ORACLE®

Coherence 12.1.2 – Hidden Gems

Harvey Raja
Principal Member Technical Staff
Oracle Coherence

Coherence Roadmap

Apr 2011

Coherence 3.7.0

- Automatic Proxy Discovery for Clients
- Auto Re-connect for Clients
- Dynamic Load Balancing for Clients
- XML Schema for Config
- Load Balancer Integration (F5)
- Native Coherence*Web Glassfish Integration
- Query Monitoring
- Partition-Level Transactions
- Elastic Data

Sept 2011

Coherence 3.7.1

- POF Enhancements
- Query Explain Plan
- REST API
- Pluggable Partitioning Schemes
- Elastic Data Improvements (Journal-based Flash Storage)
- Delta Backups
- Leverage Exalogic Exabus Technology

NOW!!

Coherence 12c (12.1.2)

- Golden Gate Adapter for Coherence
- REST security and usability improvements
- Live Events
- Configuration Modernization
- Asynchronous Backups
- Backup Management Improvements
- Maven Support
- Exalogic performance optimizations
- Coherence Container
- Dynamic Thread Pooling for Proxy Servers
- OUI/Opatch Integration
- ECID Support
- OSGi Support

ORACLE

Coherence 12.1.2

- New major release of Coherence 12c
- Key Themes
  - Container Management with WLS
  - Continued Investment in Exalogic
  - Database Synchronization
  - Configuration and Usability Improvements
  - Oracle Fusion Middleware Convergence

2013

Coherence 12c (12.1.2)

- Golden Gate Adapter for Coherence
- REST security and usability improvements
- Live Events
- Configuration Modernization
- Asynchronous Backups
- Backup Management Improvements
- Maven Support
- Exalogic performance optimizations
- Coherence Container
- Dynamic Thread Pooling for Proxy Servers
- OUI/Opatch Integration
- ECID Support
- OSGi Support

ORACLE

A Deeper Look

- Asynchronous Backups

- Backup Management Improvements

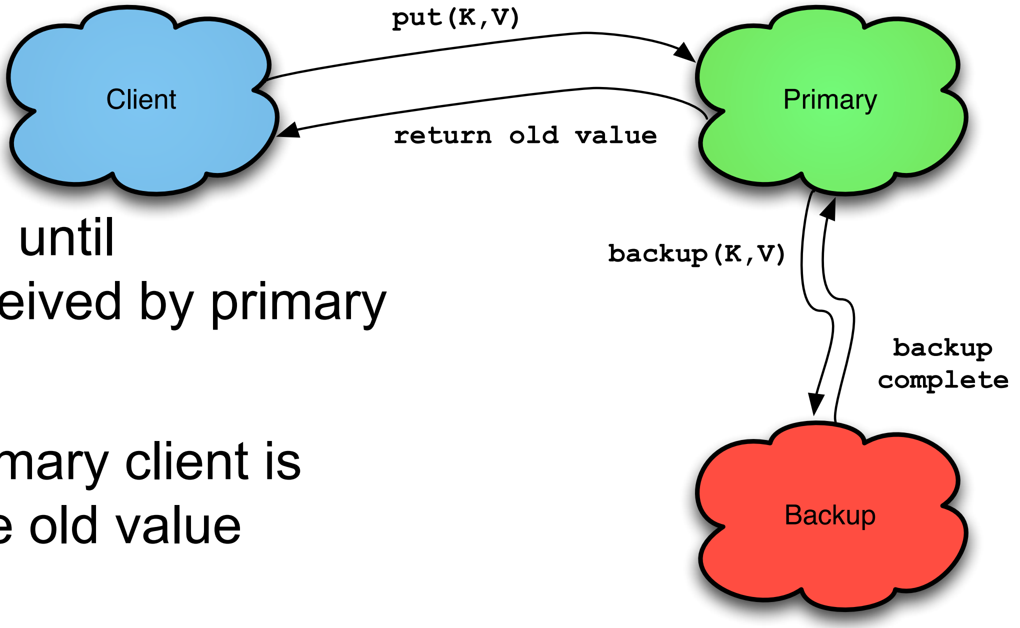
Hidden Gems

- NameService
- TransactionEvent in Live Events
- POF Configuration Generator
- BinaryEntry synthetic ops
- Poll logging

