XtreemOS
Enabling Linux for the Grid

Object Sharing Service

John Mehnert-Spahn, Marc-Florian Müller,
Kim-Thomas Möller, Michael Schöttner

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Information Society Technologies

Motivation

- **Multicore CPUs**
  - **Parallel programming (multi threading)**
    --> shared memory within one process address space

- **In-memory architectures**
  - **In-memory data grids --> shared memory for grids**
  - **In-memory files, databases --> shared memory for clouds**

- **Sharing in-memory data is useful and there are open research topics**

Objectives

- Simplify the development of distributed and parallel applications (in clusters, grids, clouds)
    - Not replacing message-passing solutions but complement them

- Speed-up data access

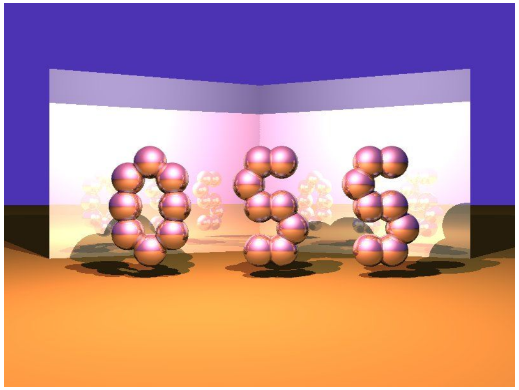
Approach

- **Speed-up data access by keeping data in RAM**
  - **Not a new file cache but keep everything in RAM**
- **Enable transparent remote memory access**
  - **System takes care of fetching non-present data**
- **Automatic replica management**
  - **For performance and reliability**
- **Multiple consistency models**
  - **Different data comes with different consistency requirements**

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Target Applications

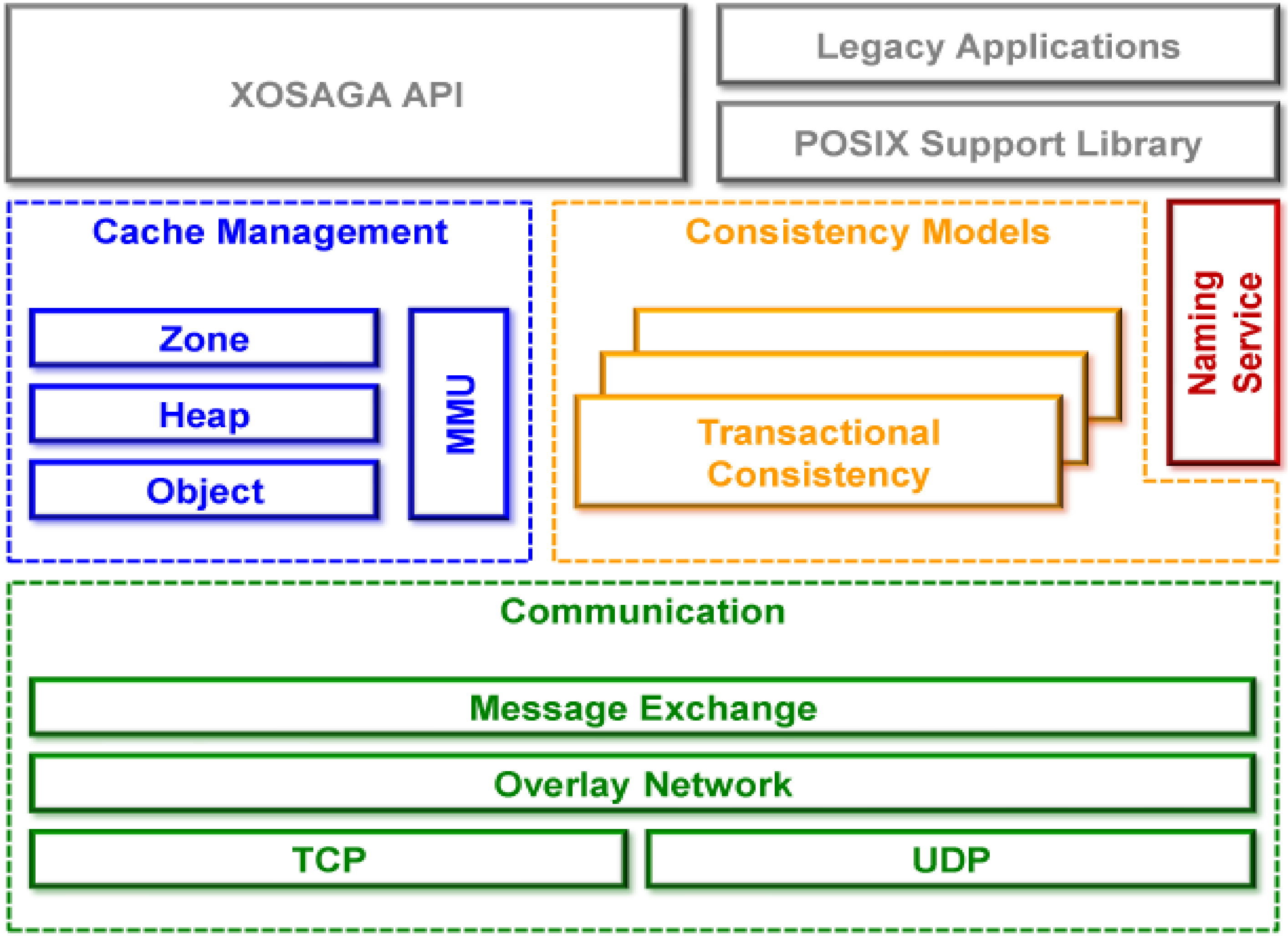
- **Computing and data intensive apps.**
  - **Simulations, data mining, ray tracing, …** →
  - **Typically a few large objects**
  - **Mostly scalar data (no pointers)**



