

XTREEMFS



**an Object-Based File System for
Large-Scale Federated IT
Infrastructures**

Jan Stender,
Zuse Institute Berlin



HPC File Systems: From Cluster To Grid
October 3-4, 2007

In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

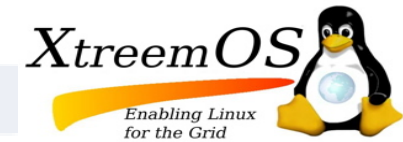
Features

Implementation

Current State & Plans



The XtremOS EU Project



- **XtremFS** is part of the **XtremOS** project
- **EU project, 18 partners** from all over Europe, incl. NEC, SAP, Telefonica, Mandriva, Red Flag Linux
- Develops a distributed operating system around Kerrighed, a single system image Linux kernel
- The **XtremFS Team**:
 - Zuse Institute Berlin
 - Barcelona Supercomputing Center
 - NEC High Performance Computing, Stuttgart
 - CNR, Pisa, Italy
 - Universität Düsseldorf
 - SAP Research



In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

Features

Implementation

Current State & Plans



Object-based File Systems

Block-based File Systems:

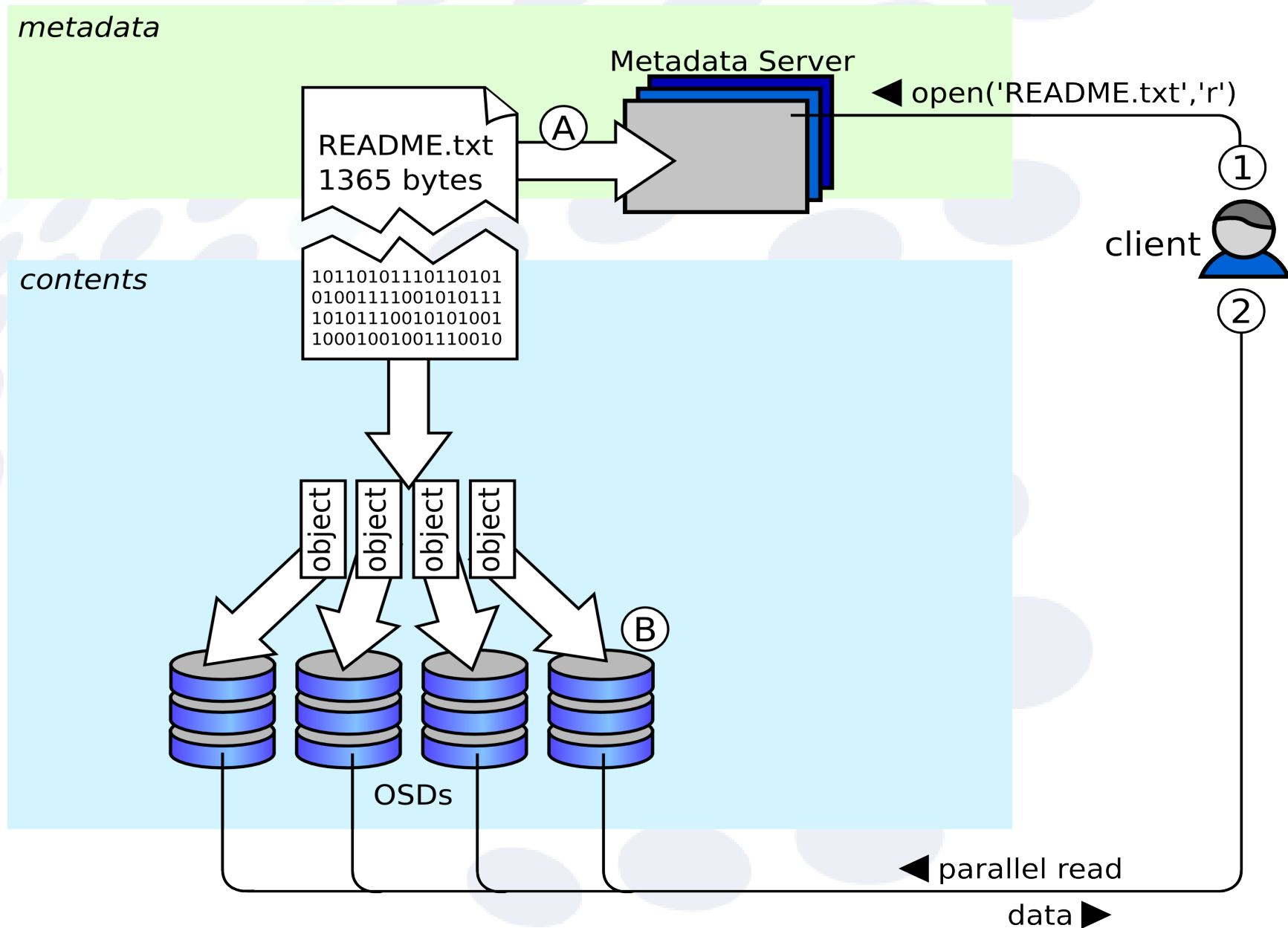
- Unit of distribution are disk blocks
- File system addresses blocks over the network
- Metadata and block-management at central server

