# 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 XtreemFS

# 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
- ...
- XtreemFS

# History

- European Research project (2006-2010)
- Part of XtreemOS
  - 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

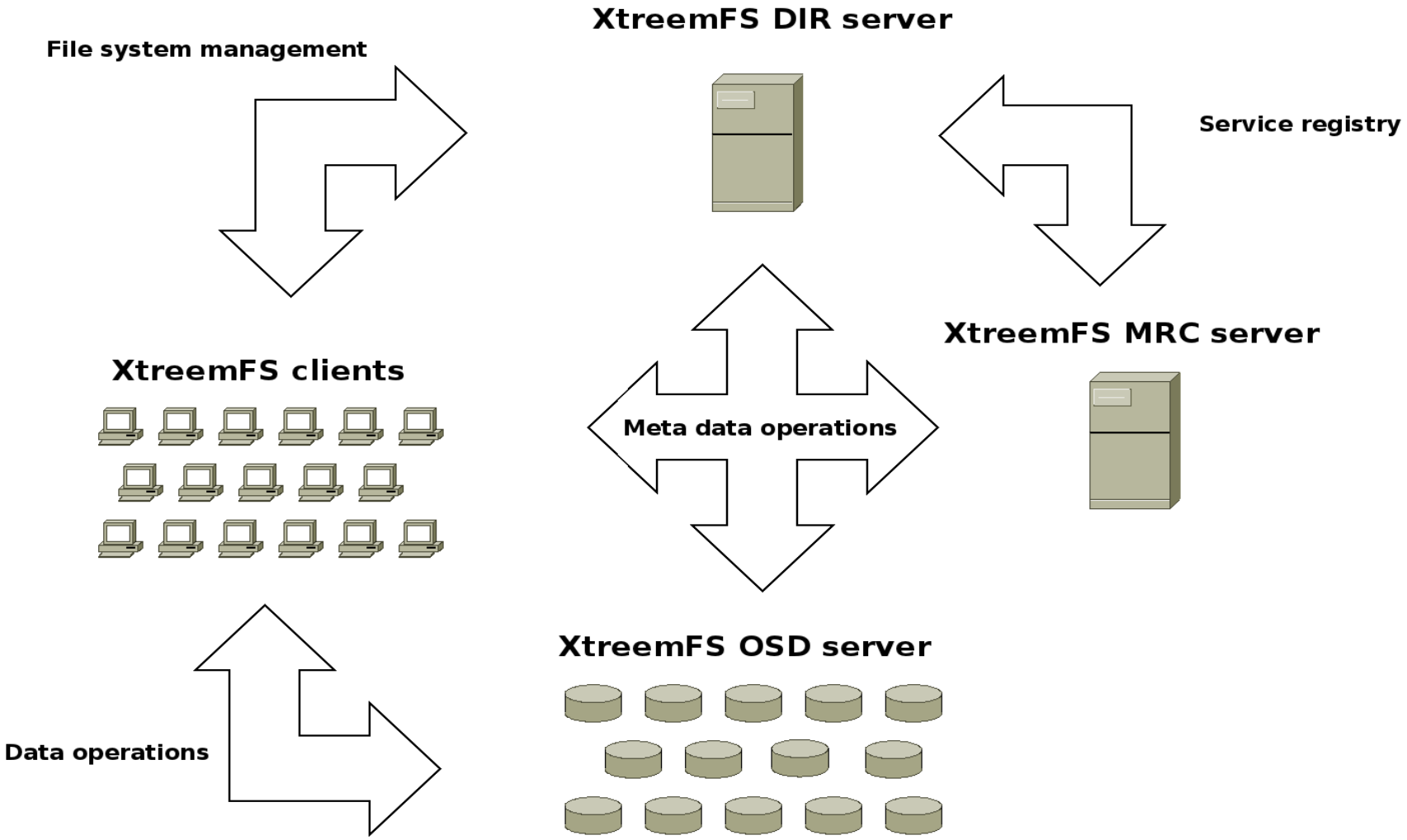
# Implementation II

- IP based
  - Different ports for DIR, MRC and OSD
  - Clear text vs. encrypted
- Object based storage
  - Software implementation
  - OSD features in XtreemFS 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 |