- **Interactive applications**
  - **Multi-user games** →
  - **Many small objects**
  - **Object structures with and without pointers**

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

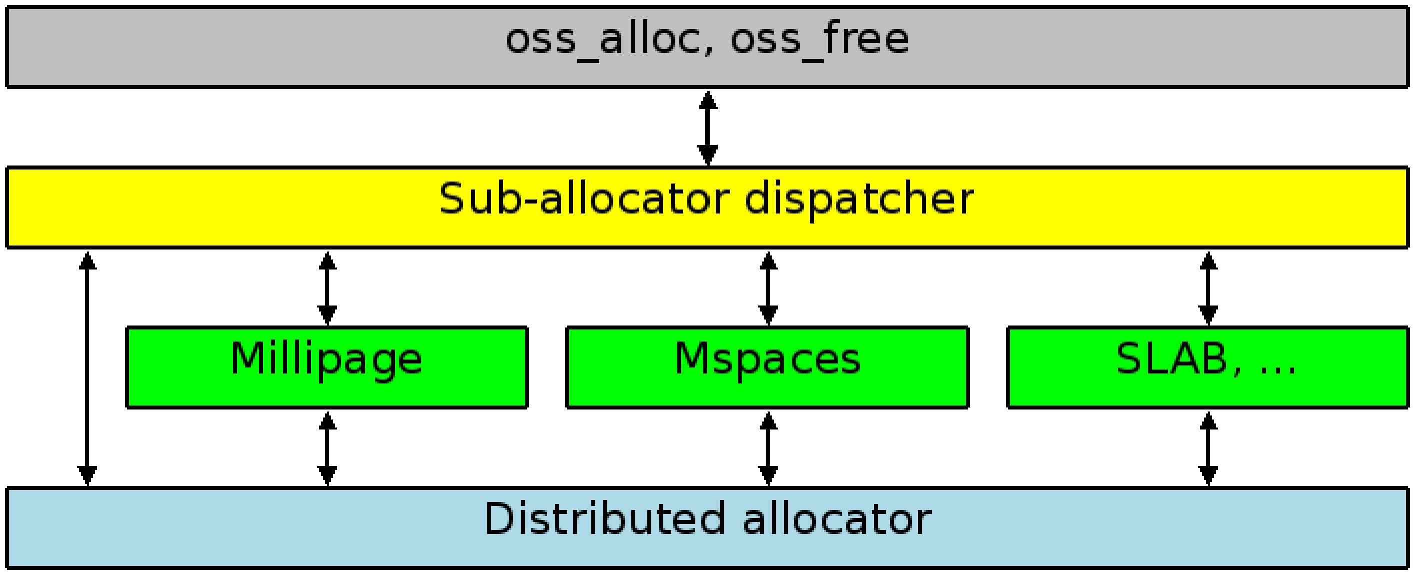
Architecture



XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Cache Management

- **Hierarchical memory management**
  - **Distributed allocator -> for allocating zones**
  - **Sub allocator -> node-local heap management**
  - **basic unit -> object**



XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Cache Management

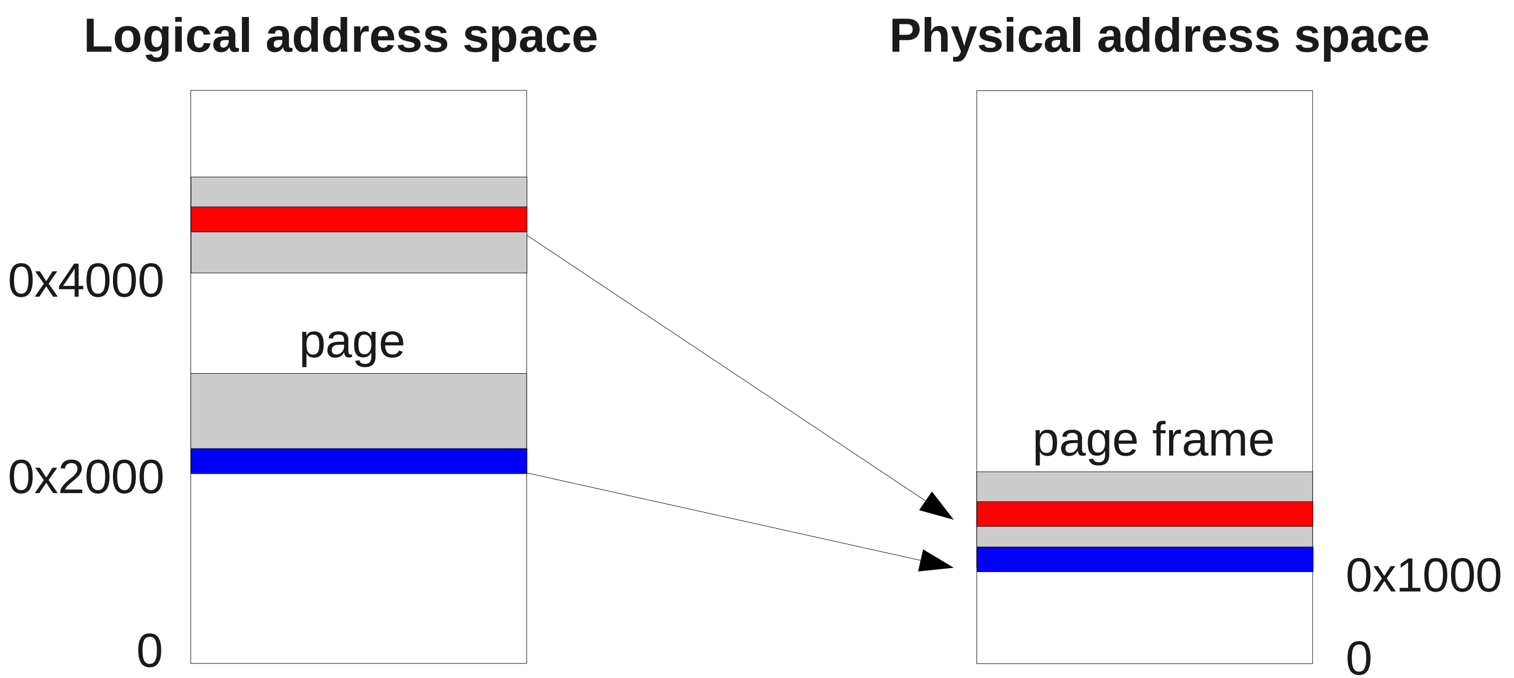
- **Access detection**
  - **Page-based approach --> language independent**