- TcpRing improvements
- SLF4J native support
- WKA address resolution carried out on separate thread
- Preprocessing
- NearCache invalidation strategy

Current Backup Approach

- A put request from the client is entirely synchronous
- Client thread blocked until backup message received by primary
- Once received by primary client is responded to with the old value
- Provides consistency guarantees



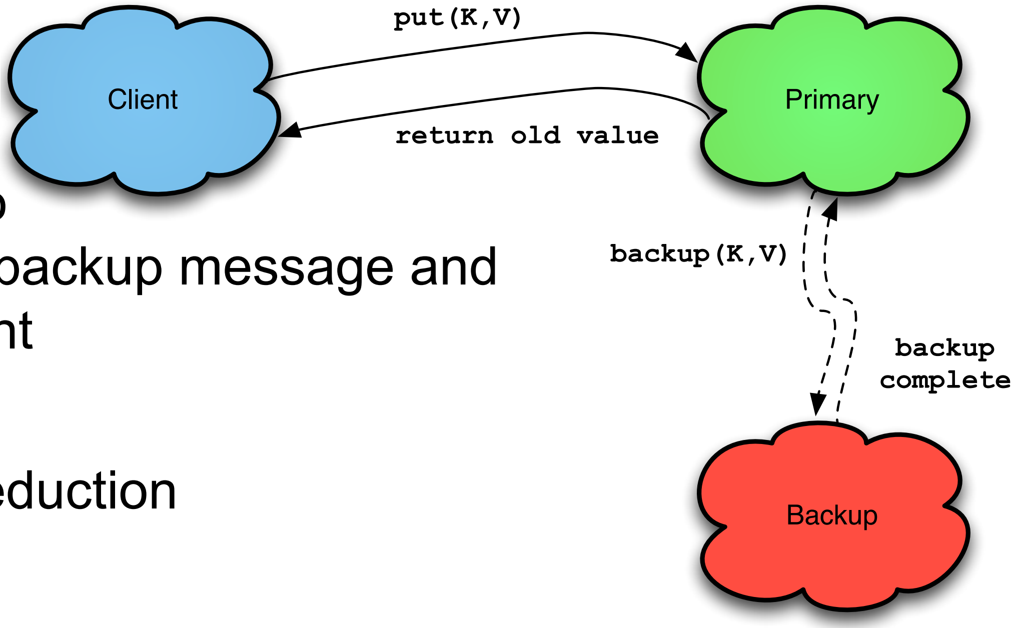
12c Backup Approach (Asynchronous)

- Opt-in strategy

```
<distributed-scheme>
    <async-backup>true</async-backup>
</distributed-scheme>
```

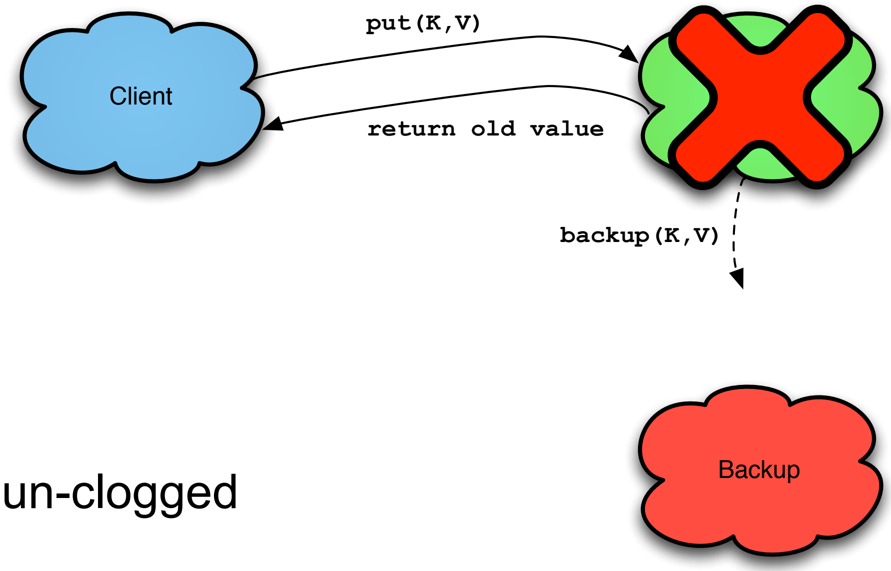
- Client request sent to primary & in parallel backup message and client response is sent

