XtreemFS — a Distributed File System for Grids and Clouds

Jan Stender
Zuse Institute Berlin

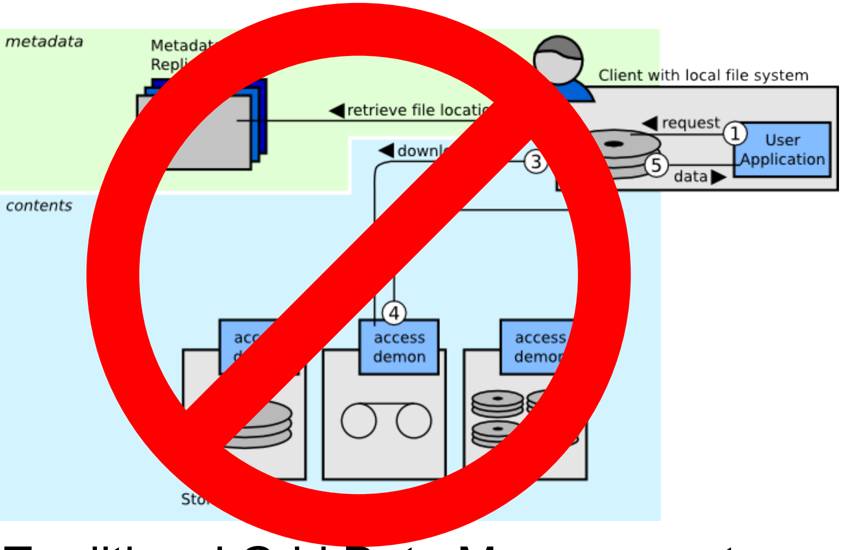
The XtreemOS Project

- Research project funded by the Euopean Comission
- 19 partners from Europe and China
- XtreemFS is the data management component
  - developed by ZIB, NEC HPC Europe, Barcelona Supercomputing Center and ICAR-CNR Italien
  - first public release in August 2008
  - current version 1.2.2

What is XtreemFS

- a **distributed** ...
  - clients, servers distributed world wide
  - mount volumes anywhere (even on a plane)
- ... and **replicated** ...
  - replicate files across data-centers for availability and locality
  - reduce latency and bandwidth consumption
- ... **POSIX** compliant file system
  - regular file system interface and semantics
  - simple to use, no need to modify applications

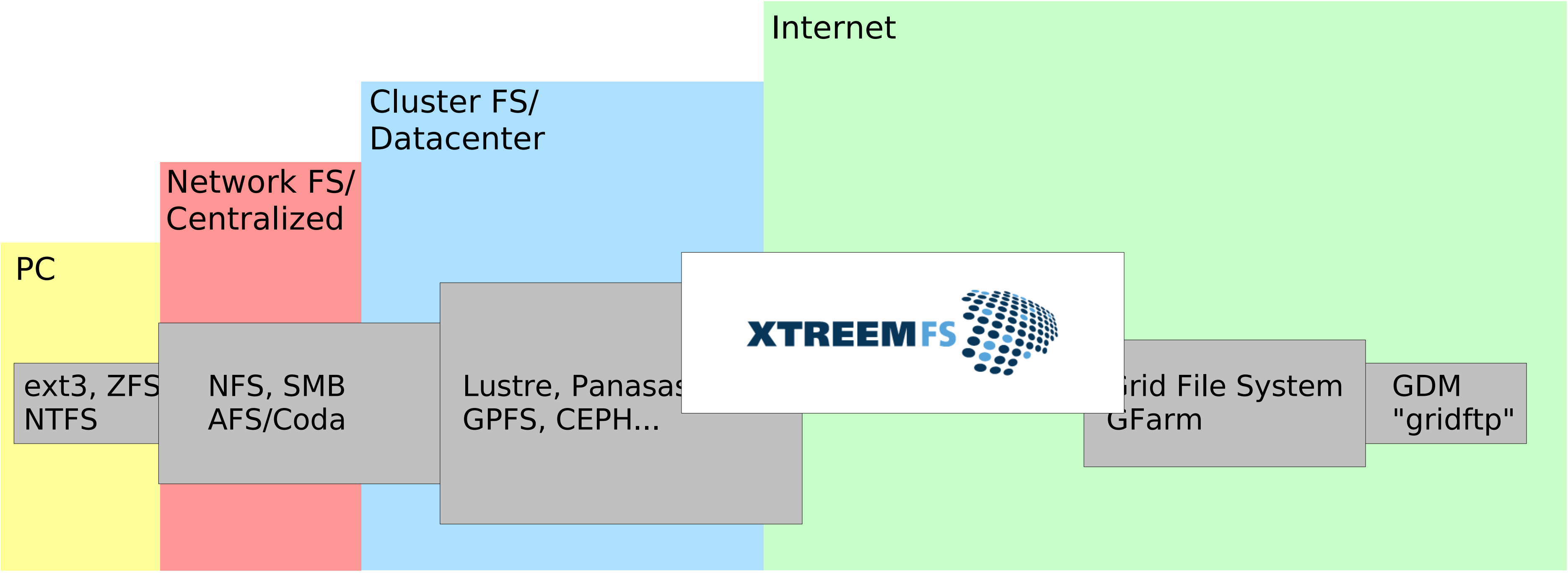
XtreemFS vs. Traditional Grid Data Management



Traditional Grid Data Management

- POSIX semantics
  - not just POSIX interface!
  - support legacy apps, not limited to write-once
  - transparent replication, remote access
- All access through XtreemFS
  - no local copies (consistency, security)
- Partial replicas
  - fetch only data used by apps
  - avoid bandwidth-peak at start-up