- **False sharing**
  - **Two or more objects reside on one page**
  - **One object is modified frequently by one node**
  - **Other objects are accessed by different nodes**

  --> result: page thrashing

False Sharing Avoidance

- **Using millipages**
  - **One page per object**
  - **Different pages mapped onto same page frame**



XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Consistency Models

- **Multiple consistency models are supported**
  - **within one application**
  - **It is easy to extend OSS by an own consistency model**
- **Typically defined during allocation of an object**
- **May be re-defined dynamically before accesses**
  - **Annotations by the programmer required**
  - **Allows optimizations**

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Distributed Transactional Memory

- Idea: optimistic instead of pessimistic transactions
  --> avoiding locks and deadlocks

- Transactions come with ACId properties

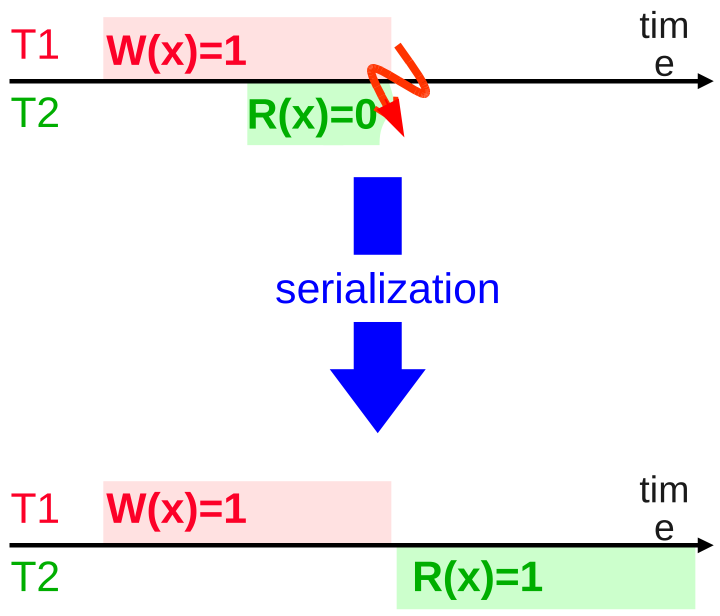
- For multicore machines but also for distributed systems

- Much interest in academia and industry in TM-systems
  - We expect more and more transactional applications

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

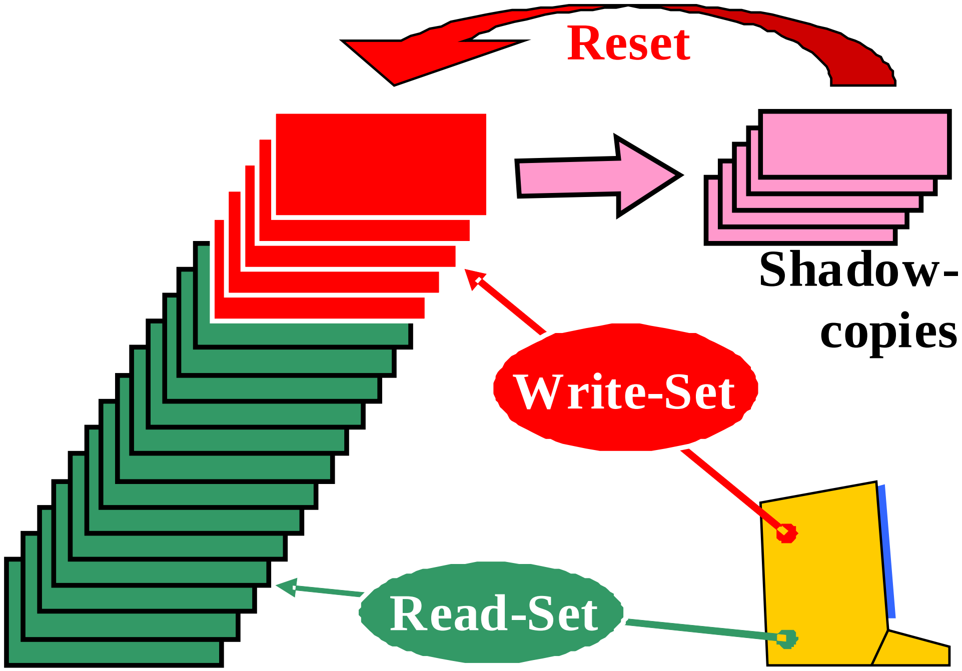
Validation

- **By comparing read and write sets of running overlapping transactions (forward validation)**
- **No conflicts --> commit transaction**
  - **Bundling of writes allows bulk network transfers**
- **Conflict --> abort and optionally restart transaction**