- ~40 – 50% latency reduction



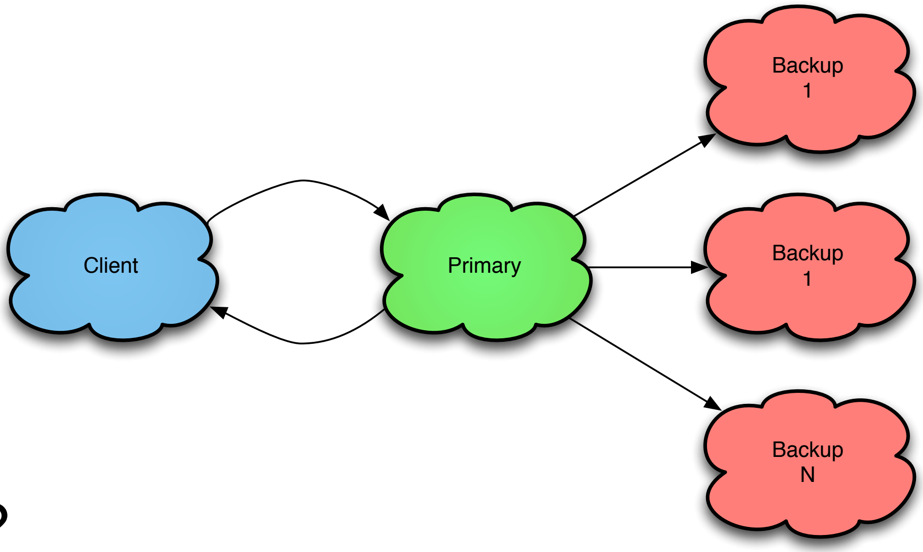
12c Backup Approach (Asynchronous)

- Speed vs Consistency
- Adaptive switch back to synchronous
  - Based on partition load
  - Falls back to async once un-clogged

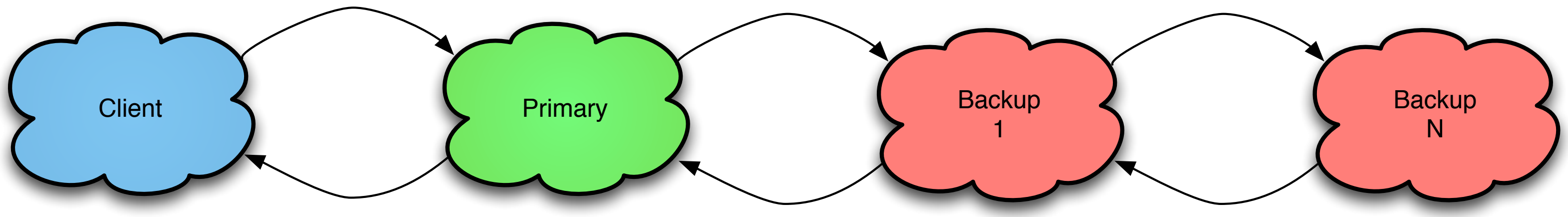


Pre-12c Backup Management

- Fan-out approach
- Requires coordination
- What if coordinator leaves?
  - Which backup received the message?



