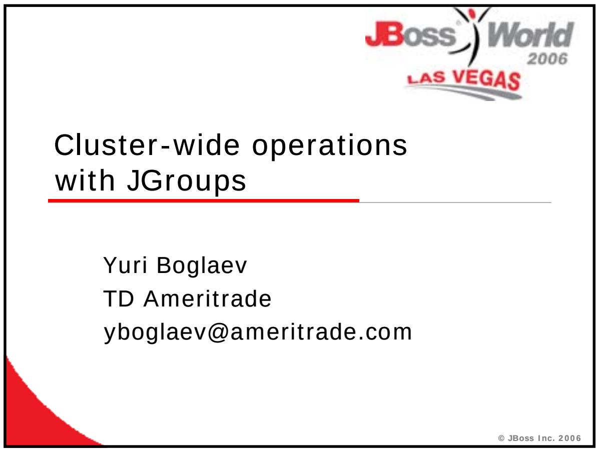
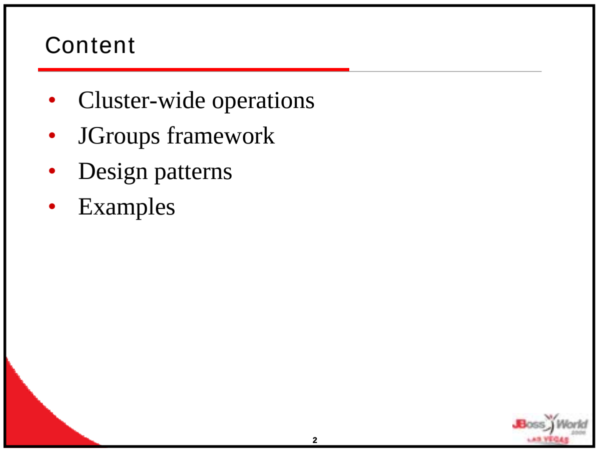


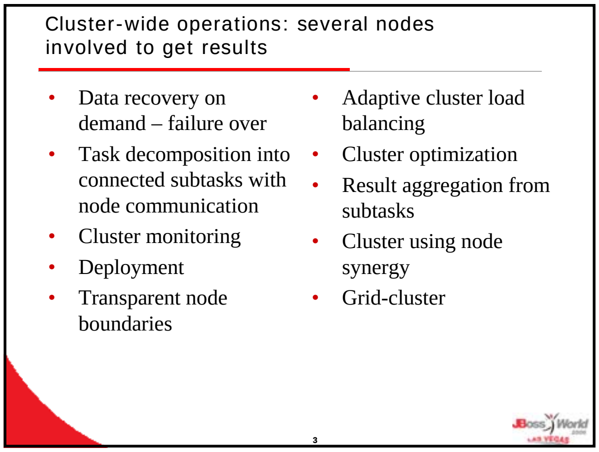
Cluster-wide operations with JGroups

Yuri Boglaev
TD Ameritrade
yboglaev@ameritrade.com



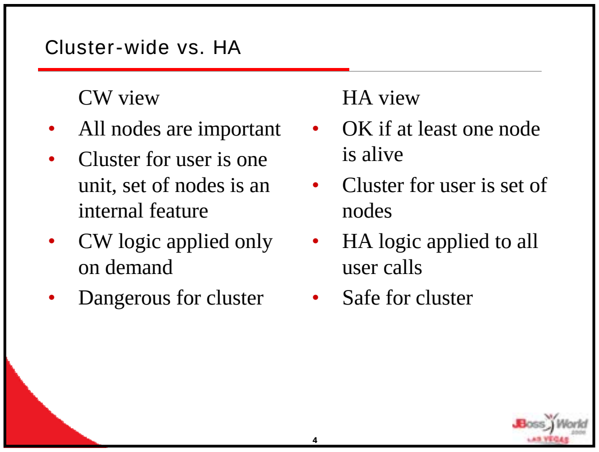
Content

- Cluster-wide operations
- JGroups framework
- Design patterns
- Examples



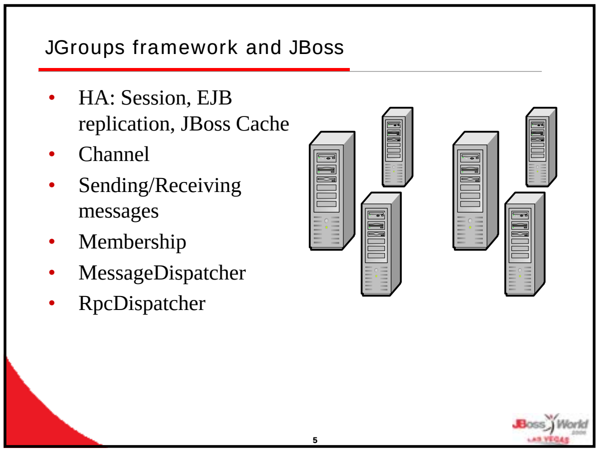
Cluster-wide operations: several nodes involved to get results