Restartability

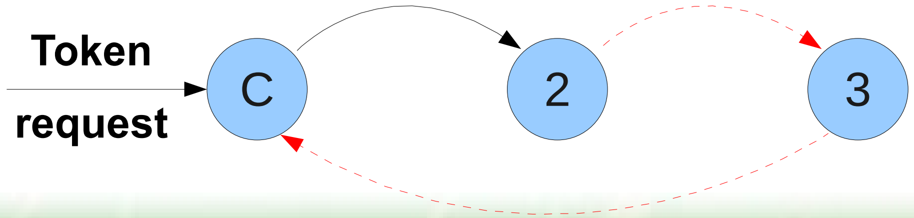
- **MMU-based access detection**
  - **One page fault for each first read or write access**
  - **Shadow copies for write accesses → restartability**



- **System- and I/O-calls are difficult …**

Commit Serialization

- **Only one transaction can commit at a point of time**
  - **Can be implemented by passing a token around**
  - **Problem: hard to find the token**
- **Solution: coordinator-based token-passing**
- **Optimization:**
  - **Extend token by a request-list (filled by coordinator)**
  - **Token can be passed directly between peers (until list is empty)**



Commit Propagation

- **Commits are numbered by a 64-Bit Commit-Number**
  - **Incremented by each commit**
  - **Stored in the token**
- **Allows sending commits without waiting for acks**
- **Receivers buffer and sort incoming commits and deliver them in ascending commit-number order**

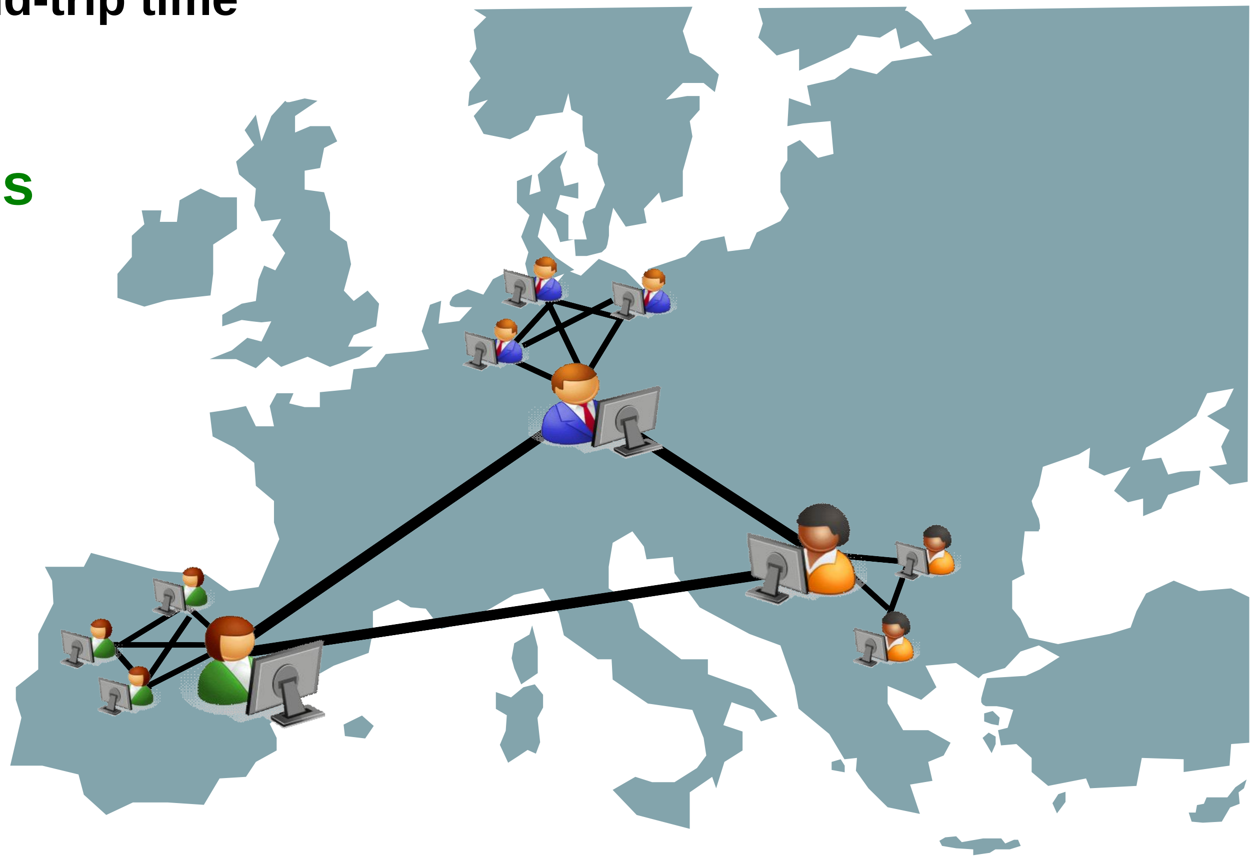
XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Overlay Network

- **Group geographically near nodes**
  - **Heuristic: ping round-trip time**
- **Super-peer candidates**
  - **Nodes with good network connection**
  - **High cpu power**

...