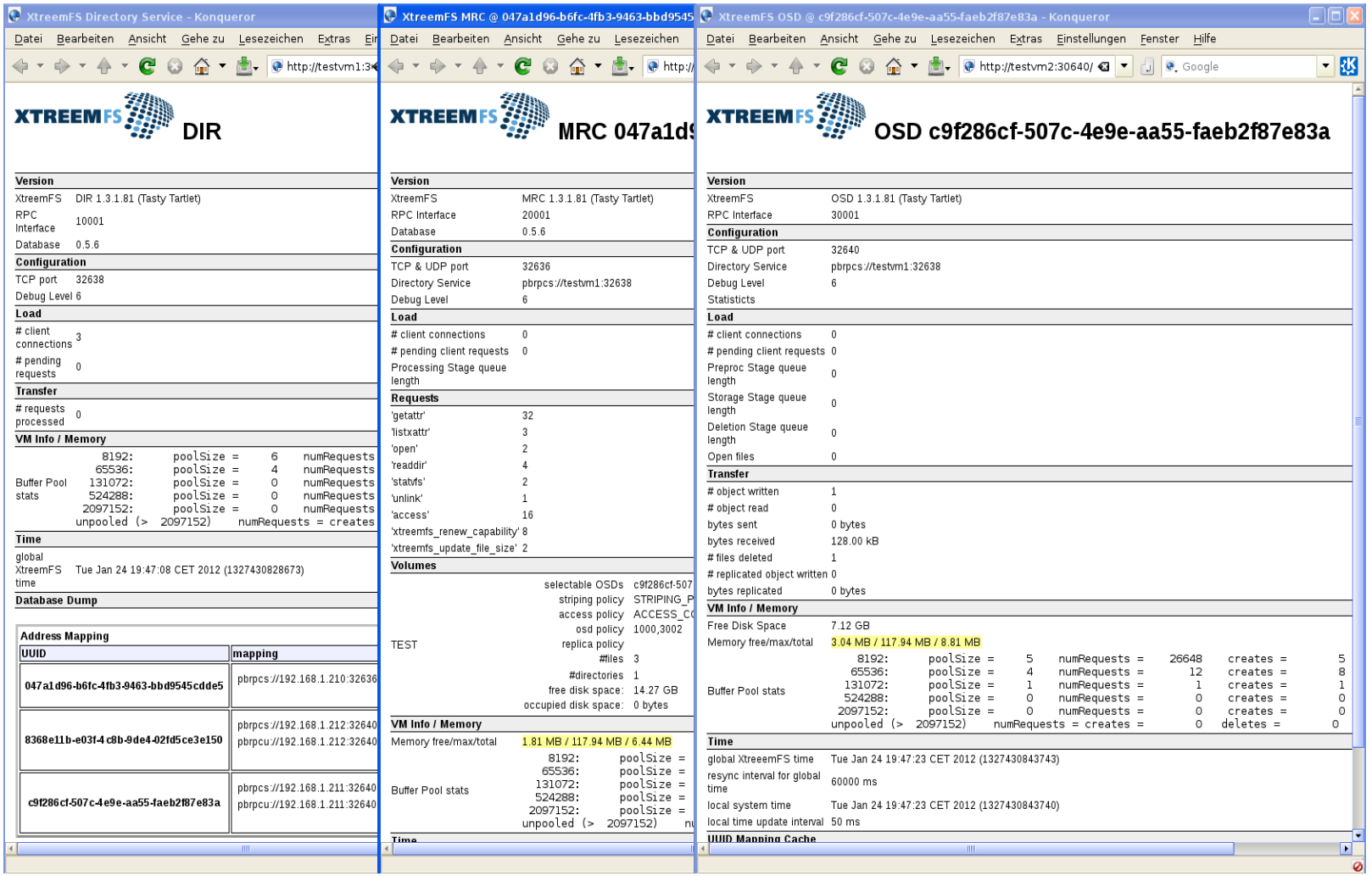
# XtreemFS – OSD data

- 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



# 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

# XtreemFS 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!