12c Backup Management



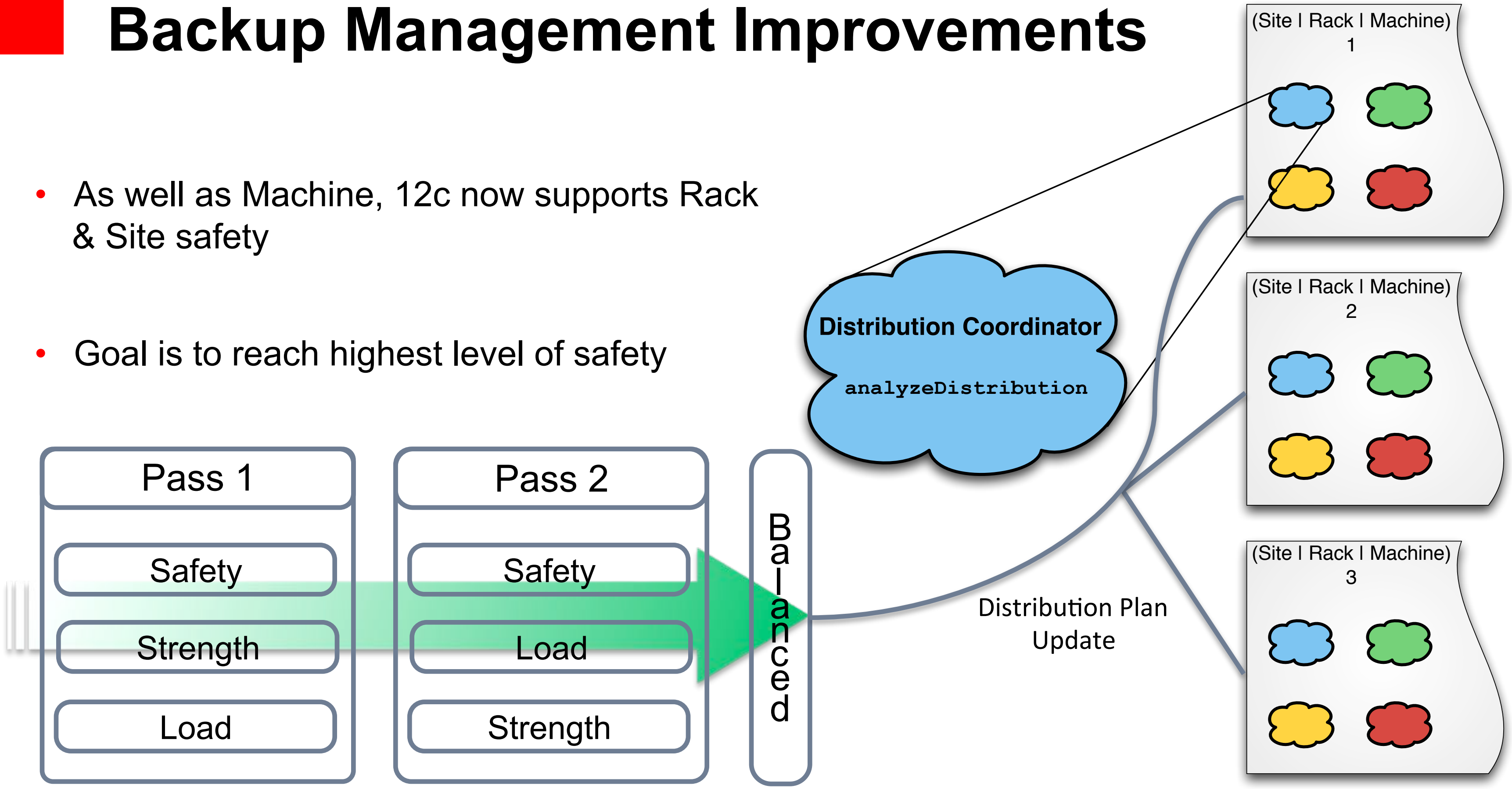
- Chained mechanics ensures prior node in the chain is responsible for ensuring delivery

- Departure of node in the chain ensures they are skipped

- Pre-process response from forwarded message

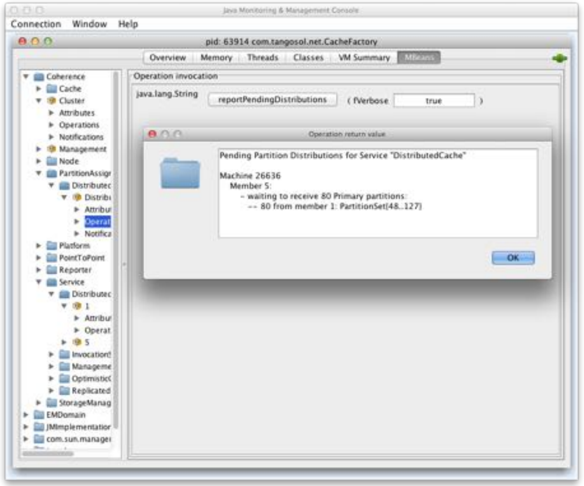
Backup Management Improvements

- As well as Machine, 12c now supports Rack & Site safety
- Goal is to reach highest level of safety