Object-based File Systems:

- Storage devices can be more intelligent today
- Split file in parts and distribute & address them
- Only metadata at server, block management by storage devices



Object-based File Systems



Object-based File Systems

several available ...

- Lustre (Open-Source)
- Panasas ActiveStore (commercial)
- Ceph (Research, Open-Source)

common properties:

- parallel designs for high-performance LAN access
- centralized, one data center, one organization
- control over failures of hardware



In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

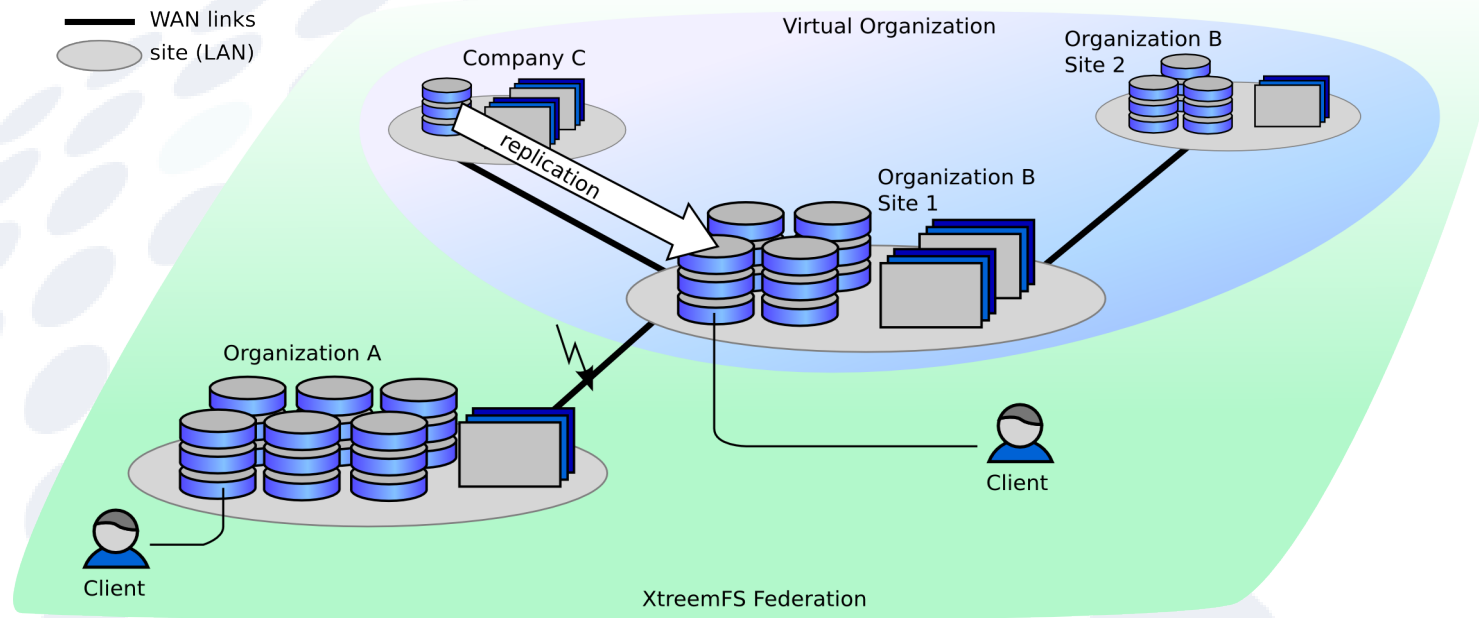
Features

Implementation

Current State & Plans



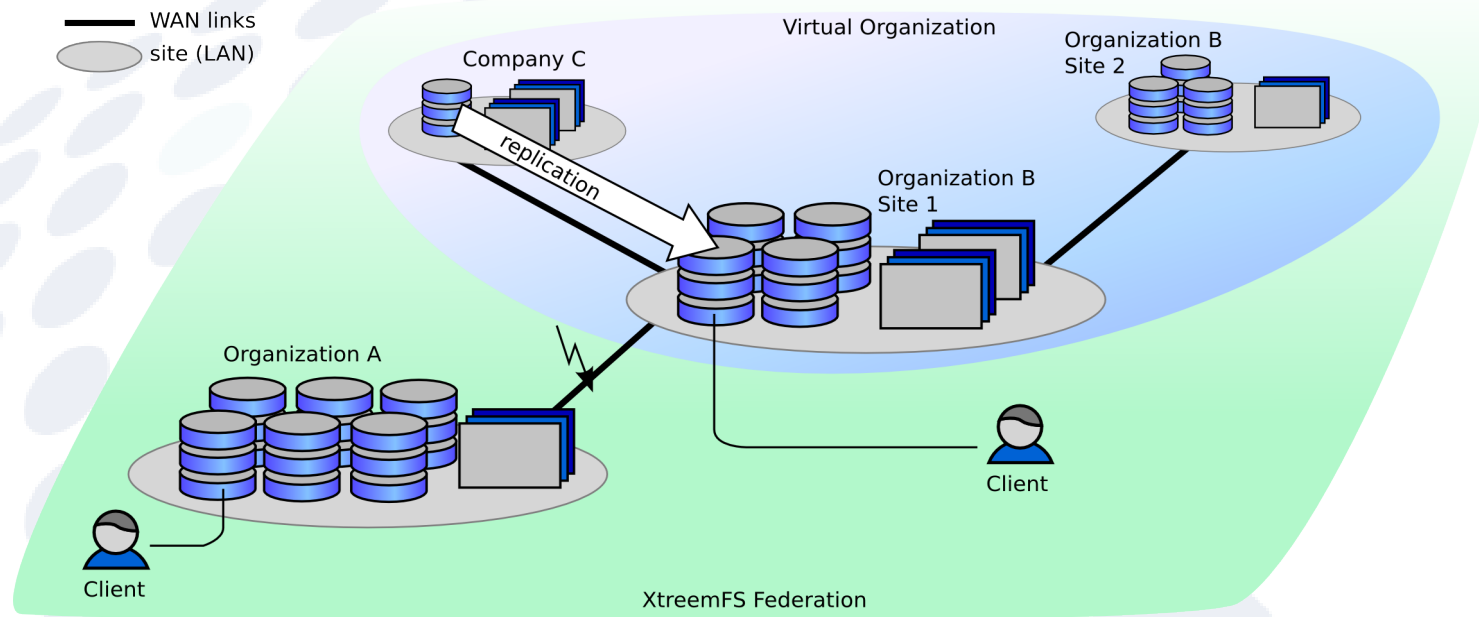
Target Environment



- federation: clusters can join/leave/fail
 - no centralized services at an organization
- connected over the Internet
 - complex failure cases (like network splits)
 - no control over hardware



Target Environment



- spanning administration domains
 - cross-organization authentication
 - virtual organization (VO) support necessary
- commonly referred to as *The Grid*