File System Landscape



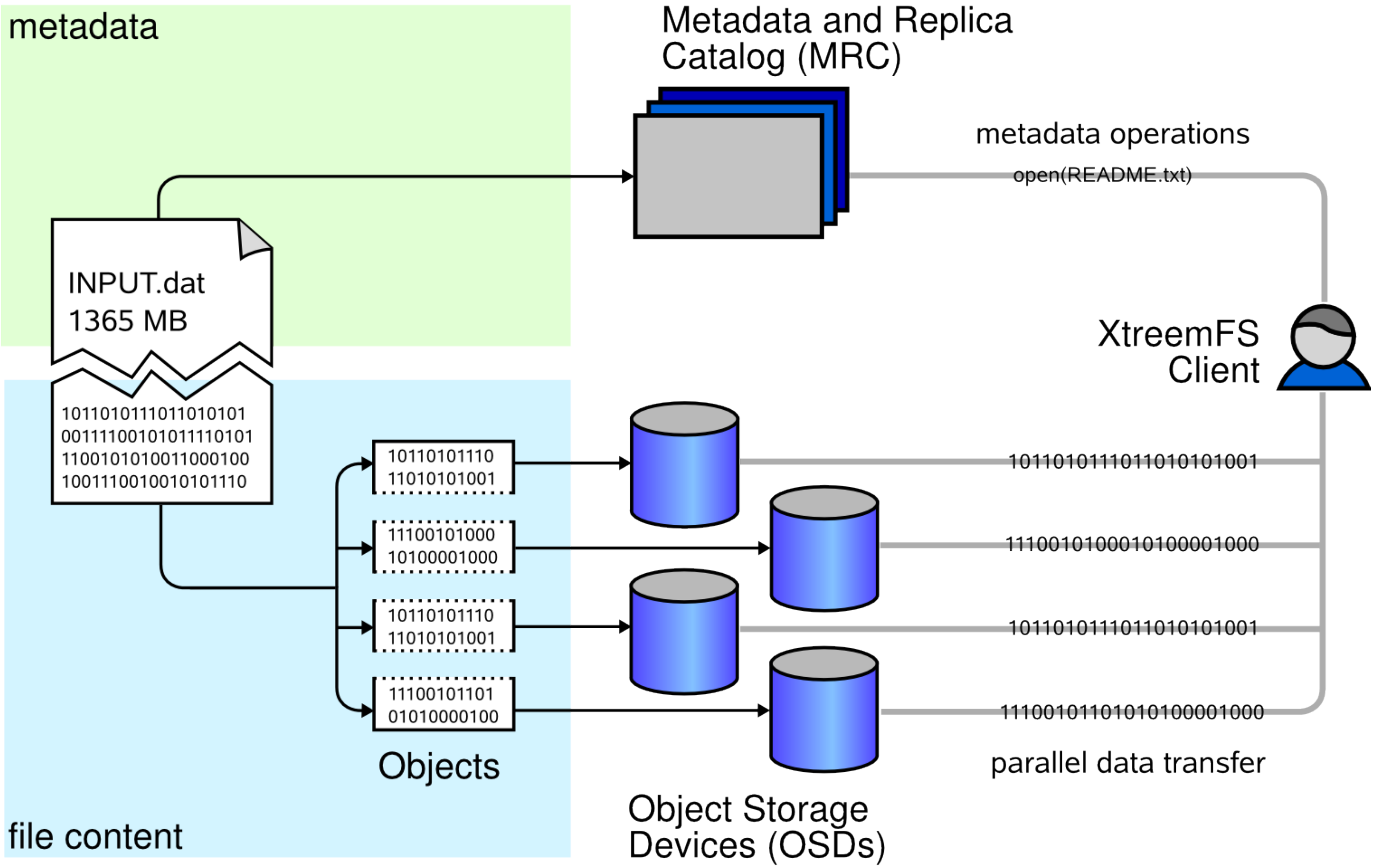
Outline

1. XtreemFS Architecture
2. XtreemFS Features
    1. Striping
    2. Replication
3. Metadata Management
    1. BabuDB
4. Development
    1. Current state
    2. Outlook

