AARHUS UNIVERSITET

Software Architecture in Practice

Tuplespace using GigaSpace Experience Report

Henrik Bærbak Christensen

AARHUS UNIVERSITET

Introduction

- I have never worked with a Tuplespace implementation, so I found it interesting to test-drive it…
- It proved more difficult than expected…
  - There is no maven repository jar
    - Thus
      - Clone the git repo (https://github.com/xap/xap)
      - Build it using *only*
        - » Java 8, Maven
      - Not simple to produce a jar to feed into gradle
    - And documentation leaves much to wish…
- Conclusion:
  - Felt like something left behind 10 years ago…

AARHUS UNIVERSITET

Tuple Space

- Basically a ‘distributed shared memory’ abstraction
  - Read and Write of Java POJO/beans

GIGASPACES

```
//write "Hello" and "World!" to the data-grid
write(space, new Message("Hello"));
write(space, new Message("World!"));

//read "Hello" and "World!" from the data
read(space, new Message());
```

```
Created embedded data-grid: myspace
write - 'Hello'
write - 'World!'
read - ['Hello', 'World!']
```

```
/*
 * Write (or update) an entity in the data-grid
 */
private static void write(GigaSpace space, Message message) {
    LeaseContext<Message> context = space.write(message);

    if (context.getVersion() == 1) {
        System.out.println("write - " + message);
    } else {
        System.out.println("update - " + message);
    }
}

/*
 * Read a matching entity from the data-grid
 * Template matching is done by field equality or any if field is null
 */
private static void read(GigaSpace space, Message msgTemplate) {
    Message[] results = space.readMultiple(msgTemplate);
    System.out.println("read - " + Arrays.toString(results));
}
```

AARHUS UNIVERSITET

Notifications

- How do I know that a value has changed?
  - Tuple spaces are perhaps best where it does not matter
- But – you can assign event notification
  - Essentially making a publish-subscribe system

```
@EventDriven
@Notify
class MessageListener {

     @EventTemplate
     Message unprocessedData() {
        Message template = new Message();
        // template.setMsg("fisk");
        return template;
     }

    @SpaceDataEvent
    public Message eventListener(Message event) {
      System.out.println(" -> Event triggered/ Write of : " + event);
        // process Payment
        return null;
    }
}
```

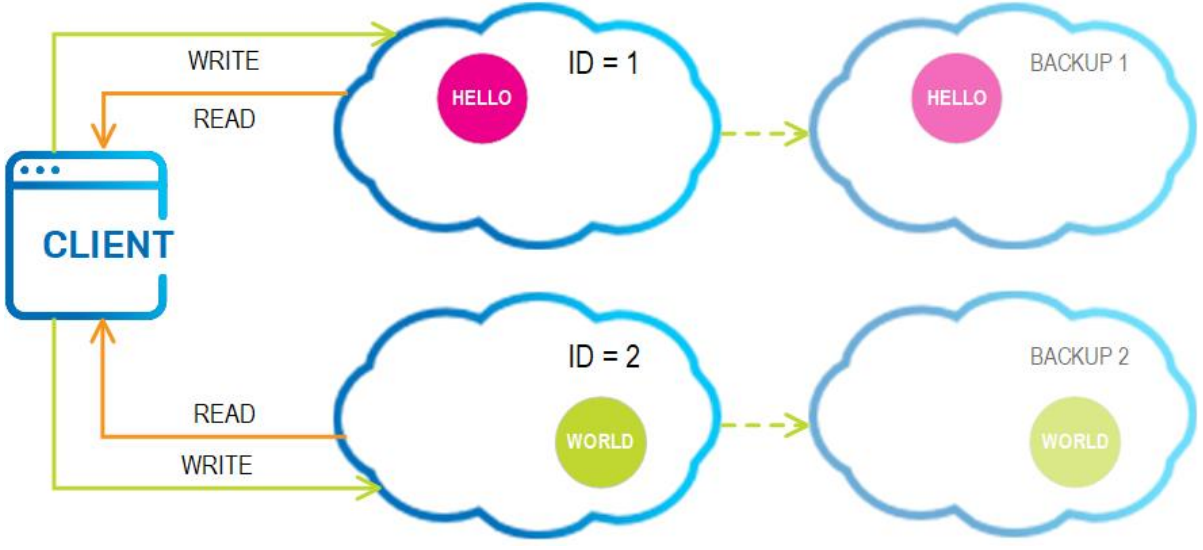
```
// Register a listener on message events
SimpleNotifyEventListenerContainer
  notifyEventListenerContainer =
  new SimpleNotifyContainerConfigurer(space)
  .eventListenerAnnotation(new MessageListener())
 .notifyContainer();
notifyEventListenerContainer.start();
```

```
Created embedded data-grid: myspace
write - 'Hello'
write - 'World!'
 -> Event triggered/ Write of : 'Hello'
 -> Event triggered/ Write of : 'World!'
read - ['Hello', 'World!']
```

AARHUS UNIVERSITET

Partitioning & Replication

- The tuple space is *single point of failure* so classic replication techniques can be employed
  - As well as partitioning schemes…



QA

- But reliable storage is not the primary concern
- *Low latency* and *high performance* are QA drivers…
- Scalability through parallel processing…

- **Nature of data is essential**
  - Event based events flow between components
    - Important that each and every event is handled
  - State based components read/write states
    - Only most recent state often matters!

Example

- ABS braking system
  - Wheel velocity, v, is continuously monitored
    - Loosen brakes in case v = 0
  - Does not really matter what v was 2 seconds ago, the only thing that matters is *now!*
- Typical for state-based system

- CAN bus
  - Write semantics all read same value or all fail
  - Just poll the last value is enough

AARHUS UNIVERSITET

Complex Query Language

- Gigaspaces approaches a full database ☺

SQL Query

The SQLQuery class is used to query the space with an SQL-like syntax. The query statement includes only the WHERE statement part. An SQLQuery is composed from the class of entry to query and an expression in SQL syntax.

```
public User[] sqlFindUsersByName() {
     SQLQuery<User> query = new SQLQuery<User>(User.class,"name = 'John Doe'");
     return space.readMultiple(query);
}

public User[] sqlFindUsersByNameAndCreditLimit() {
     SQLQuery<User> query = new SQLQuery<User>(User.class,"name = 'John Doe' AND creditLimit > 1000");
     return space.readMultiple(query);
}

public User[] sqlFindUsersByNameAndIds() {
     SQLQuery<User> query = new SQLQuery<User>(User.class,"name = 'John Doe' AND id IN(1L,3L,5L)");
     return space.readMultiple(query);
}
```

AARHUS UNIVERSITET

According to Wikipedia

- Mostly used in financial and telco

JavaSpaces is part of the Java Jini technology, which on its own has not been a commercial success.[1] The technology has found and kept new users over the years and some vendors are offering JavaSpaces-based products. JavaSpaces remains a niche technology mostly used in the financial services and telco industries where it continues to maintain a faithful following. The announcement of Jini/JavaSpaces created quite some hype although Sun co-founder and chief Jini architect Bill Joy put it straight that this distributed systems dream will take "*a quantum leap in thinking*".[2]

AARHUS UNIVERSITET

Summary

- I would pick another technology than GigaSpaces unless I really need all the bells and whistles