- Data recovery on demand – failure over
- Task decomposition into connected subtasks with node communication
- Cluster monitoring
- Deployment
- Transparent node boundaries
- Adaptive cluster load balancing
- Cluster optimization
- Result aggregation from subtasks
- Cluster using node synergy
- Grid-cluster



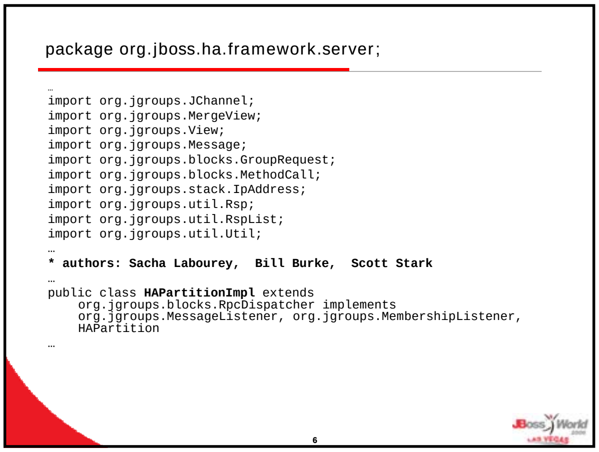
Cluster-wide vs. HA

CW view	HA view
• All nodes are important	• OK if at least one node is alive
• Cluster for user is one unit, set of nodes is an internal feature	• Cluster for user is set of nodes
• CW logic applied only on demand	• HA logic applied to all user calls
• Dangerous for cluster	• Safe for cluster



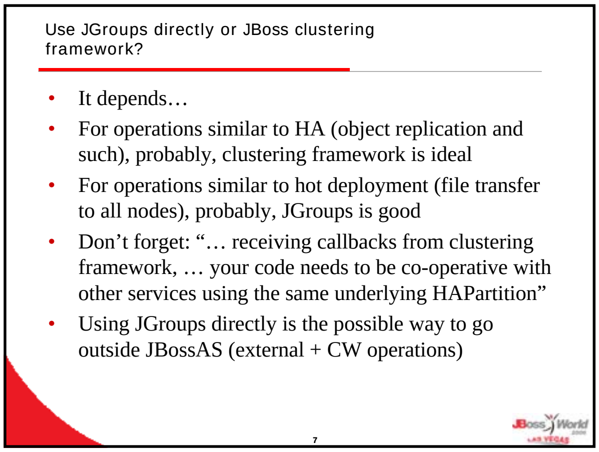
JGroups framework and JBoss

- HA: Session, EJB replication, JBoss Cache
- Channel
- Sending/Receiving messages
- Membership
- MessageDispatcher
- RpcDispatcher



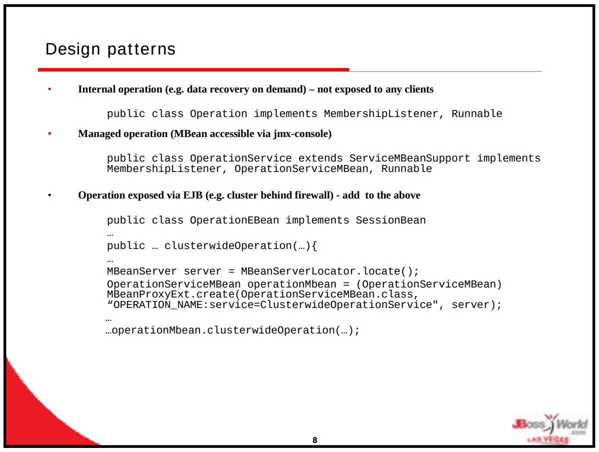
package org.jboss.ha.framework.server;

```
…
import org.jgroups.JChannel;
import org.jgroups.MergeView;
import org.jgroups.View;
import org.jgroups.Message;
import org.jgroups.blocks.GroupRequest;
import org.jgroups.blocks.MethodCall;
import org.jgroups.stack.IpAddress;
import org.jgroups.util.Rsp;
import org.jgroups.util.RspList;
import org.jgroups.util.Util;
…
* authors: Sacha Labourey,  Bill Burke,  Scott Stark
…
public class HAPartitionImpl extends
    org.jgroups.blocks.RpcDispatcher implements
    org.jgroups.MessageListener, org.jgroups.MembershipListener,
    HAPartition
…
```



Use JGroups directly or JBoss clustering framework?