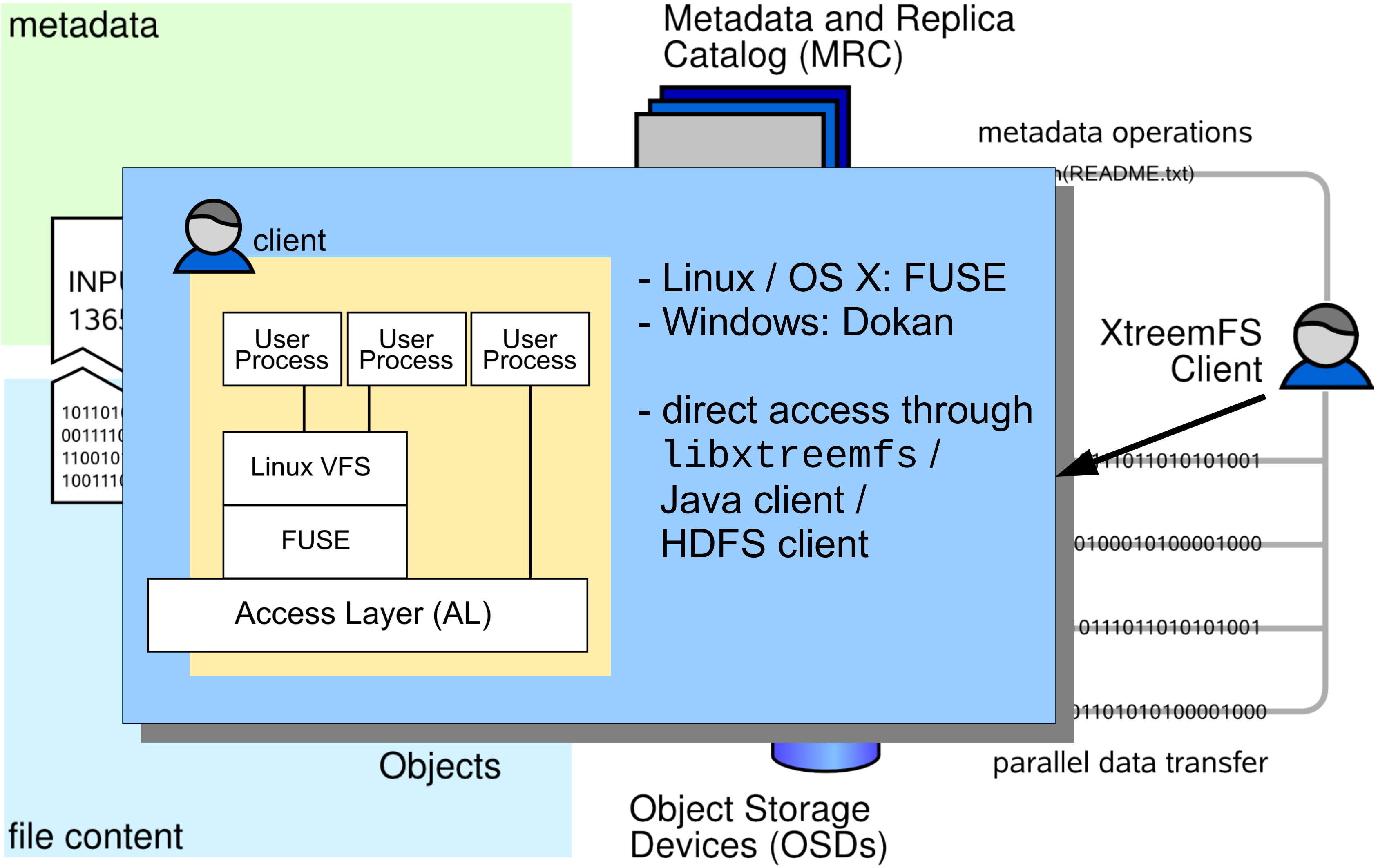
Outline

1. **XtreemFS Architecture**
2. XtreemFS Features
    1. Striping
    2. Replication
3. Metadata Management
    1. BabuDB
4. Development
    1. Current state
    2. Outlook