In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

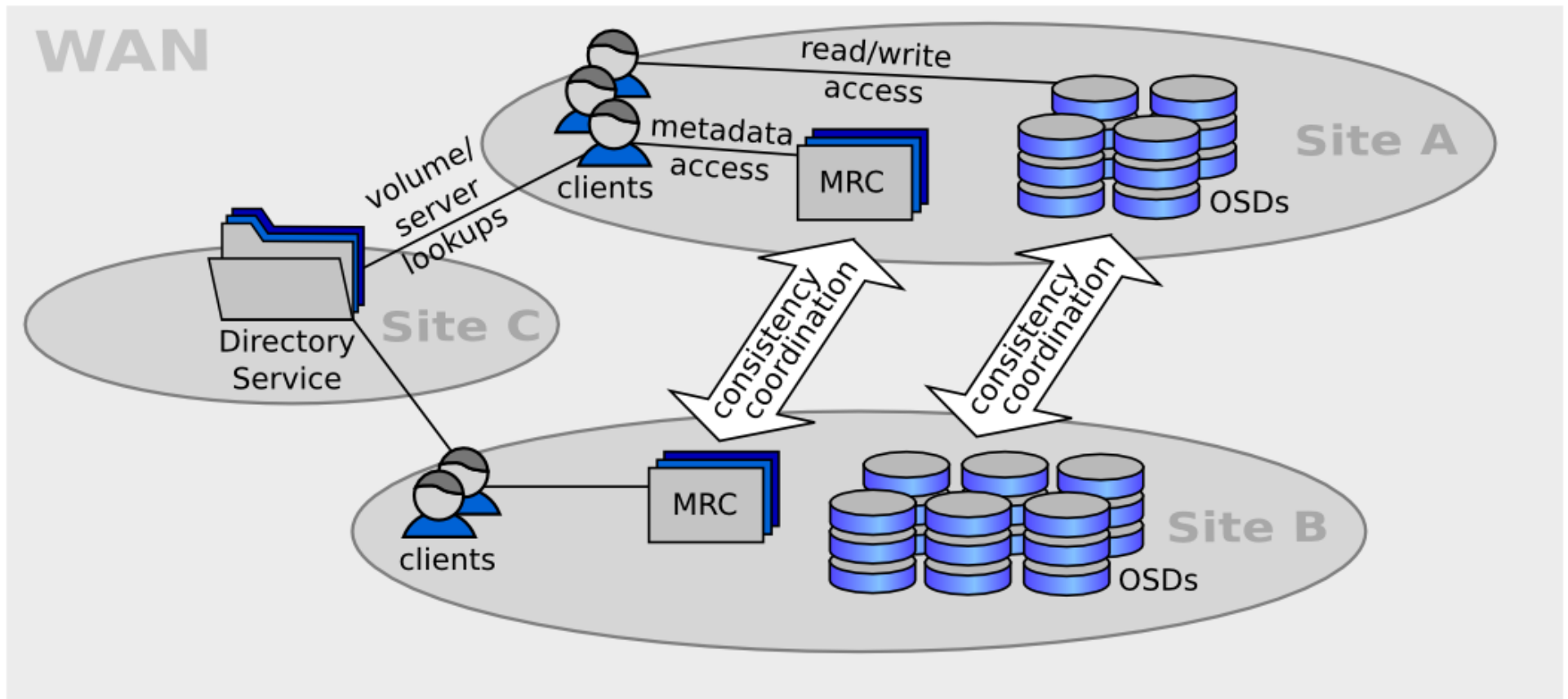
Features

Implementation

Current State & Plans



Architecture



In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

Features

Implementation

Current State & Plans



Features

- POSIX-compliant file system API ✓
- advanced metadata management
 - replication ✓ and partitioning of metadata
 - extended metadata ✓ and queries
- high performance
 - parallel file access (striping) ✓
 - client-side caching
- high data safety and availability
 - replication of files ✓
 - automatic access pattern-based replica management
 - RAID, end-to-end checksums



Features

- **POSIX-compliant file system API** ✓
- advanced metadata management
 - replication ✓ and partitioning of metadata
 - extended metadata ✓ and queries
- high performance
 - parallel file access (striping) ✓
 - client-side caching
- high data safety and availability
 - replication of files ✓
 - automatic access pattern-based replica management
 - RAID, end-to-end checksums

