

# XtreemFS

## Extreme cloud file system?!

Udo Seidel

# Agenda

- Background/motivation
- High level overview
- High Availability
- Security
- Summary

# Distributed file systems

- Part of shared file systems family
- Around for a while
- “back” in scope
  - Storage challenges
    - More
    - Faster
    - Cheaper
  - XaaS

# Shared file systems family

- Multiple server access the same data
- Different approaches
  - Network based, e.g. NFS, CIFS
  - Clustered
    - Shared disk, e.g. CXFS, CFS, GFS(2), OCFS2
    - Distributed, e.g. Lustre, CephFS, GlusterFS .... and XtremFS

# Distributed file systems – why?

- More efficient utilization of distributed hardware
  - Storage
  - CPU/Network
- Scalability ... capacity demands
  - Amount
  - I/O requirements

# Distributed file systems – which?

- HDFS (Hadoop)
- CephFS .. SUSE
- GlusterFS .. RedHat
- ...
- XtremFS

# History

- European Research project (2006-2010)
- Part of XtremOS
  - Linux based grid O/S
  - Member of OpenGridForum
  - Need of distributed file system

# XtreemFS and storage

- Distributed file system => distributed storage
- Object base storage approach



# Storage – looking back

- Not very intelligent
- Simple and well documented interface, e.g. SCSI standard
- Storage management outside the disks

# Storage – these days

- Storage hardware powerful => Re-define: tasks of storage hardware and attached computer
- Shift of responsibilities towards storage
  - Block allocation
  - Space management
- Storage objects instead of blocks
  - Extension of interface -> OSD standard

# Object Based Storage I

- Objects of quite general nature
  - Files
  - Partitions
- ID for each storage object
- Separation of meta data operation and storing file data
- HA not covered at all
- Object based Storage Devices

# Object Based Storage II

- OSD software implementation
  - Usual an additional layer between between computer and storage
  - Presents object-based file system to the computer
  - Use a “normal” file system to store data on the storage
  - Delivered as part of Ceph
- File systems: LUSTRE, EXOFS, CephFS

# Implementation I

- Java
  - Supported O/S
    - Linux
    - MacOS X with manual work
    - Free/Net/OpenBSD?
    - No Windows anymore
  - Server and Client (fuse)
- Non-privileged user