- **For scalability**



XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Super-Peer Commit

- **Token passed among super peers, only**
  - **Reduced number of nodes compete for the token**
- **Super-peers can commit a bunch of transactions**
- **While waiting for the token they can validate pending transactions of their group against each other**
  - **Conflicts may be detected before token arrives --> affected transaction can be aborted immediately**
  - **Also allows transaction ordering, e.g. for fairness**

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Further Scalability Aspects

- **Consistency domains**
  - **Transactions run within a certain domain**
  - **Domains are synchronized separately**
- **Local Commit (without network communication)**
  - **If transaction has not written any data**
  - **Or if only data has been modified that is not replicated**
- **Chained transactions to mask commit latency**
  - **Start with next transaction, while commit is pending**

Replica Management

- **Adaptive replication based on monitoring access patterns**
  - **Mix of updates and invalidates**
  - **For performance near accessing clients**
  - **For reliability spread in the network**

- **Backup replicas**
  - **Needed to allow local commits**
  - **Must not be accessed directly**

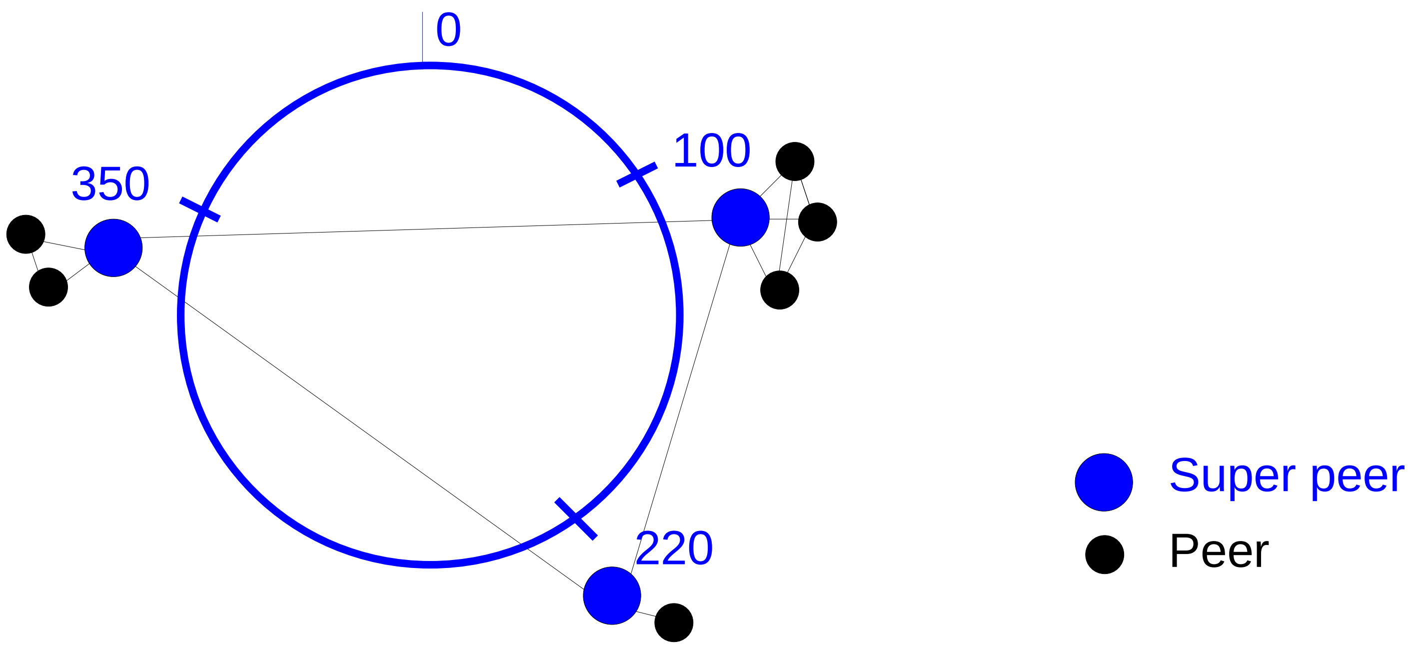
Data Search

- **Data objects can be named and registered in the built-in naming service**
  - **Entries point to data objects using an address/ID**
  - **Where to find this address/ID ?**

- **Super peers manage the global address/ID space in a ring**
  - **A new arriving super peer gets its logical position in the ring using a hash function**
  - **It takes over a part of the ID space from its successor**
  - **From then it is reponsible for zone allocations in this subspace**
  - **This allows a conflict-free global address/ID space management**

Address Space Mgmt.