XtreemFS Architecture



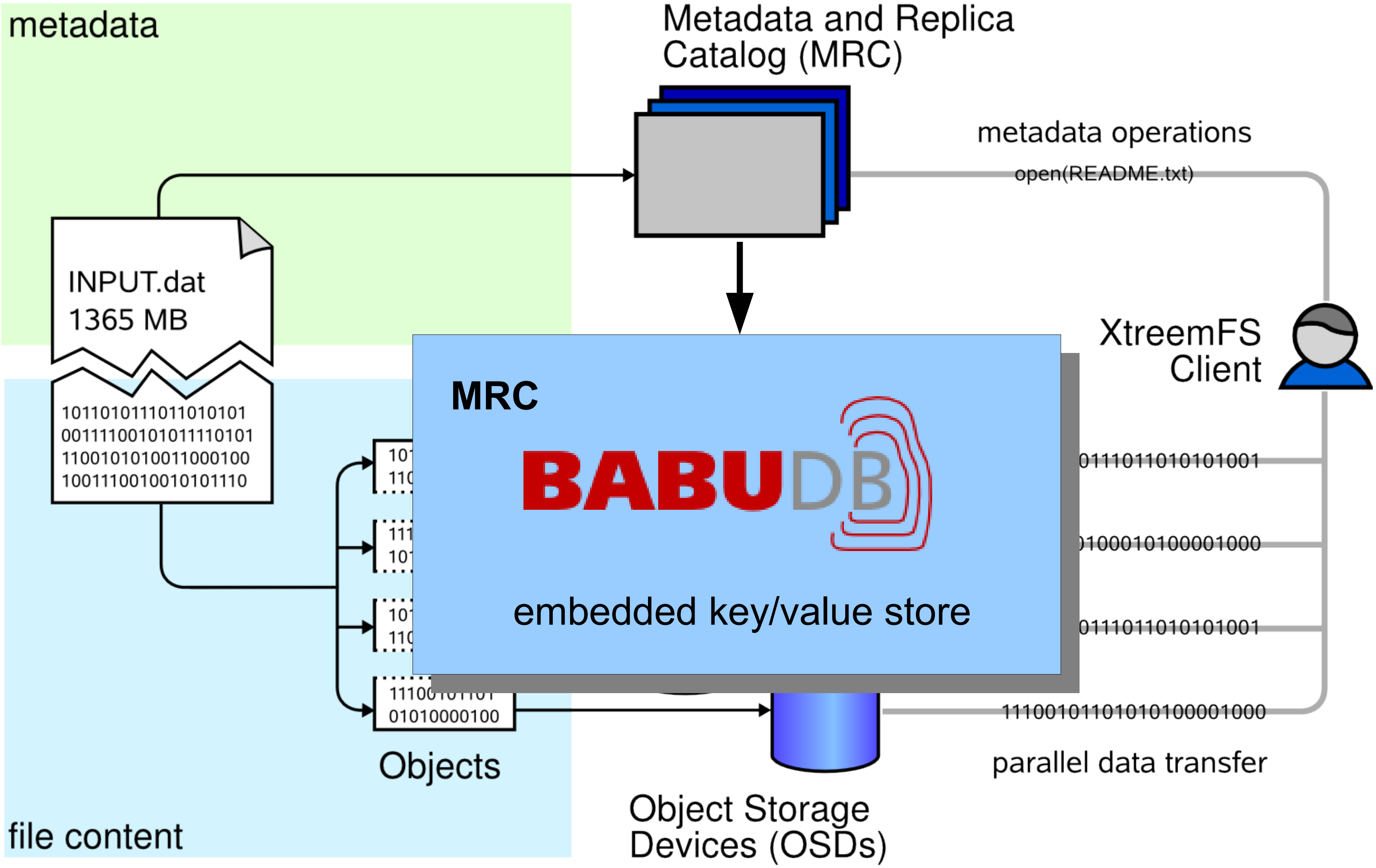
XtreemFS Architecture



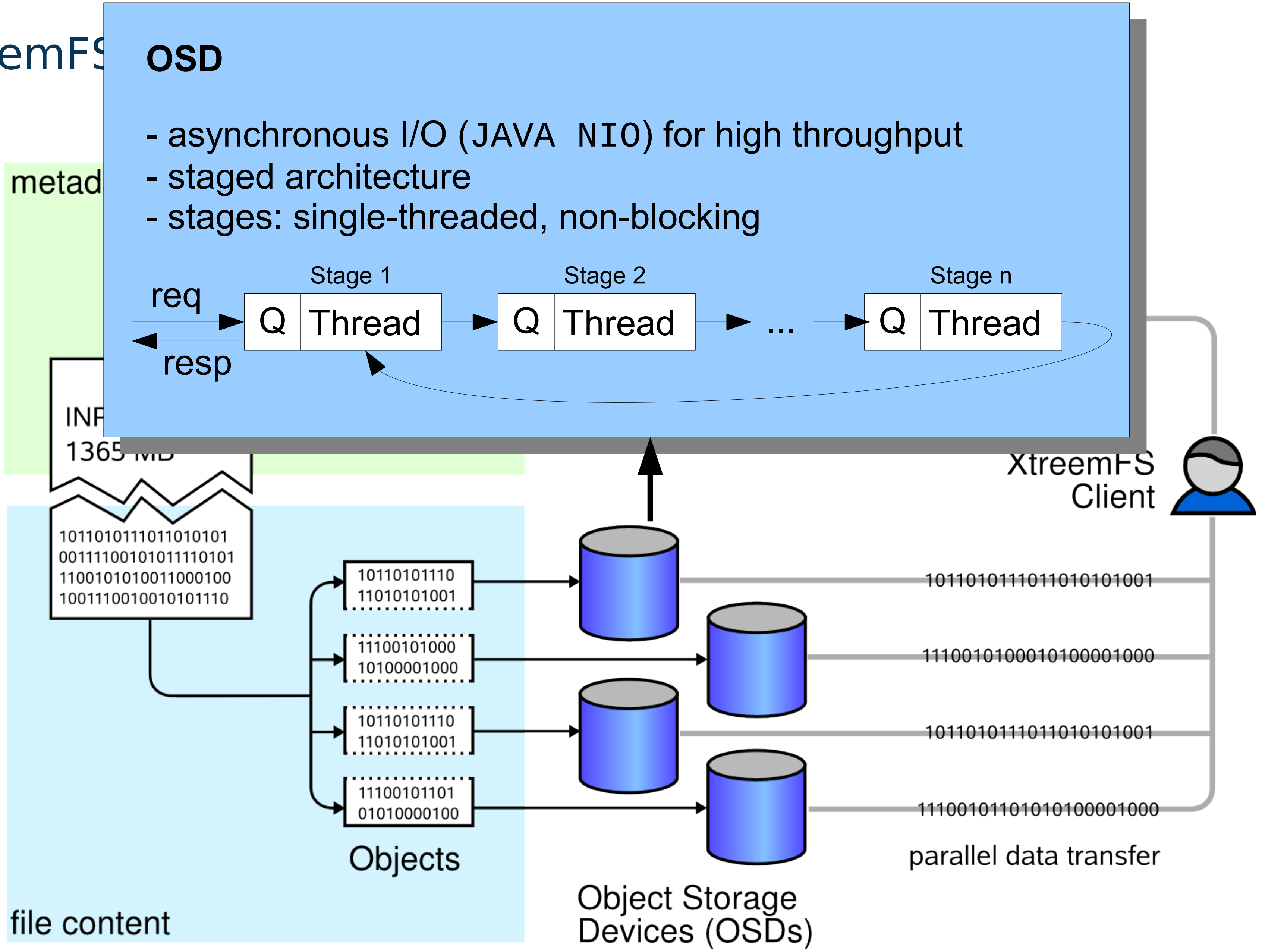
- Linux / OS X: FUSE
- Windows: Dokan

- direct access through `libxtreemfs` / Java client / HDFS client

XtreemFS Architecture



XtreemFS



OSD

- asynchronous I/O (`JAVA NIO`) for high throughput
- staged architecture
- stages: single-threaded, non-blocking

Outline

1. XtreemFS Architecture
2. **XtreemFS Features**
    1. **Striping**
    2. **Replication**
3. Metadata Management
    1. BabuDB
4. Development
    1. Current state
    2. Outlook

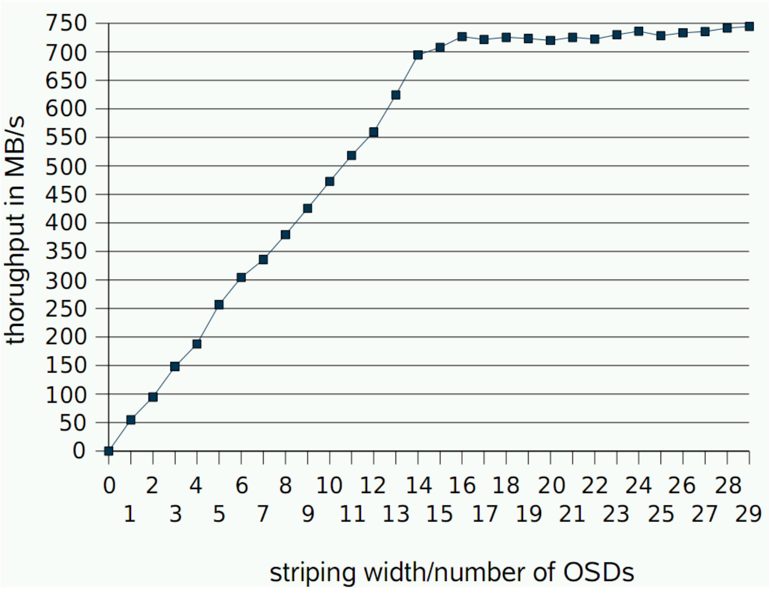
Features

- POSIX compatibility
  - interface and semantics
- Striping (parallel I/O)
- Transparent replication
  - read-only
  - read/write (sequential consistency)
  - partial replicas
- SSL & X.509 support
- Checksums
- Extensions / plug-ins