- It depends…
- For operations similar to HA (object replication and such), probably, clustering framework is ideal
- For operations similar to hot deployment (file transfer to all nodes), probably, JGroups is good
- Don't forget: "… receiving callbacks from clustering framework, … your code needs to be co-operative with other services using the same underlying HAPartition"
- Using JGroups directly is the possible way to go outside JBossAS (external + CW operations)



Design patterns

- **Internal operation (e.g. data recovery on demand) – not exposed to any clients**

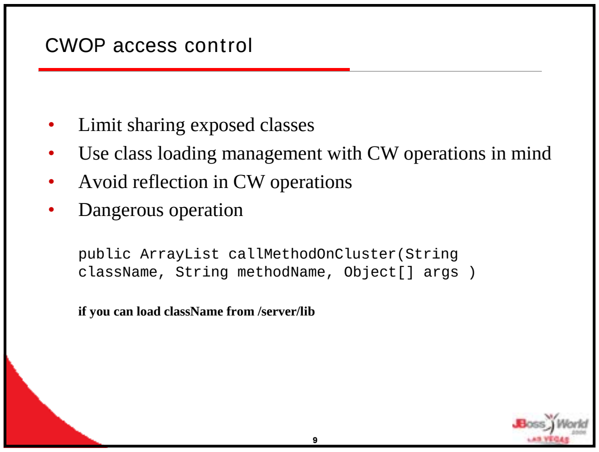
```
public class Operation implements MembershipListener, Runnable
```

- **Managed operation (MBean accessible via jmx-console)**

```
public class OperationService extends ServiceMBeanSupport implements
MembershipListener, OperationServiceMBean, Runnable
```

- **Operation exposed via EJB (e.g. cluster behind firewall) - add to the above**

```
public class OperationEBean implements SessionBean
…
public … clusterwideOperation(…){
…
MBeanServer server = MBeanServerLocator.locate();
OperationServiceMBean operationMbean = (OperationServiceMBean)
MBeanProxyExt.create(OperationServiceMBean.class,
"OPERATION_NAME:service=ClusterwideOperationService", server);
…
…operationMbean.clusterwideOperation(…);
```

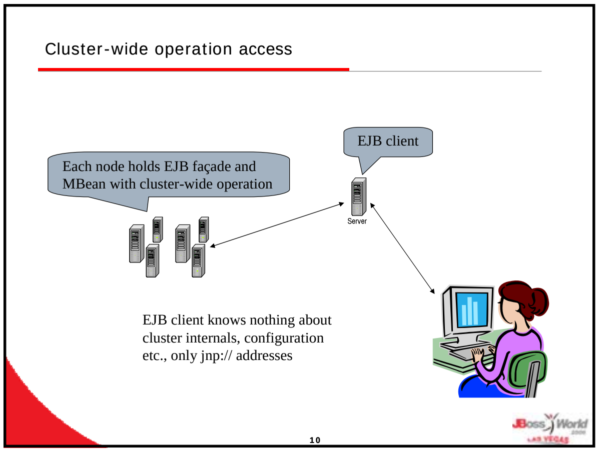


CWOP access control

- Limit sharing exposed classes
- Use class loading management with CW operations in mind
- Avoid reflection in CW operations
- Dangerous operation

```
public ArrayList callMethodOnCluster(String
className, String methodName, Object[] args )
```