- **Example: 3 super peers, 6 additional peers**



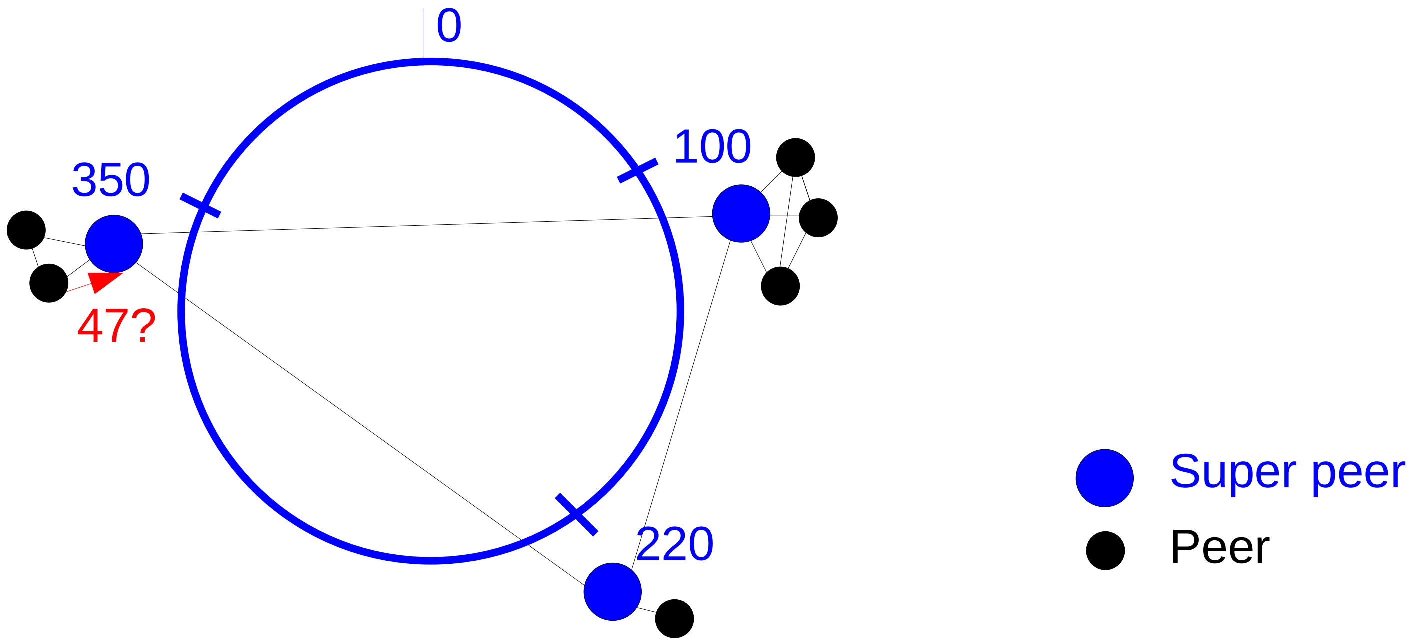
XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Data Search

- **If an address/ID is unknown a node contacts its super peer**
  - **The super peer can easily determine the super peer responsible for the address/ID region wherein the searched address/ID resides**
  - **It contacts this super-peer which will know which node has allocated a zone within its managed region**
  - **This note has still this object or knows at least to which node it has sent the object**

