theoretical Throughput

create index: write ~60 KB		
clients	aggreg.	each
1	55.5	55.5
2	111.0	55.5
4	186.3	46.6

index index: write ~4.8 KB		
clients	aggreg.	each
1	8.2	8.2
2	16.4	8.2
4	32.0	8.0

index index: read ~5.4 KB		
clients	aggreg.	each
1	9.1	9.1
2	18.3	9.1
4	35.4	8.9

- All numbers in MiB/s
- Throughput reduction: switch limit → next slide
- Index index: too few data to gain momentum

Theoretical Throughput

- Calculations based on *Performance Analysis of the PVFS2 Persistency Layer* by Julian M. Kunkel because it's the same cluster
- Assuming optimal read-ahead and maximum write buffering
- Reduction of throughput by switch limit (more network traffic than the switch can handle)

Overview

- Test Scenarios
- Test Patterns
- Cluster Hardware
- Benchmark Results ←
- Workflow Optimization
- Conclusion

Benchmark Results

- See detailed report for:
  - Throughput comparison
  - Each operation's duration
  - Each test's theoretical duration
  - Comparison by block size (basic operations)
- => Lots of precise numbers.
  - In these slides: plain and simple comparison by test duration with 4 servers and 1 client.
  - Basic Op. Test: partially estimated for 1k it.

Benchmark Results

Create	100 it.	10 k it.
OCFS	2s	3m 19s
Gluster	17s	28m 23s
Ceph	6m 46s	(~ 11h)

Delete	100 it.	10 k it.
OCFS	2s	6m 55s
Gluster	20s	32m 54s
Ceph	9m 12s	(~ 15h)

Index	100 it.	10 k it.
OCFS	4s	(~ 42m)
Gluster	38s	80m 9s
Ceph	32m 32s	(~ 2d 6h)

Index	Real it.	1k it.
OCFS	1 000	7m 21s
Gluster	100	13m 50s
Ceph	10	16m 40s

Overview

- Test Scenarios
- Test Patterns
- Cluster Hardware
- Benchmark Results
- Workflow Optimization ←
- Conclusion

Workflow Optimization

- Many tests, few variables
  - Number of servers
  - Number of clients
  - Number of iterations
- => Utilities to…
  - prepare and clean up the test environment
  - generate *Parabench* scripts from templates
  - run them, collect data, reformat for OpenOffice

File System Management Scripts

- Specific management scripts for each FS
- **`./<f>.sh start <s> <c>`**
  - Initializes the test environment for file system <f> with <s> servers and <c> clients
  - Node roles are appointed dynamically, based on the list in **`available_nodes.txt`**
  - Summarizes all their hundreds of vacuous system messages to a simple status report

File System Management Scripts

- **`./<f>.sh stop`**
  - Stops all servers and clients, based on the lists in the **`server_nodes`** and **`client_nodes`** files generated by “start”
- HTTP notification:
  - After completing their operations, the scripts can notify the tester’s web server, which in our case then sent us an XMPP instant message.

FS-specific Features

- OCFS: **`ocfs2.sh`**
  - Creates or assembles the RAID
  - Recreates or cleans OCFS2 file system (automatically guesses quicker method)
- GlusterFS: **`gluster-manager.sh`**
  - Generates and distributes the config files
  - Starts the servers in parallel, because the Gluster servers take ages to start. ☹

FS-specific Features

- Ceph basics: **`ceph-helper.sh`**
  - Modified version of Dennis Runz's start script.
- Ceph wrapper: **`manage-ceph.sh`**
  - Wrapper for **`ceph-helper.sh`**; output is filtered to prevent message flood
  - Generates and distributes the config files
  - Simplifies **`init`**, **`start`**, **`mount`**, **`umount`**, **`stop`**, and **`clean`** to just **`start`** and **`stop`**.

FS-independent Utilities

- **`paralog.pl`** (used in **`wiz.sh`**)
  - Copies *Parabench*'s output to files, like **`tee`**
  - Stops *Parabench* when it reports errors, to prevent message flood
- **`results/sumtimes.pl`** (used in **`wiz.sh`**)
  - Collects *Parabench*'s time files
  - Calculates minimum, maximum, average
  - Reformats them for copy & paste to OO Calc

FS-independent Utilities

- **`./wiz.sh <t> <i>`**
  - Prepares the test environment
  - Generates the *Parabench* script for test <t> with <i> iterations per client
  - Runs test with *Parabench* and **`paralog.pl`**
  - Runs solo part, if applicable
  - Notifies the tester's web server (again, XMPP)
  - Gathers results (**`sumtimes.pl`**)
  - Displays wall time summary

Overview

- Test Scenarios
- Test Patterns
- Cluster Hardware
- Benchmark Results
- Workflow Optimization
- Conclusion ←

Conclusion

- Ceph seems to scale almost linearly
  - but very slow for our test patterns
- GlusterFS too, and acceptable speed, but
  - “no space left o.d.” when only a few % used
- OCFS was the fastest FS, but
  - seems to have a limit on the number of files
  - non-linear scaling
- => OCFS clearly wins these tests
  - but GlusterFS might win for larger data

Thank you for listening

For sources, see the detailed report