FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

An Adapative and Flexible Replication Mechanism
for the XVSM Reference Implementation Mozartspaces

Masterstudium:
Software Engineering &
Internet Computing

Jürgen Hirsch

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übersetzerbau
Betreuerin: A.o. Univ. Prof. Dr. Dipl.-Ing. eva Kühn

Mozartspaces

- Based on eXtensible Virtual Shared Memory (XVSM)
- Provides a shared view of data (space)
- High level of reliability, scalability
- Data arranged in containers
  - May have name, capacity
- Coordinators to structure data in containers
  - Built-In: FIFO, LIFO, Vector, Linda, Key, Label, Query, Random
  - Additional user defined coordinators
- Provided operations:
  - Lookup container
  - Create, destroy or lock a container
  - Write, take, delete or read entries from a container
  - Create, commit or rollback a transaction
  - Add or remove container or space aspect

Replication

- Obligatory in distributed environment
- Improves performance, availability and safety
- Create multiple replicas on different nodes
- Additional difficulties in WAN environments
  - High latency
  - Node not always available

- Different approaches:
  - Middleware or integrated replication
  - Strategy (master-slave or multi master)
  - Update propagation (lazy or eager)
  - Reliability (ACID, BASE)
  - Isolation (serializable, repeatable reads, read committed, read uncommitted)

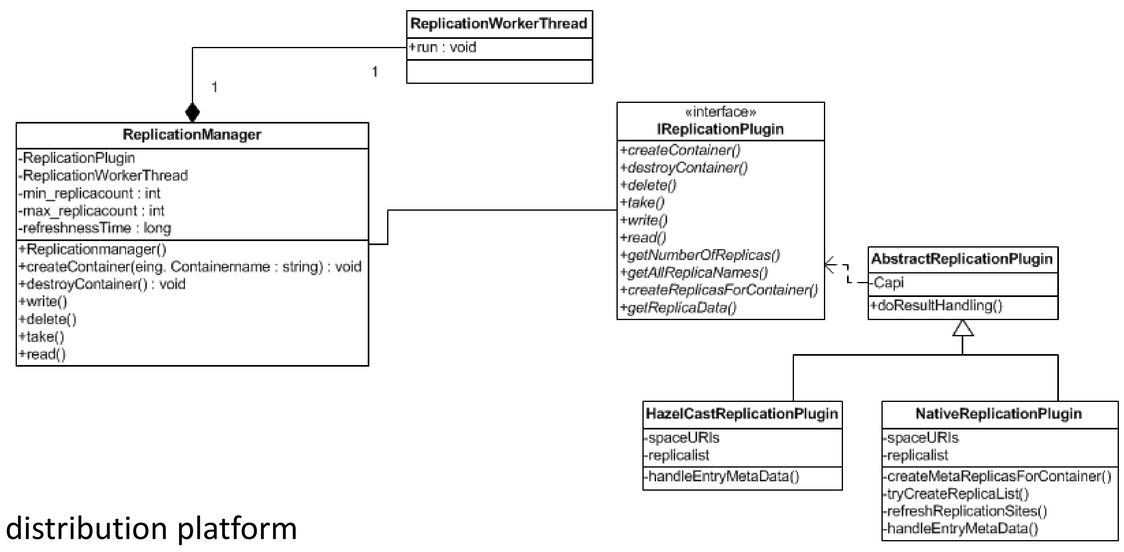
Motivation

- Asynchronous replication mechanism
- Increase data availability and safety
- Integration as profile for Mozartspaces v2.0
- Support all features of Mozartspaces
- Minimum influence on performance

- Replace Mozartspaces v1.0 replication mechanism
  - used in RealSafe project (ASFINAG)
  - depends on old middleware
  - limited coordinator support
  - missing error handling

Implementation

- Implementation as facade (Replication manager)
- Replication manager uses plugins which contain concrete replication logic
- Replication manager provides a replication API
- Two different plugins implemented
  - Hazelcast plugin
  - Native plugin
- Quality of service component
  - Monitors created replicas
  - Creates new replicas if count falls below certain level
- Error handling
  - Quorum approach to deal with errors on individual nodes
- Mozartspaces specific consistency strategies
  - Strict and loose consistency