if you can load className from /server/lib

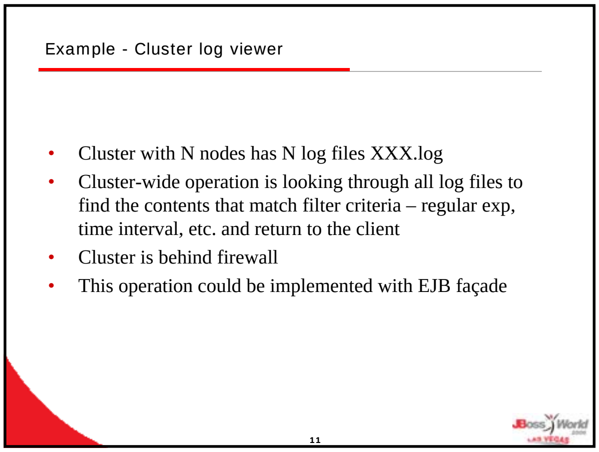


Cluster-wide operation access

EJB client

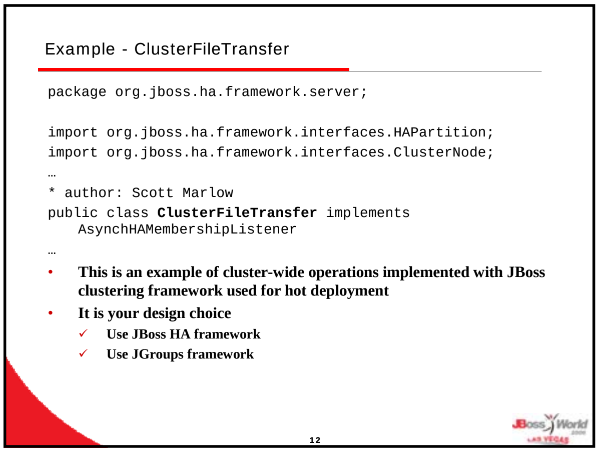
Each node holds EJB façade and MBean with cluster-wide operation

EJB client knows nothing about cluster internals, configuration etc., only jnp:// addresses



Example - Cluster log viewer

- Cluster with N nodes has N log files XXX.log
- Cluster-wide operation is looking through all log files to find the contents that match filter criteria – regular exp, time interval, etc. and return to the client
- Cluster is behind firewall
- This operation could be implemented with EJB façade



Example - ClusterFileTransfer

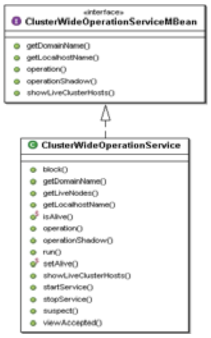
```
package org.jboss.ha.framework.server;

import org.jboss.ha.framework.interfaces.HAPartition;
import org.jboss.ha.framework.interfaces.ClusterNode;
…
* author: Scott Marlow
public class ClusterFileTransfer implements
    AsynchHAMembershipListener
…
```

- **This is an example of cluster-wide operations implemented with JBoss clustering framework used for hot deployment**
- **It is your design choice**
  - ✓ **Use JBoss HA framework**
  - ✓ **Use JGroups framework**

Example – Data recovery on demand

- All nodes get data streaming from an external source in sync until connection or the source is broken or a node is down
- Continuous data replication is not feasible (large volume) and has no sense till failure occurs (rare case)
- CW operation is a data recovery when this event is trapped
- Operation steps:
  - ✓ Discover a node in good state
  - ✓ Get missed data
  - ✓ Merge data

Cluster-wide operation service



CW operation implementation -1

```
package org.jboss.clusterwide.invocation;
…
import org.jboss.system.ServiceMBeanSupport;
import org.jgroups.Address;
import org.jgroups.Channel;
import org.jgroups.JChannel;
import org.jgroups.MembershipListener;
import org.jgroups.View;
import org.jgroups.blocks.PullPushAdapter;
import org.jgroups.util.Util;

…
public class ClusterWideOperationService extends ServiceMBeanSupport implements
Runnable, MembershipListener, ClusterWideOperationServiceMBean
{

    /**
     * @jmx.managed-operation description="operation from remote caller"
     *
     */
    public boolean operationShadow(String className, String operationName,
                                                        String param)
    {

    }
```

CW operation implementation -2

```
/**
 * @jmx.managed-operation description="operation from remote caller"
 *
 */
public boolean operation(String className, String operationName, String param)
{
      …

      serv = (MBeanServerConnection) ctx.lookup("jmx/invoker/RMIAdaptor");

      …

      Boolean resultOp =(Boolean) serv.invoke(new ObjectName(
      "org.jboss.clusterwide:service=ClusterWideOperationService"),
      "operationShadow",  new Object[] { className, operationName, param },
       new String[]{ String.class.getName(), String.class.getName(),
                                 String.class.getName()});
}
```

Conclusion

- Cluster-wide operations are very common tasks in server side development
- Although cluster-wide operations could be implemented under HA, they seem to be coming from another dimension
- They could be included in JBoss development either using JGroups directly or creating separate framework on the top of JGroups
- We have an excellent common denominator – JGroups, let’s use it in one way or another