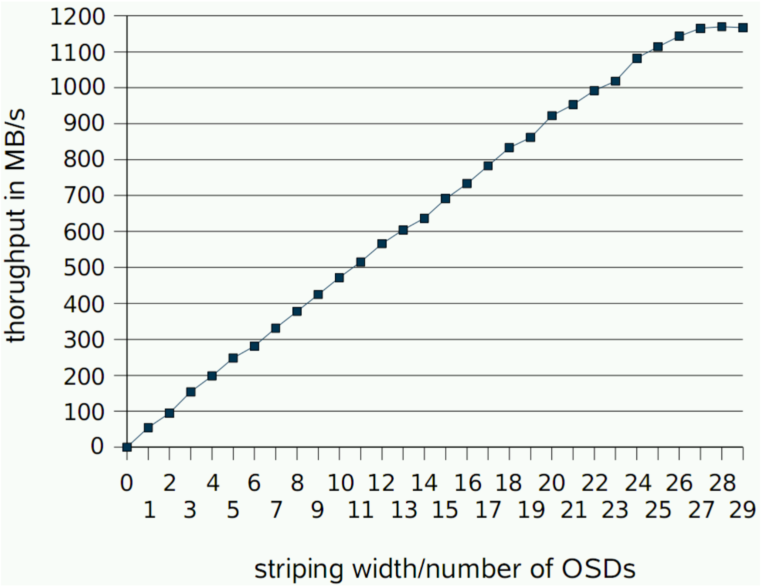
Features: Striping

- Striping
  - parallel transfer from/to many OSDs in a cluster
  - bandwidth scales with the number of OSDs
  - supports RAID0

READ



WRITE

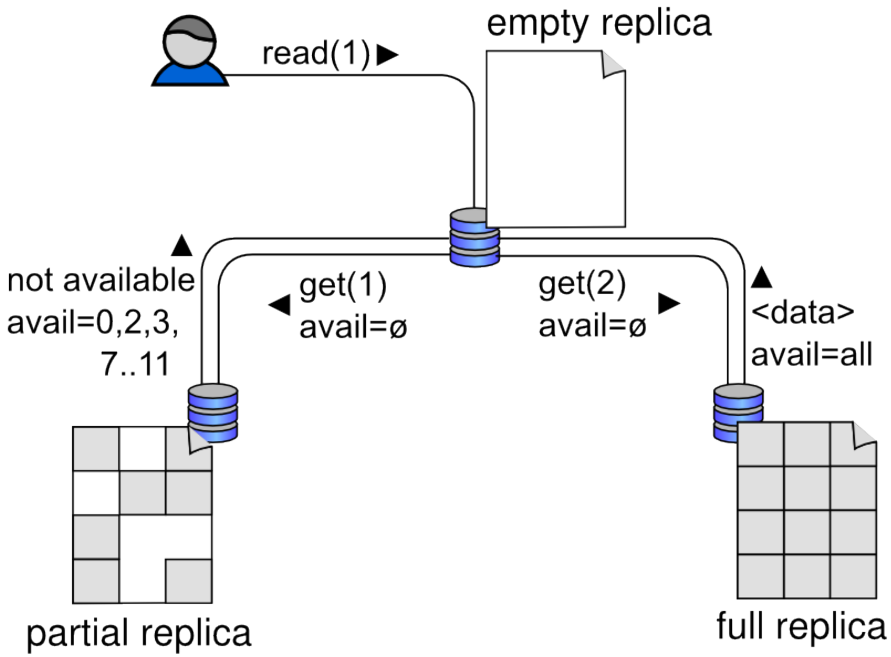


Features: Replication

- Transparent to applications and users (server-driven)
- »Read-only« Replication
  - fast and efficient distribution of files over many OSDs
  - suitable for Grid and caching
- »Read/Write« Replication
  - sequential consistency of replicas (POSIX compliant)
  - master/slave replication with automatic fail-over

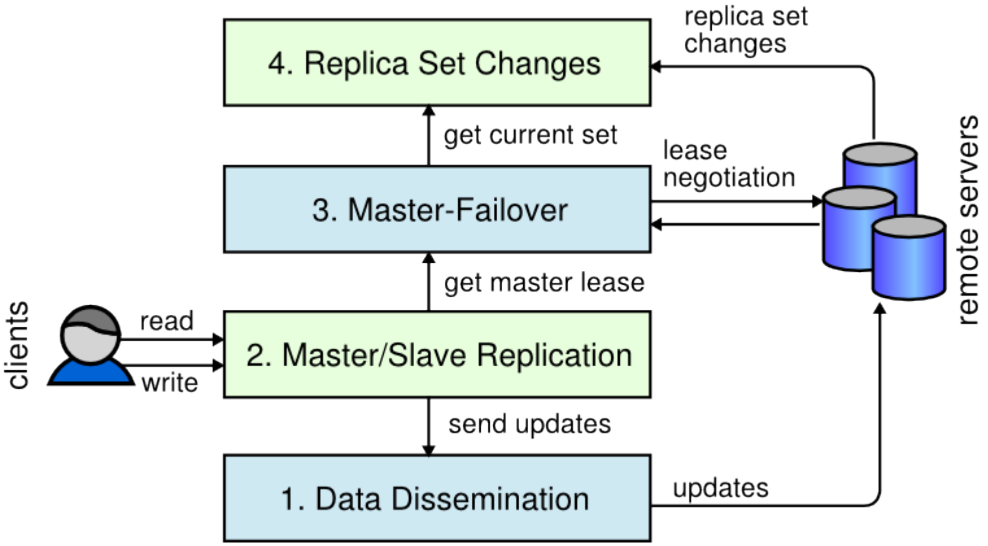
»Read-only« Replication

- Transfer strategies
  (some ideas borrowed from p2p)
  - OSDs exchange "object lists"
  - fetch objects
    - in order
    - rarest first
  - select OSDs
    - according to object lists
    - bandwidth
    - replica selection mechanisms
      (network coordinates, datacenter map)
- Prefetching (for partial replicas)
- Client requests are always served first

