Hello

**JGroups:** a Java system for communicating with a group of machines

Savaş Ali TOKMEN

MSc in Computer Engineering

UFR IMA, Grenoble, FRANCE
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol
IP multicast

- Aim: resource-efficient data transmission to multiple targets
- IGMP protocol

- Radio / Video transmission, OSPF
JGroups multicast

- Initial aim: to provide a Java class for IP multicast
- Extensions:
  - Choice between protocols: UDP, TCP, JMX
  - Subscription and unsubscription notification
  - Crashed member detection
  - Reliable and ordered transmissions
  - Point-to-point messaging
  - Fragmentation of big messages
  - Scheduling policies: atomic (all or none), FIFO, total order
  - Encryption

IP multicast
JGroups multicast
JGroups API
Example
Demonstration
How JGroups works
Conclusions
Questions
References
JGroups API

- Channel creation: `new JChannel(props);` with properties such as:
  - Transport protocol
  - Reliable or non-reliable transmissions
  - Scheduling policies
- `connect` method to join the group
- `send` method for multicast or unicast sending
- `receive` method (blocking) or publish / subscribe alerts for receiving
- `disconnect` method to quit group
Example

- Request distribution system
• **Distributor's code**

```java
public void OnRequest( HttpServletRequest request ) {
    // Get the list of available servers
    Vector<Address> servers = channel.getView().getMembers();
    if( servers.size() <= 1 ){
        // In this case, there are no servers!
        OnError( 501 );
    } else {
        int gatewayPosition = servers.indexOf(channel.getLocalAddress());
        int targetServer = gatewayPosition;
        // Pick a server to process request, make sure it really
        // is a server and not the distributor
        while ( targetServer  == gatewayPosition ) {
            targetServer = random.nextInt(servers.size());
        }
        // Redirect request to server. Server will respond to client directly.
        channel.send(new Message( servers.get(targetServer), null, request ));
    }
}
```

• **Servers' code**

```java
while( true ) {
    // Wait for a request
    Object o = channel.receive(0);
    // Only process normal messages
    if( o instanceof Message ) {
        o = ((Message) o).getObject();
        // Verify that it's an HTTP request
        if( o instanceof HttpServletRequest ) {
            // Call the standard HTTP server program with request
            ProcessRequest( (HttpServletRequest) o );
        }
    }
}
```
How JGroups works

• Ethereal
Pros and perspectives

• Pros:
  - Stable system for dynamic groups
  - Very modular architecture (OSI)
  - Ships with 19 examples and 117 tests

• Open projects
  - Ethereal plugin
  - Bluetooth support
  - Compatibility with Java standards: JavaSpaces, ...
  - HTTP load balancer
  - ...
Cons

- Functionnalities not offered by JGroups but offered by rivals such as .NET:
  - Packet coalescing
  - Priorities
  - Per-message transmission options
  - Subgroups
  - Stream exchanging
  - State exchange
- JGroups doesn't have a logo!
- User's manual partially empty
Questions?

- IP multicast
- JGroups multicast
- JGroups API
- Example
- Demonstration
- How JGroups works
- Conclusions
- Questions
- References
References

- JGroups web site and links in it (mostly JBoss presentations)
- Web sites such as Wikipedia or Cisco for information about grouping and load balancing protocols in TCP/IP
- Ethereal
- Didier Donsez' lecture notes on Jini
- Microsoft's MSDN library
Thank you

Documents available on

http://scholar.alishomepage.com/Master/JGroups/