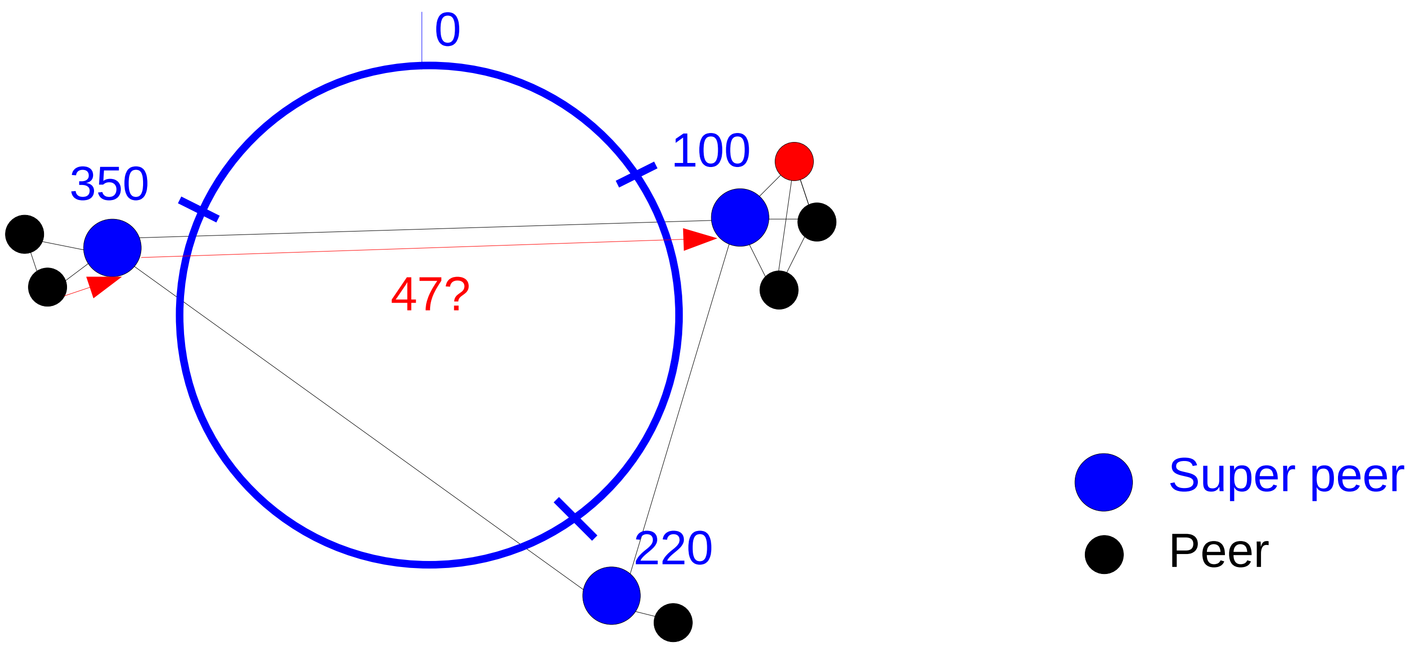
Data Search Example

- A node asks its super peer for object 47



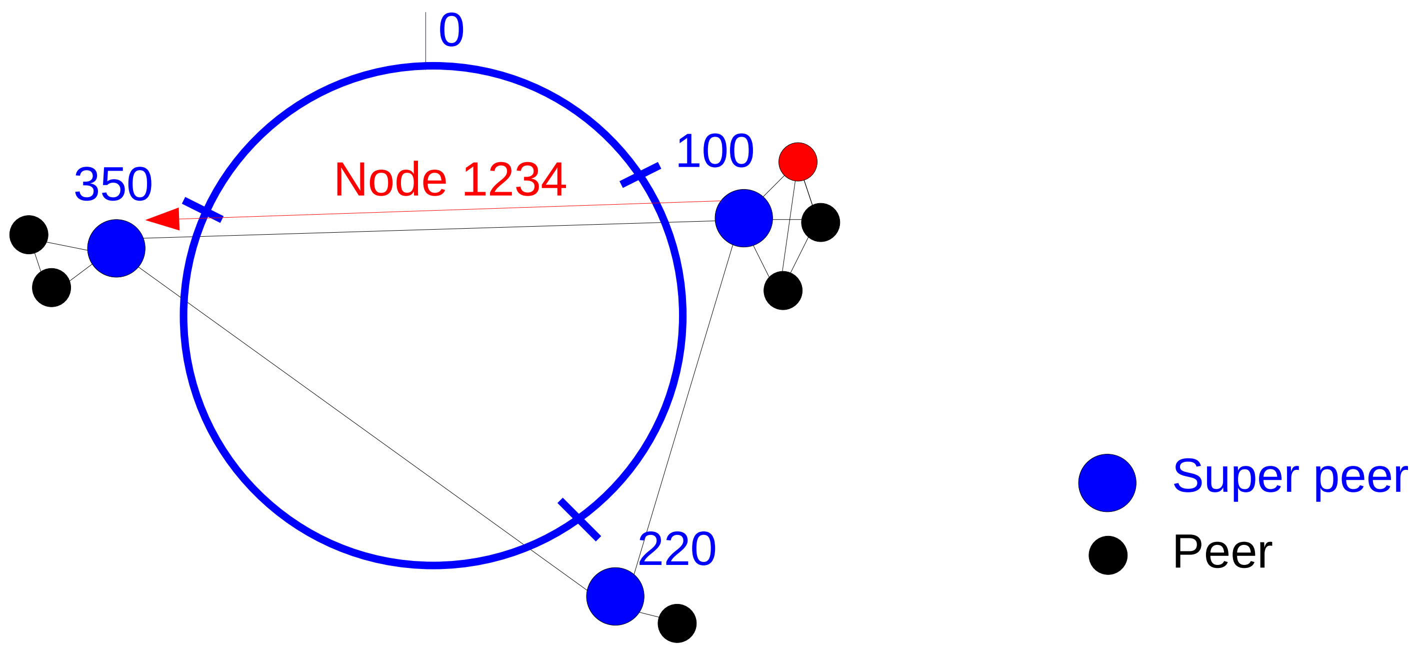
Data Search Example

- The super peer forwards this request to the appropriate super peer



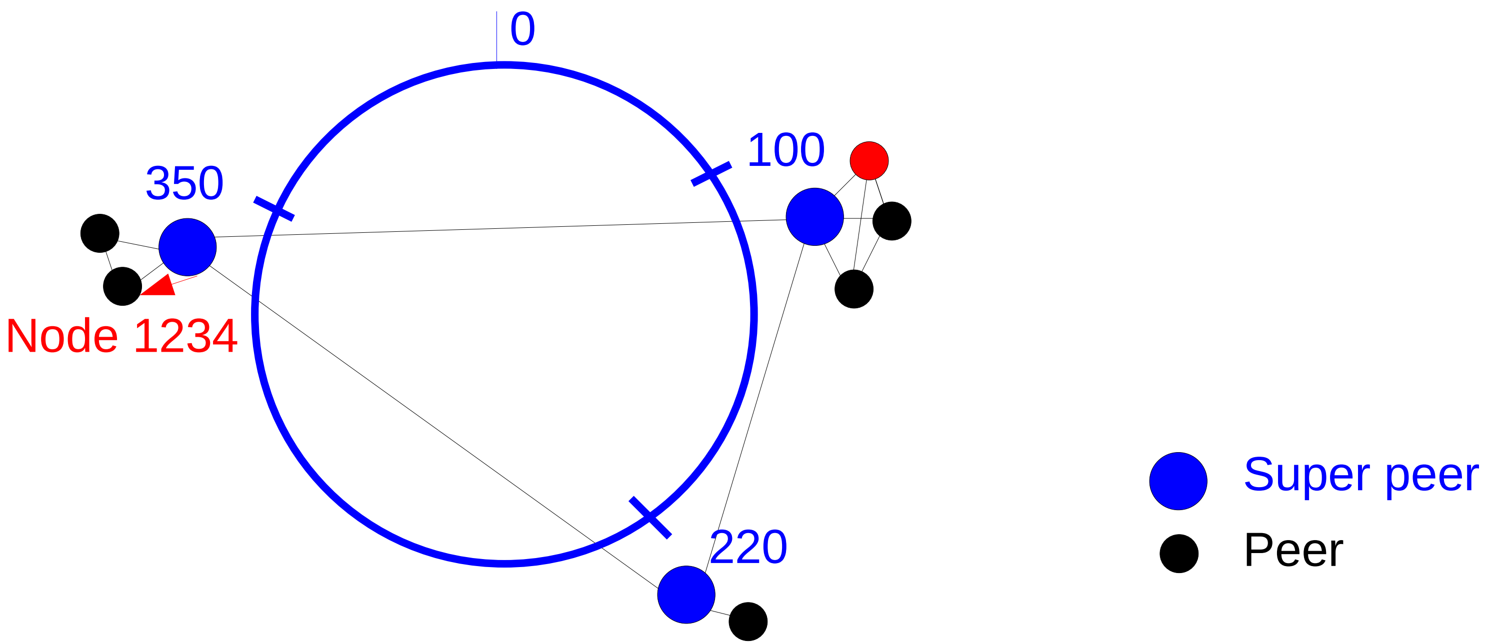
Data Search Example

- **This super peer replies with the peer ID that has allocated the relevant zone**