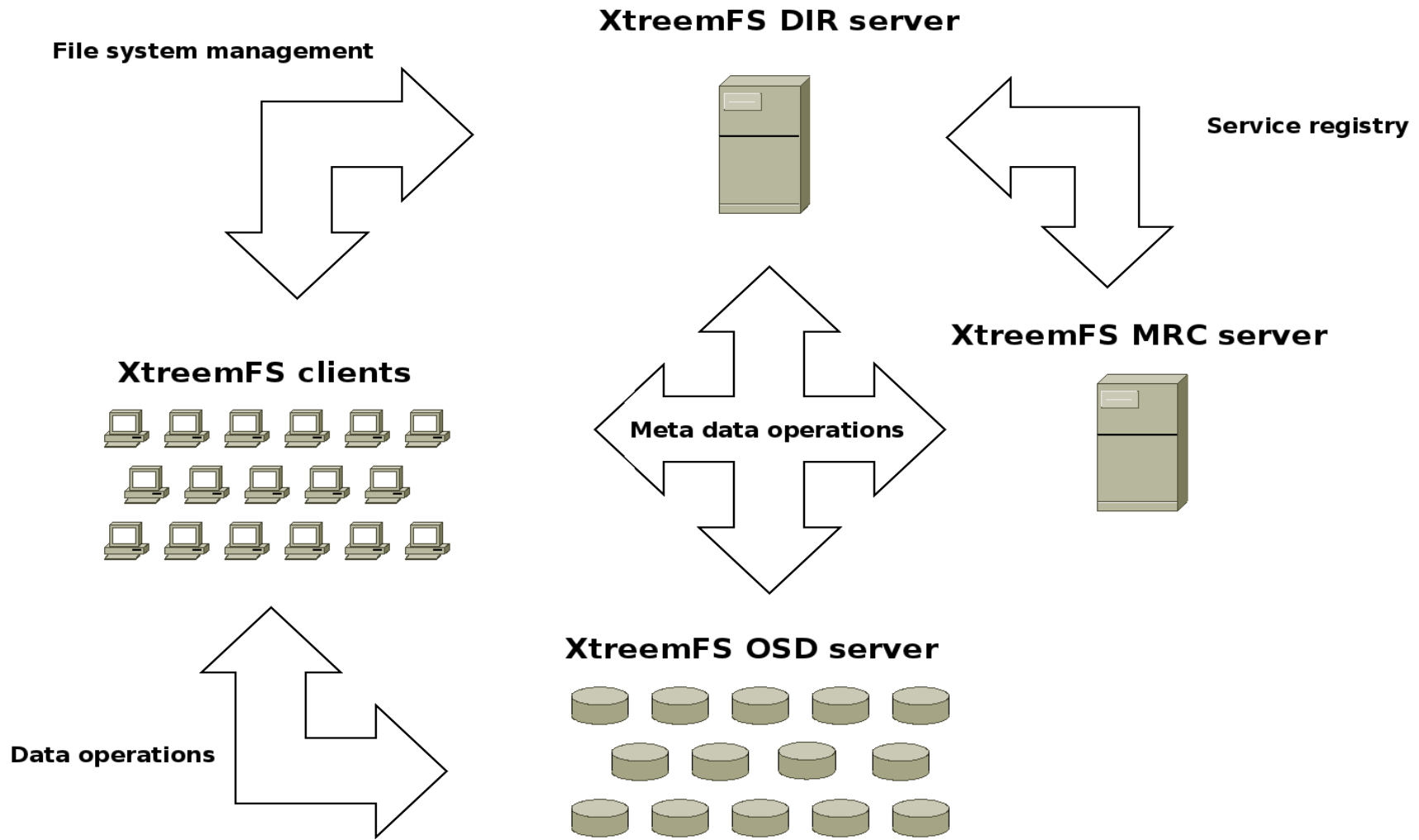
# Implementation II

- IP based
  - Different ports for DIR, MRC and OSD
  - Clear text vs. encrypted
- Object based storage
  - Software implementation
  - OSD features in XtreamFS code
    - Copy on write
    - Snapshotting

# XtreemFS – the architecture I

- 4 components
  - Object based Storage Devices
  - Meta Data and Replica Catalogue Servers
  - Directory Service
  - Clients ;-)

# XtreemFS – the architecture II





# XtreemFS services

- Several
  - OSD
  - MRC
  - Volumes
- UUID's
  - Abstraction from network
  - Change requires outage
  - Plans for topology

# XtreemFS – DIR/MRC data

- Data stored locally
  - BabuDB
  - Independent of OSD
- Write buffers

Modus	Description
ASYNC	Asynchronous log entry write
FSYNC	Fsync() called after log entry write and before ack'ing of operation
SYNC_WRITE	Synchronous log entry write, ack'ing of operation before meta data update
SYNC_WRITE_METADATA	Synchronous log entry write and meta data update before ack'ing of operation

# XtreemFS – OSD data

- File cut in 128 Kbyte pieces
- Default: entire file on one OSD
- Distribution across multiple OSD's possible
  - RAID 0 implemented
  - RAID 5 planned
  - Parallel reads/writes

# XtreemFS interfaces

- HTTP
  - Read-only
  - Read-write planned
- Command line
  - All purposes

# XtreemFS interfaces

The image displays three screenshots of the XtreemFS web interfaces, each showing a different component: Directory Service (DIR), Metadata Request Cache (MRC), and Object Store Daemon (OSD).