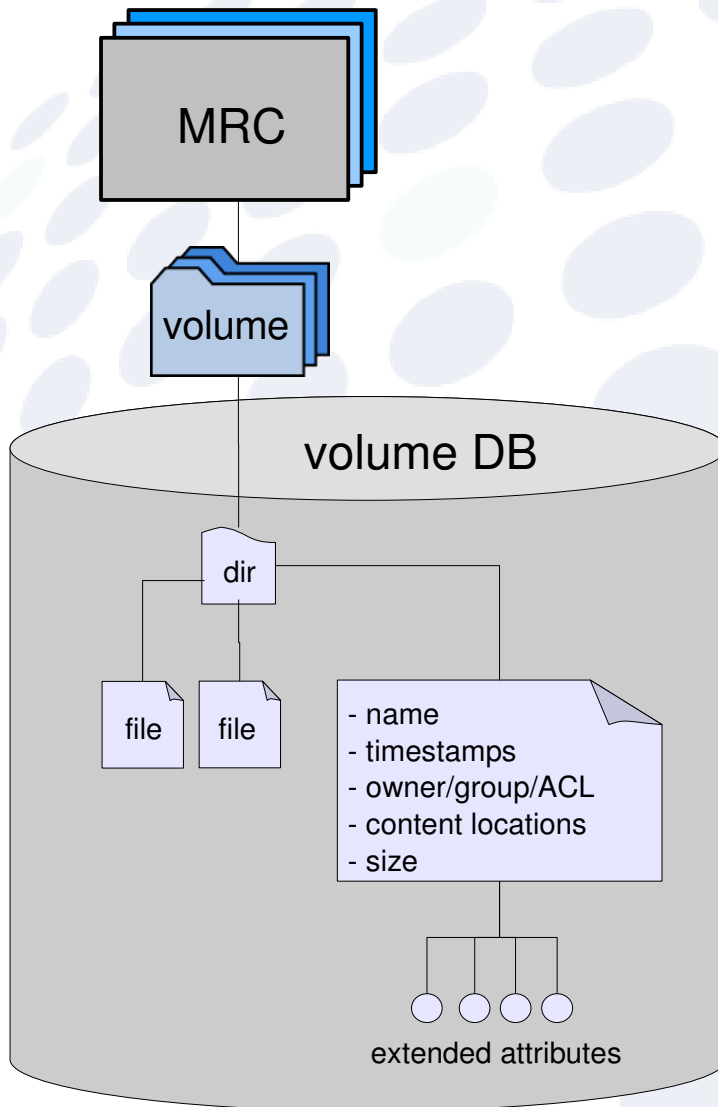


Features

- POSIX-compliant file system API ✓
- **advanced metadata management**
 - **replication ✓ and partitioning of metadata**
 - **extended metadata ✓ and queries**
- high performance
 - parallel file access (striping) ✓
 - client-side caching
- high data safety and availability
 - replication of files ✓
 - automatic access pattern-based replica management
 - RAID, end-to-end checksums



Features - Metadata Management

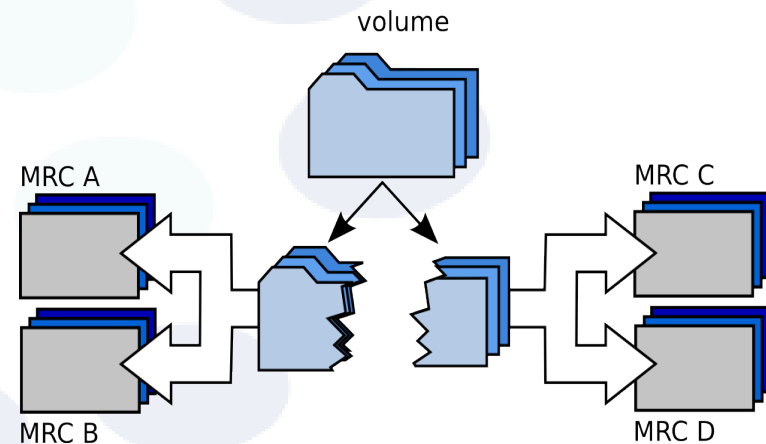


- **partitioning:**

- split up volume (DB) into smaller parts

- **replication:**

- primary/secondary with fail-over
- granularity: volumes / volume partitions



Features

- POSIX-compliant file system API ✓
- advanced metadata management
 - replication ✓ and partitioning of metadata
 - extended metadata ✓ and queries
- **high performance**
 - **parallel file access (striping)** ✓
 - **client-side caching**
- high data safety and availability
 - replication of files ✓
 - automatic access pattern-based replica management
 - RAID, end-to-end checksums