Hazelcast

- Open source data distribution platform
- Provides distributed hashmap, publish / subscribe mechanism
- Native Java
- Provides internal replication mechanism

Hazelcast Plugin

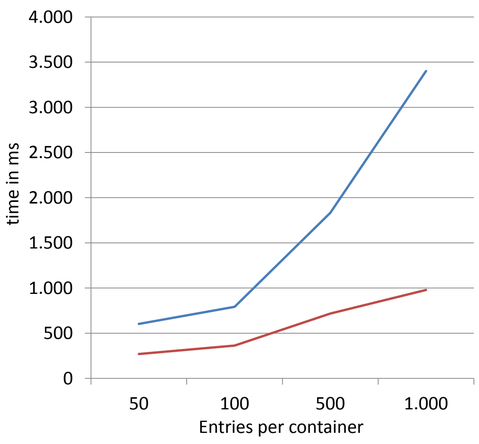
- Uses Hazelcasts distributed hashmap
- Asynchronous multi master replication
- Read once, write all available (ROWAA) approach
- Provides 1 copy serializability
- Takes advantage of Hazelcasts internal replication mechanism
- Uses Hazelcasts locking mechanism to acquire locks on multiple replicas

Native Plugin

- Only uses functionality of Mozartspaces v2.0
- Asynchronous multi master replication
- ROWAA approach with 1 copy serializability
- Metacontainer for storing Mozartspaces and replication related metadata
  - needs to be replicated too
- Lockcontainer to acquire locks when performing updates

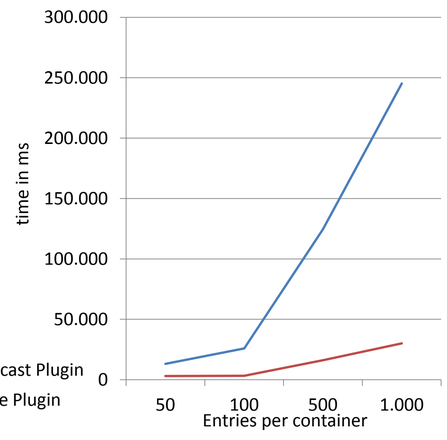
Results

- Native plugin performs better
  - when number of nodes low
- Hazelcast accesses expensive



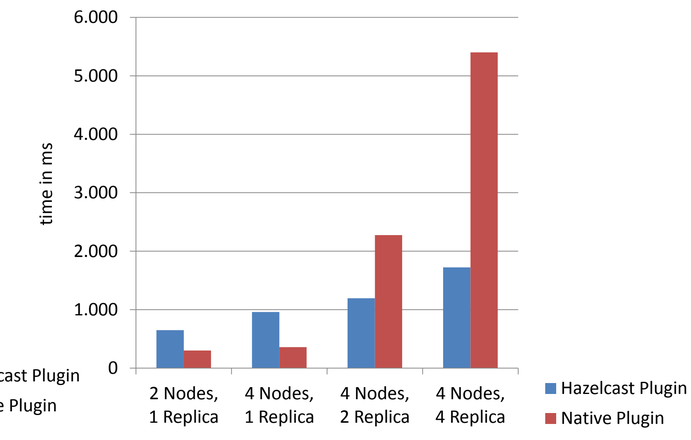
100 Container – WRITE performance

- Hazelcast plugin scales better
- At certain level Hazelcast access cheaper than native metadata handling



100 Container – TAKE performance

- Future improvements:
  - Cache Hazelcast information to avoid unnecessary accesses



Scalability – WRITE performance

Conclusion

Fulfilled requirements:
- Adaptable & flexible replication
  - due to different plugins
- Supports all coordinators
- Locking mechanism
- Asynchronous replication
- Advanced error handling

Future work:
- Performance optimizations
- Additional plugins
- Integration in Mozartspaces API
- Distributed transactions
- Integrate meta model querying mechanism of Mozartspaces

MozartSpaces is open source, download it at www.mozartspaces.org.

Contact: juergen.hirsch@gmail.com