### XtreemFS DIR

**Version**  
 XtreamFS DIR 1.3.1.81 (Tasty Tartlet)  
 RPC 10001  
 Interface  
 Database 0.5.6

**Configuration**  
 TCP port 32638  
 Debug Level 6

**Load**  
 # client connections 3  
 # pending requests 0

**Transfer**  
 # requests processed 0

**VM Info / Memory**

8192:	poolSize = 6	numRequests
65536:	poolSize = 4	numRequests
131072:	poolSize = 0	numRequests
524288:	poolSize = 0	numRequests
2097152:	poolSize = 0	numRequests
unpooled (> 2097152):		numRequests = creates

**Time**  
 global  
 XtreamFS Tue Jan 24 19:47:08 CET 2012 (1327430828673)  
 time

**Database Dump**

**Address Mapping**

UUID	mapping
047a1d96-b6fc-4fb3-9463-bbd9545cdd5	pbrpcs://192.168.1.210:32636
8368e11b-e031-4c8b-9de4-02fd5ce3e150	pbrpcs://192.168.1.212:32640 pbrpcu://192.168.1.212:32640
c9f286cf-507c-4e9e-aa55-faeb2f87e83a	pbrpcs://192.168.1.211:32640 pbrpcu://192.168.1.211:32640

### XtreemFS MRC @ 047a1d96-b6fc-4fb3-9463-bbd9545c

**Version**  
 XtreamFS MRC 1.3.1.81 (Tasty Tartlet)  
 RPC Interface 20001  
 Database 0.5.6

**Configuration**  
 TCP & UDP port 32636  
 Directory Service pbrpcs://testvm1:32638  
 Debug Level 6

**Load**  
 # client connections 0  
 # pending client requests 0  
 Processing Stage queue length

**Requests**

'getattr'	32
'listxattr'	3
'open'	2
'readdir'	4
'statvfs'	2
'unlink'	1
'access'	16
'xtreamfs_renew_capability'	8
'xtreamfs_update_file_size'	2

**Volumes**

selectable OSDs	c9f286cf-507
striping policy	STRIPING_P
access policy	ACCESS_C
osd policy	1000,3002
replica policy	
files	3
#directories	1
free disk space:	14.27 GB
occupied disk space:	0 bytes

**VM Info / Memory**

Memory free/max/total	1.81 MB / 117.94 MB / 6.44 MB
-----------------------	-------------------------------

**Buffer Pool stats**

8192:	poolSize =
65536:	poolSize =
131072:	poolSize =
524288:	poolSize =
2097152:	poolSize =
unpooled (> 2097152):	numRequests =

**Time**

### XtreemFS OSD @ c9f286cf-507c-4e9e-aa55-faeb2f87e83a

**Version**  
 XtreamFS OSD 1.3.1.81 (Tasty Tartlet)  
 RPC Interface 30001

**Configuration**  
 TCP & UDP port 32640  
 Directory Service pbrpcs://testvm1:32638  
 Debug Level 6  
 Statistics

**Load**  
 # client connections 0  
 # pending client requests 0  
 Preproc Stage queue length  
 Storage Stage queue length  
 Deletion Stage queue length  
 Open files 0

**Transfer**  
 # object written 1  
 # object read 0  
 bytes sent 0 bytes  
 bytes received 128.00 kB  
 # files deleted 1  
 # replicated object written 0  
 bytes replicated 0 bytes

**VM Info / Memory**

Free Disk Space	7.12 GB
Memory free/max/total	3.04 MB / 117.94 MB / 8.81 MB

**Buffer Pool stats**

8192:	poolSize = 5	numRequests = 26648	creates = 5
65536:	poolSize = 4	numRequests = 12	creates = 8
131072:	poolSize = 1	numRequests = 1	creates = 1
524288:	poolSize = 0	numRequests = 0	creates = 0
2097152:	poolSize = 0	numRequests = 0	creates = 0
unpooled (> 2097152):		numRequests = creates =	0 deletes = 0

**Time**  
 global XtreamFS time Tue Jan 24 19:47:23 CET 2012 (1327430843743)  
 resync interval for global 60000 ms  
 time  
 local system time Tue Jan 24 19:47:23 CET 2012 (1327430843740)  
 local time update interval 50 ms