Features

- POSIX-compliant file system API ✓
- advanced metadata management
 - replication ✓ and partitioning of metadata
 - extended metadata ✓ and queries
- high performance
 - parallel file access (striping) ✓
 - client-side caching
- **high data safety and availability**
 - **replication of files** ✓
 - automatic access pattern-based replica management
 - RAID, end-to-end checksums



XtreemFS – Replication

replication of files

- read/write replication
- fully transparent to client
- guarantees sequential consistency
- primary/secondary approach with fault-tolerant lease negotiation

consistency coordination

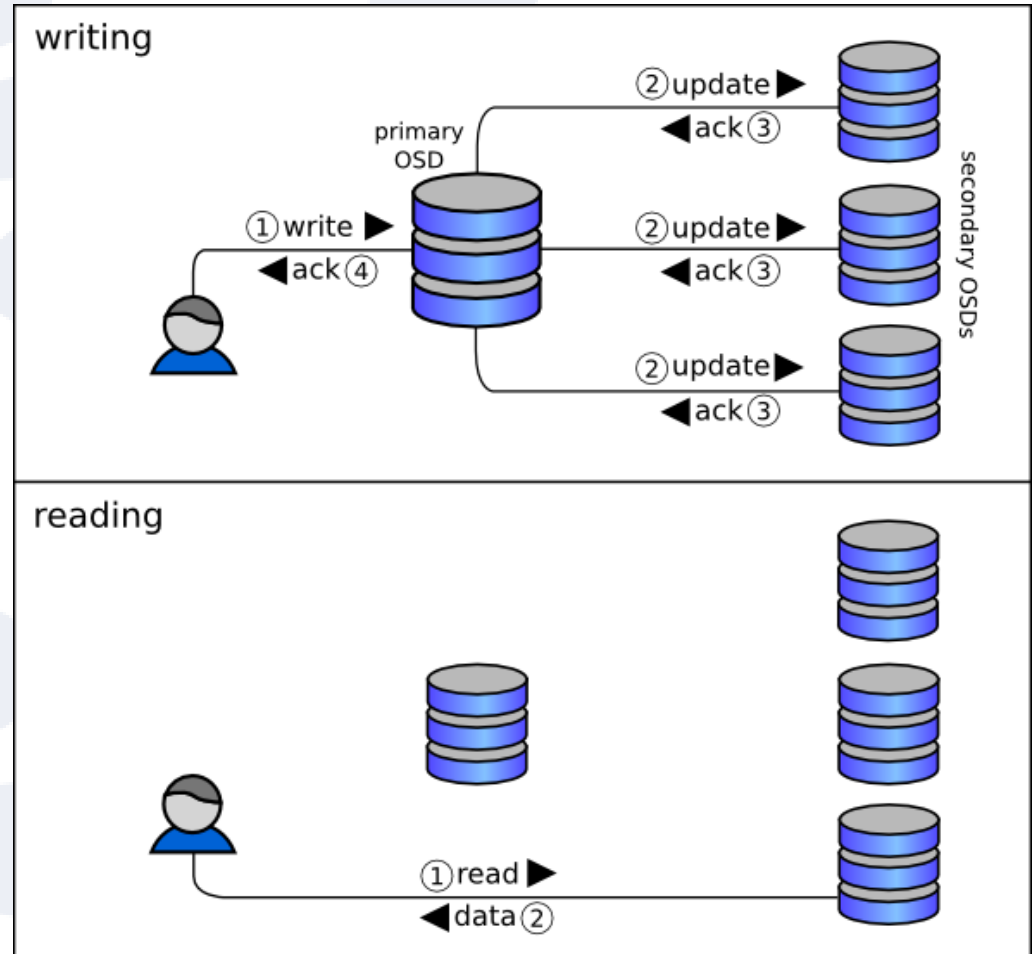
- currently at object level
- synchronous, asynchronous or on-demand