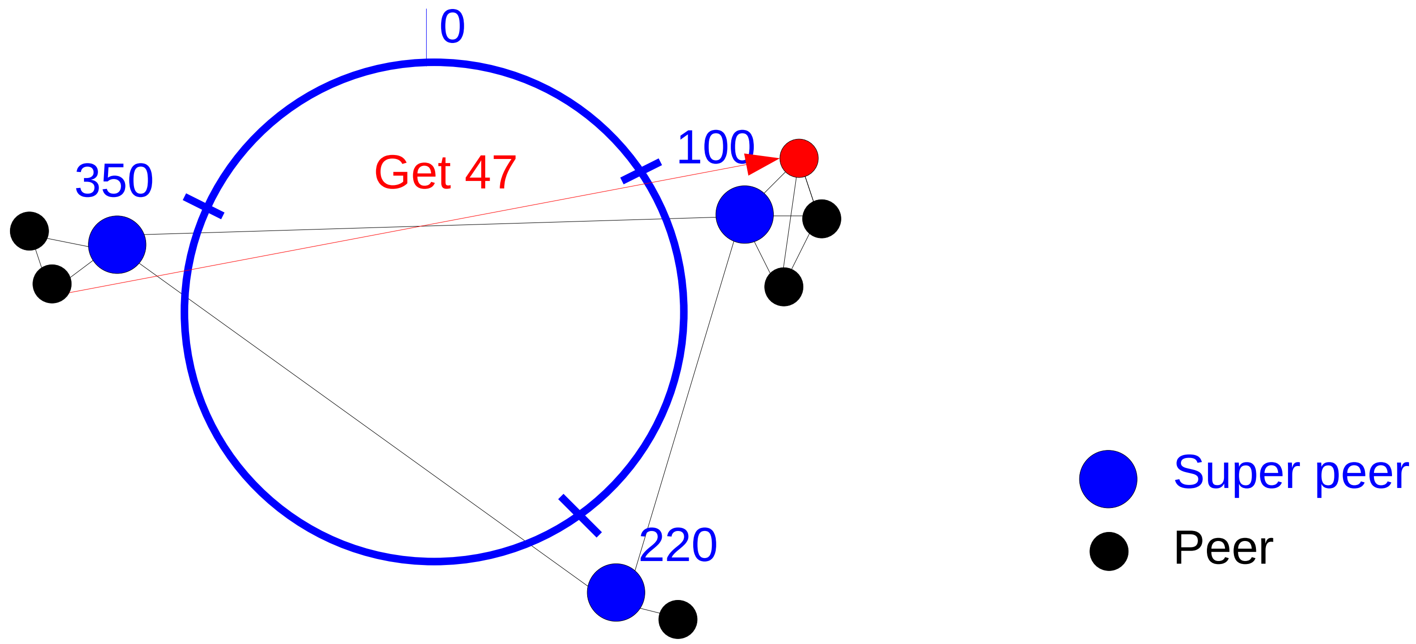
Data Search Example

- **The super peer forwards the reply to the requesting peer**



Data Search Example

- Finally, the node sends the request directly to the node managing the relevant zone



It may then receive the object or the node ID of the actual object holder

Data Search Optimizations

- **Super peers and other peers see commits**
--> all nodes learn about object migrations

- **Super peers store finished commits (transaction history)**
  - **Necessary if node needs to recover from missed commits**
  - **But also useful for data search:**
    - **If a super peer has the searched object ID in its cache it directly contacts this node**
    - **If this search fails it searches the zone holder (like described before)**

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Reliability Aspects

- **Fault-tolerant overlay-network**
- **Adaptive replication**
  - **For performance near accessing clients**
  - **For fault reliability spread in the network**
- **Transaction history buffer to recovery missed write sets**
- **Checkpointing for fault tolerance**

XOSAGA Example


```
shared_buffer_service   sbs(bootstrap_url, local_url);
transactional_domain    dom;
shared_buffer           buf;
transaction_id          tid;

dom = sbs.create_transactional_domain("test_transactional");
buf = dom.create_buffer(buf_name, buf_size);

tid = dom.begin();
// ...
dom.commit(tid);
```


XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Conclusions

- **In-memory data sharing allows fast data access and simplifies the development of distributed & parallel applications**

- **Automatic replication provides performance & reliability**

- **Multiple consistency models within one app. for different needs**
  - **Including distributed transactional memory --> strong consistency, comparable efficient**
  - **But weaker consistency models, too**

- **Tested with different apps., including a multi-user virtual world**

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Outlook

- **Tests with further applications:**
  - **software engineering**
  - **bioinformatics**
  - **...**

- **Data sharing for cloud applications (map&reduce)**

- **Optimization of replica management, transactions, ...**

Contacts

Marc-Florian Müller: transaction, replication, network
Marc-Florian.Mueller@uni-duesseldorf.de

Kim-Thomas Möller: cache management, replication, data search, network
Kim-Thomas.Moeller@uni-duesseldorf.de

Michael Schöttner: team leader
Michael.Schoettner@uni-duesseldorf.de