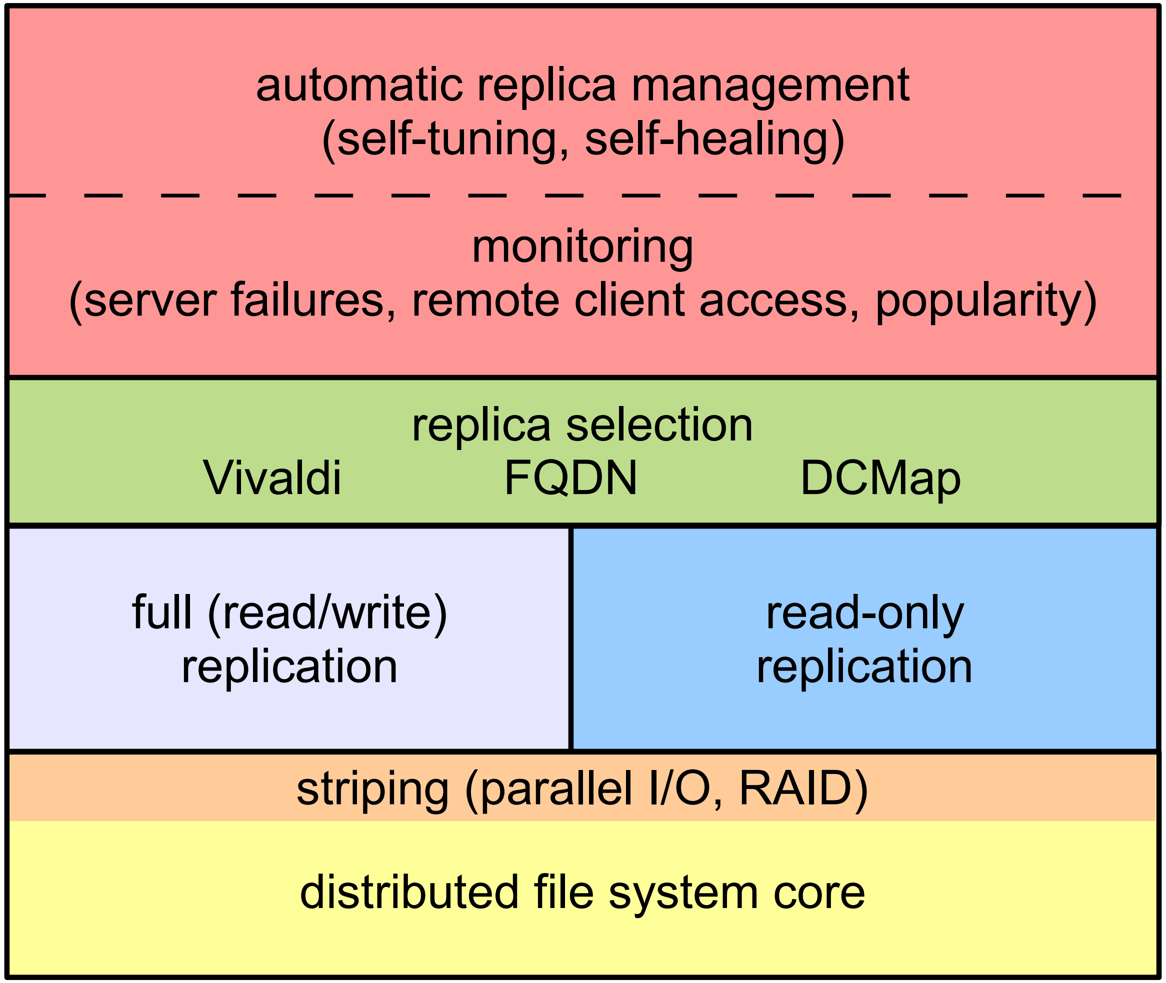


»Read-write« Replication

- Master/slave scheme
  - master defines order on updates
- Automatic fail-over w/ leases
  - master acquires lease
  - lease expires at a certain point in time
- Lease negotiation algorithm: **Flease**



Replication Architecture



automatic replica management
(self-tuning, self-healing)

monitoring
(server failures, remote client access, popularity)

replica selection
Vivaldi FQDN DCMap

full (read/write)
replication

read-only
replication

striping (parallel I/O, RAID)