Features - Replication - Consistency Coordination

synchronous

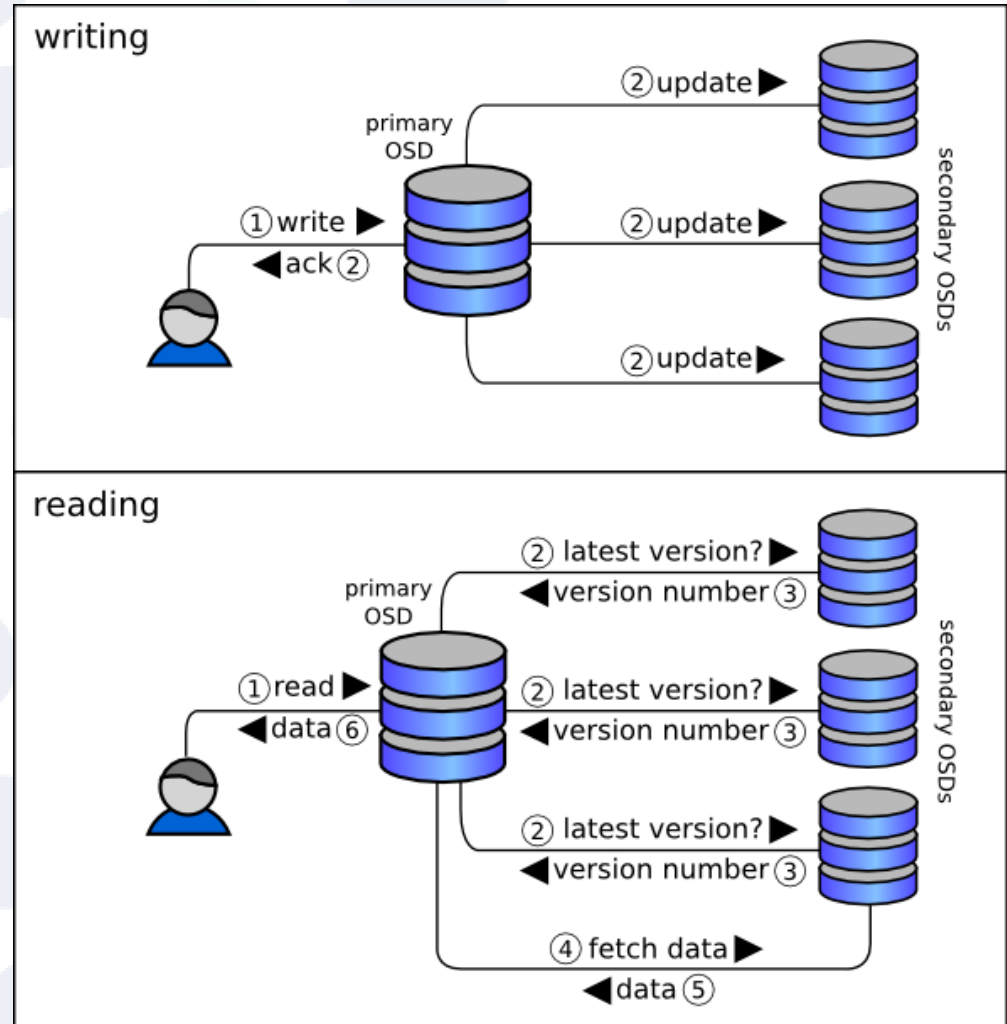
- **writing:** acknowledge after all updates have been acknowledged
- **reading:** on any replica



