

JGroups: Multicast Messaging

Introduction to Middleware

Vojtěch Horký Petr Tůma

Department of Distributed and Dependable Systems
Faculty of Mathematics and Physics
Charles University

2010 – 2022

Outline

- 1 Technology Overview
- 2 Assignment Part I
- 3 Interface Overview
- 4 Assignment Part II

Technology Overview

Goals

Provide reliable group messaging mechanism.

Features

- Basic group messaging interface.
- Groups identified by names.
- Messages are byte arrays.
- Configurable protocol stack.
 - ▶ Multiple underlying transports.
 - ▶ Multiple reliability mechanisms.
 - ▶ Multiple membership discovery mechanisms.
 - ▶ Multiple error recovery mechanisms.
 - ▶ ...

... <http://www.jgroups.org>

Outline

- 1 Technology Overview
- 2 Assignment Part I**
- 3 Interface Overview
- 4 Assignment Part II

Assignment

Peer

Implement a process that will update a shared hash map.

- The shared hash map is available through SharedHashMap channel.
- The updates are transmitted through UpdateEvent class.

```
import java.io.Serializable;

public class UpdateEvent implements Serializable {
    private static final long serialVersionUID = 0xBAADBAADBAADL;

    public int key;
    public String value;
}
```

Examples To Begin With ...

```
> git clone http://github.com/d-iii-s/teaching-introduction-middlewre.git
```

Java

```
> cd teaching-introduction-middlewre/src/jgroups-basic-peer/java  
> cat README.md
```

Outline

- 1 Technology Overview
- 2 Assignment Part I
- 3 Interface Overview**
- 4 Assignment Part II

JChannel Class

```
public class JChannel implements Closeable {
    public JChannel ();
    public JChannel (String properties);
    public JChannel (InputStream configuration);

    public JChannel connect (String cluster_name);
    public JChannel disconnect ();

    public JChannel send (Message msg);
    public JChannel send (Address dst, Object obj);
    public JChannel send (Address dst, byte [] buf);

    public JChannel setReceiver (Receiver r);
    public Receiver getReceiver ();

    public View getView ();

    public JChannel addChannelListener (ChannelListener listener);
    public JChannel removeChannelListener (ChannelListener listener);

    ...
}
```


Message Interface

```
public interface Message {  
    public Address getDest ();  
    public Message setDest (Address new_dest);  
    public Address getSrc ();  
    public Message setSrc (Address new_src);  
  
    ...  
}
```

ByteMessage Class

```
public class ByteMessage implements Message {
    public ByteMessage ();
    public ByteMessage (Address dest);
    public ByteMessage (Address dest, byte [] array);
    public ByteMessage (Address dest, byte [] array, int offset, int length);

    public int getOffset ();
    public int getLength ();
    public byte [] getArray ();
    public ByteMessage setArray (byte [] b, int offset, int length);

    ...
}
```

ObjectMessage Class

```
public class ObjectMessage implements Message {  
    public ObjectMessage ();  
    public ObjectMessage (Address dest);  
    public ObjectMessage (Address dest, Object obj);  
  
    public <T extends Object> T getObject ();  
    public ObjectMessage setObject (Object obj);  
  
    ...  
}
```

Receiver Interface

```
public interface Receiver {  
    default void receive (Message msg);  
    default void receive (MessageBatch batch);  
  
    default void viewAccepted (View new_view);  
  
    default void block ();  
    default void unblock ();  
  
    default void setState (InputStream input);  
    default void getState (OutputStream output);  
}
```

ChannelListener Interface

```
public interface ChannelListener {  
    public void channelClosed (JChannel channel);  
    public void channelConnected (JChannel channel);  
    public void channelDisconnected (JChannel channel);  
}
```

Code Now ...



<http://www.commitstrip.com/en/2018/11/20/one-final-detail>

Show Your Code ...

Query Host Name

```
> hostname  
u1-22
```

Run Screen Sharing

```
> x11vnc -viewonly
```

Outline

- 1 Technology Overview
- 2 Assignment Part I
- 3 Interface Overview
- 4 Assignment Part II**

Assignment

Peer

Implement a process that will track and display a shared hash map state.

- The shared hash map is available through SharedHashMap channel.
- The updates are transmitted through UpdateEvent class.

```
import java.io.Serializable;
```

```
public class UpdateEvent implements Serializable {  
    private static final long serialVersionUID = 0xBAADBAADBAADL;  
  
    public int key;  
    public String value;  
}
```

Quiz

- How would you go about measuring the cluster throughput ?
- Will the entire cluster see the same state ?

Submission

GitLab

Use your personal GitLab repository under <https://gitlab.mff.cuni.cz/teaching/nswi163/2022>.

Requirements

- Use the assignment subdirectory.
- Write brief report in SOLUTION.md.
- Include build scripts with instructions.
- Do not commit binaries or temporary build artifacts.
- Tag your solution with task-04 and push the tag.