distributed file system core

Outline

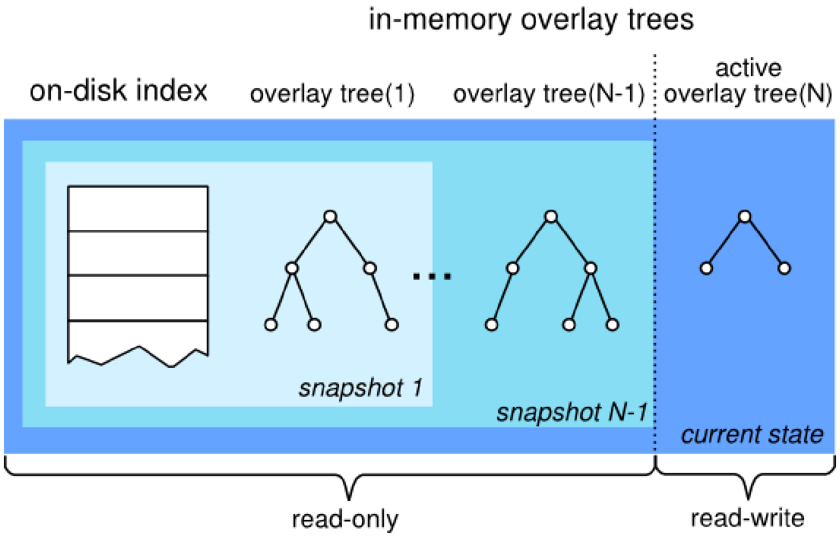
1. XtreemFS Architecture
2. XtreemFS Features
    1. Striping
    2. Replication
3. **Metadata Management**
    1. **BabuDB**
4. Development
    1. Current state
    2. Outlook

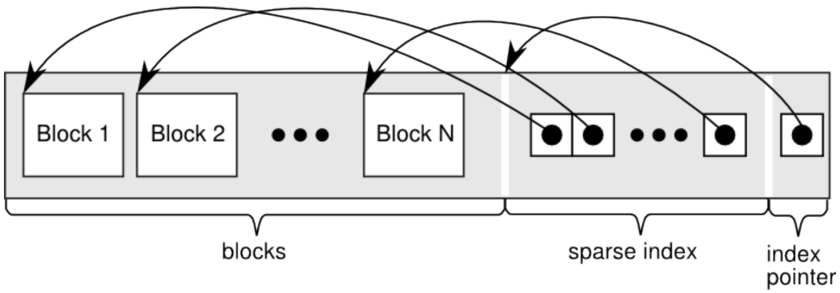
Metadata Management

- Metadata stored in database
  - exchangeable storage backends
- BabuDB: storage backend based on LSM-trees
  - key-value store, non-transactional
  - optimized for MRC and file system workloads
  - asynchronous checkpoints and snapshots
  - short recovery and start-up times
  - thousands of file creates/s, tens of thousands of `stat` requests/s