**UUID Mapping Cache**

# XtreemFS – high level summary

- Multi-platform
- Abstraction via UUID
- Communication separation
- Freedom of choice of OSD backend file system
- HPC out of scope

# XtreemFS – HA in general

- One part: OSD
  - Replication via policies
- Other part: MRC and DIR
  - Local data stored in BabuDB's
  - Synchronization via BabuDB methods

# XtreemFS – HA for MRC/DIR

- Master/slave
  - Master changes -> log file without buffering
  - Log file entries propagation to slaves
  - Quorum needed => at least 3 instances
  - No automation for DIR
- Synchronization
  - in clear text
  - Encryption via SSL possible



# XtreemFS OSD replication

- File replication
  - Read-only
    - Since 1.0
    - Easy to handle
  - Read-write
    - Only since 1.3
    - Later more
- Copies
  - Full
  - Partial aka on-demand

# XtreemFS r/o replication

- Arbitrary amount of replicas
- Equally treated replicas
- Only OSD local access
- No sync needed
- Use case
  - Static files :-)
  - Low bandwidth (partial replica)
  - Big static files (partial replica)

# XtreemFS r/w replication

- Primary/secondary
- Election on demand with leases
- Read/write access
  - First primary
  - Propagated to secondaries

# XtreemFS r/w replication - failure

- Secondary
  - Behaviour configurable
  - Write failure vs. Write on remaining
    - Quorum needed
- Primary
  - Behaviour configurable
  - Write failure vs. Write on remaining
    - Quorum needed

# XtreemFS OSD/replica policies

- OSD selection for new files
- Replica selection for new/additional copies
- Categories: filter, group, sort
- Combination of rules

Policy	Category
Standard OSD	filter
FQDN based	filter, group, sort
UUID based	filter
Data center topology	group, sort
random	sort

# XtreemFS HA summary

- Homework needed for DIR and MRC
- OSD
  - Lateness of OSD read-write replication
  - OSD Read-only replication
    - Mature and WAN ready
  - Access time improvement via striping
  - Flexibility of policies

# XtreemFS encryption

- Not on file system level
- For communication
  - Interaction of DIR, MRC and OSD
  - Data replication for HA for DIR and/or MRC

# XtremFS channel encryption

- Via SSL
  - PCKS#12 or Java Key Store (JKS)
  - Locally stored
    - service/client certificates
    - root CA certificates
- Two modes
  - All-Or-Nothing approach
  - Grid-SSL
    - just authentication



# XtreemFS secure channel encryption

- Password protection of certificates
  - MRC/DIR/OSD: stored service configuration
  - Client: via CLI!!

```

Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe
root 21042 0.0 0.0 0 0 ? S Jan22 0:00 [flush-btrfs-4]
root 21477 0.0 0.0 0 0 ? S 11:01 0:02 [kworker/0:0]
root 21679 0.0 0.0 0 0 ? S 20:01 0:00 [kworker/0:2]
root 21701 0.0 0.0 0 0 ? S 21:01 0:00 [kworker/0:1]
root 21916 0.1 1.8 125908 4508 ? S 21:02 0:00 sshd: root@pts/0
root 21918 0.2 1.5 118300 3760 pts/0 Ss 21:02 0:00 -bash
root 21972 0.0 1.5 590624 3860 ? Ssl 21:02 0:00 mount.xtreemfs --pkcs12-file-path=/etc/xos/xtreem
fs/truststore/certs/client.p12 --pkcs12-passphrase=passphrase pbrpcs://testvm1/TEST /xtfs
root 21987 0.0 0.4 115688 1140 pts/0 R+ 21:03 0:00 ps auxww
$ █
```

# XtreemFS encryption summary

- Data encryption on POSIX layer?
- SSL obvious choice for TCP/IP channels
  - Missing PKI contradicts scalability
  - Password protection needs re-design

# Summary

- High self-defined goals
  - Some dropped?
  - Some partially implemented
- Ok for R&D Labs
  - HA and housekeeping improvement needed
  - Encryption w/o PKI

# References

- <http://www.xtreemfs.org>
- <http://babudb.googlecode.com>

Thank you!