Features - Replication - Consistency Coordination

asynchronous

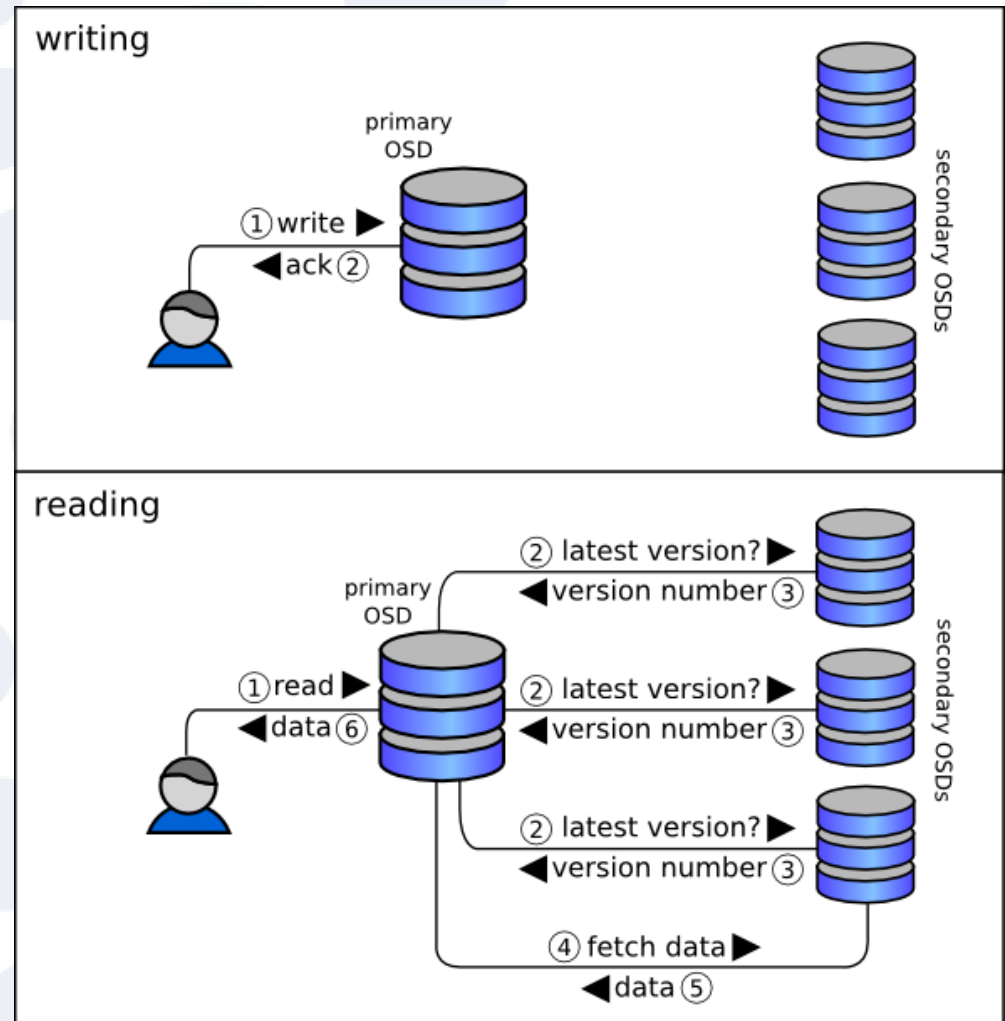
- **writing:** acknowledge when performed locally
- **reading:** check and fetch latest data



Features - Replication - Consistency Coordination

on-demand

- **writing:** acknowledge when performed locally, do not disseminate updates
- **reading:** check and fetch latest data



Features - Replication - Use Cases

problem

- large file staged / generated by a single process
- access by many clients
- each client accesses only a small portion
- clients reside on different sites

solution

- creation of a new (initially empty) local replica per client
- replicas are updated in background / on demand
- replica can be used immediately, required objects may be transferred on demand
- example: large database



Features - Replication - Use Cases

problem

- a producer gradually generates a large file
- a consumer wants to access already written parts of the file
- consumer and producer concurrently work on the same file

solution

- consumer and producer each have a local replica
- producer asynchronously updates consumer's replica
- consumer can access written objects locally



Features

- POSIX-compliant file system API ✓
- advanced metadata management
 - replication ✓ and partitioning of metadata
 - extended metadata ✓ and queries
- high performance
 - parallel file access (striping) ✓
 - client-side caching
- **high data safety and availability**
 - replication of files ✓
 - **automatic access pattern-based replica management**
 - **RAID, end-to-end checksums**



In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

Features

Implementation

Current State & Plans



Implementation

Protocol:

- HTTP (with JSON encoding for RPCs)

MRC, OSD, Directory Service:

- staged server implementation (non-blocking I/O)
- Java (~40.000 LOC) + BerkeleyDB (MRC)

File System Client:

- FUSE-based implementation (for now)
- C (~13.000 LOC)



In this talk ...

Introduction: Object-based File Systems

Target Environment

Architecture

Features

Implementation

Current State & Plans



Next Steps & Future Plans

next steps:

- performance improvements
 - read/write access
 - striping
 - replication (failure-free case)
- public release (by the end of 2007)

medium to long-term goals:

- RAID & checksums
- monitoring



Thanks for your attention!