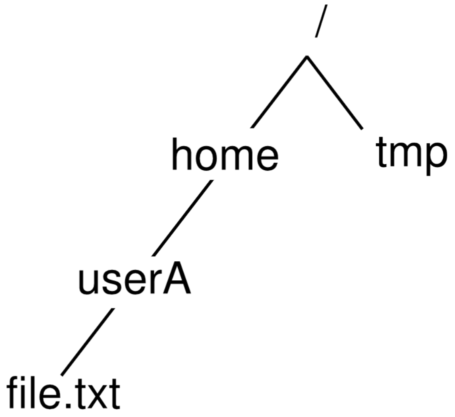
Metadata Management: BabuDB

Index



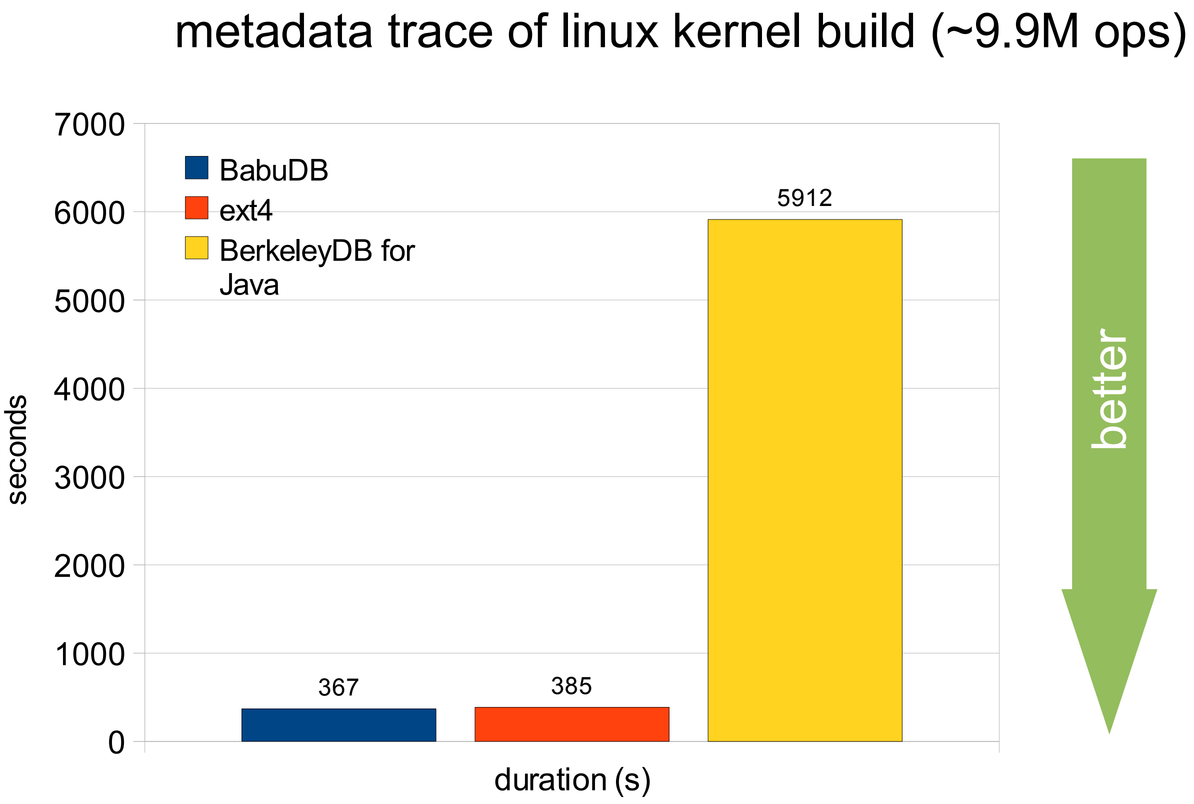


Mapping



key	value
0,/,1	atime=2009-01-01 12:00 CET...
0,/,2	ID=1,perm=rwxr-x---...
0,/,3	empty
1,home,1	atime=2009-01-01 12:00 CET...
1,home,2	ID=2,perm=rwxr-x---...
1,home,3	*empty*
1,tmp,1	atime=2008-10-21 05:21 CET...
1,tmp,2	ID=3,perm=rwxrwx---...
1,tmp,3	*empty*
2,userA,1	atime=2009-01-01 12:00 CET...
2,userA,2	ID=4,perm=rwx------...
2,userA,3	*empty*
4,file.txt,1	atime=2008-10-05 23:49 CET...
4,file.txt,2	ID=5,perm=rwx------...
4,file.txt,3	*empty*

Metadata Management: BabuDB Performance



Outline

1. XtreemFS Architecture
2. XtreemFS Features
    1. Striping
    2. Replication
3. Metadata Management
    1. BabuDB
4. **Development**
    1. **Current state**
    2. **Outlook**

Current State: Facts and Figures

- Current release: XtreemFS 1.2.2
- 3 core developers, 2 students
- ~3.5 years of development
- ~100k LOC (Java servers & C++ client)
- ~75 subscribers to support mailing list
- ~20 active users (survey result)

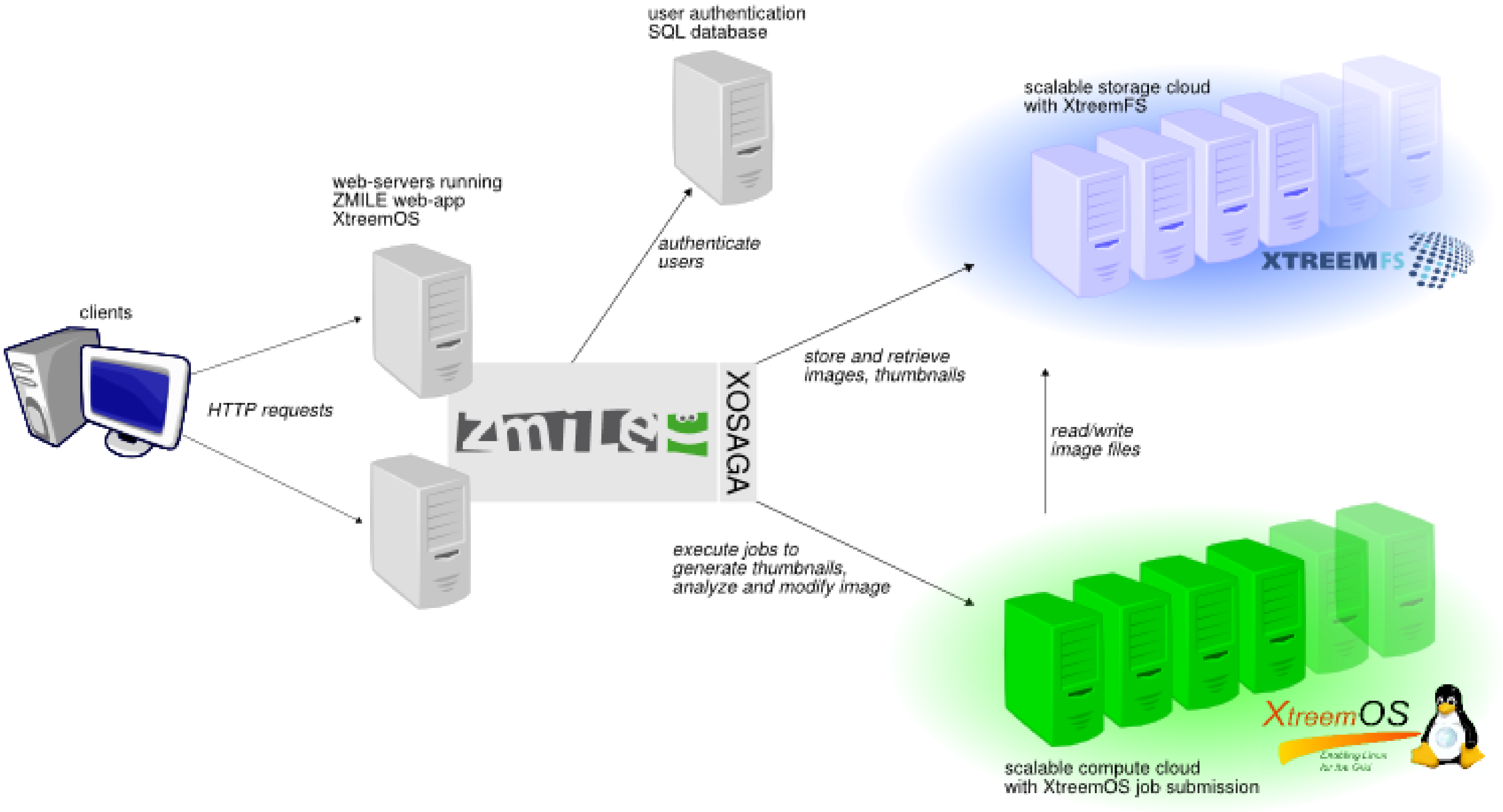
Outlook: Future Development

- No-SPOF – replication of all services
- Automatic replica management
  - replica creation, deletion, replacement, factor
- Backups and consistent snapshots
- NFSv4/WebDAV exporters
- Federation support

How to get involved?

- Open source project (GPL/BSD) at xtreemfs.googlecode.com
- Mailing Lists xtreemfs@googlegroups.com
- IRC Channel #xtreemos-dev at freenode

zmile: an XtreemOS / XtreemFS Demonstrator



http://www.zmile.eu