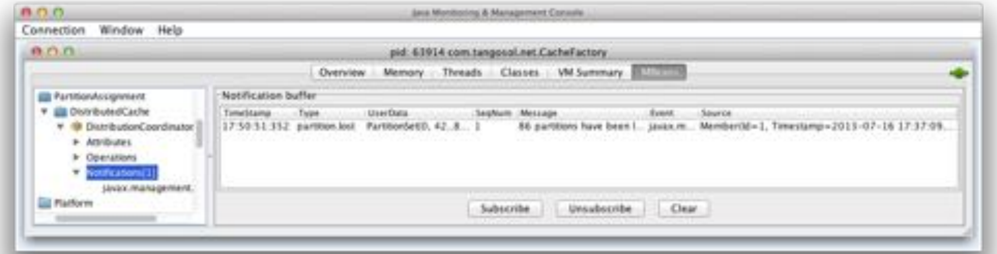
Backup Management Improvements

- Mirroring Strategy
  - Attempt to mirror assignments of another service
- JMX Enhancements
  - Pending distributions
  - PartitionLost JMX Notification



```
Pending Partition Distributions for Service "DistributedCache"

Machine 26636
  Member 5:
    - waiting to receive 80 Primary partitions:
      -- 80 from member 1: PartitionSet[48..127]
```



TimeStamp	Type	UserData	SeqNum	Message	Event	Source
17:50:31:332	partition.lost	PartitionSet[0, 42..8...	1	86 partitions have been l...	javax.m...	MemberId=1, Timestamp=2013-07-16 17:37:09...

Coherence *Extend - NameService

Apr 2010

Coherence 3.6

- List of InetAddresses on the client

Apr 2011

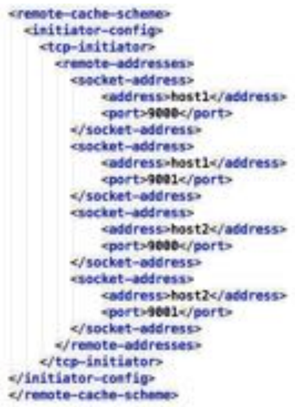
Coherence 3.7.0

- Automatic Proxy Discovery for Clients
- Dynamic Load Balancing for Clients

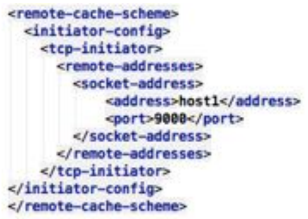
2013

Coherence 12c (12.1.2)

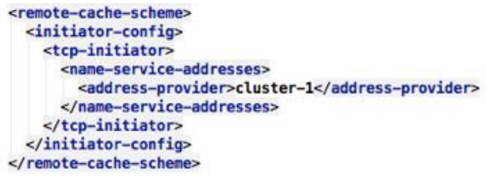
- NameService



```
<remote-cache-scheme>
  <initiator-config>
    <tcp-initiator>
      <remote-addresses>
        <socket-address>
            <address>host1</address>
            <port>9000</port>
        </socket-address>
        <socket-address>
            <address>host1</address>
            <port>9001</port>
        </socket-address>
        <socket-address>
            <address>host2</address>
            <port>9000</port>
        </socket-address>
        <socket-address>
            <address>host2</address>
            <port>9001</port>
        </socket-address>
      </remote-addresses>
    </tcp-initiator>
</initiator-config>
</remote-cache-scheme>
```



```
<remote-cache-scheme>
  <initiator-config>
    <tcp-initiator>
      <remote-addresses>
        <socket-address>
            <address>host1</address>
            <port>9000</port>
        </socket-address>
      </remote-addresses>
    </tcp-initiator>
</initiator-config>
</remote-cache-scheme>
```

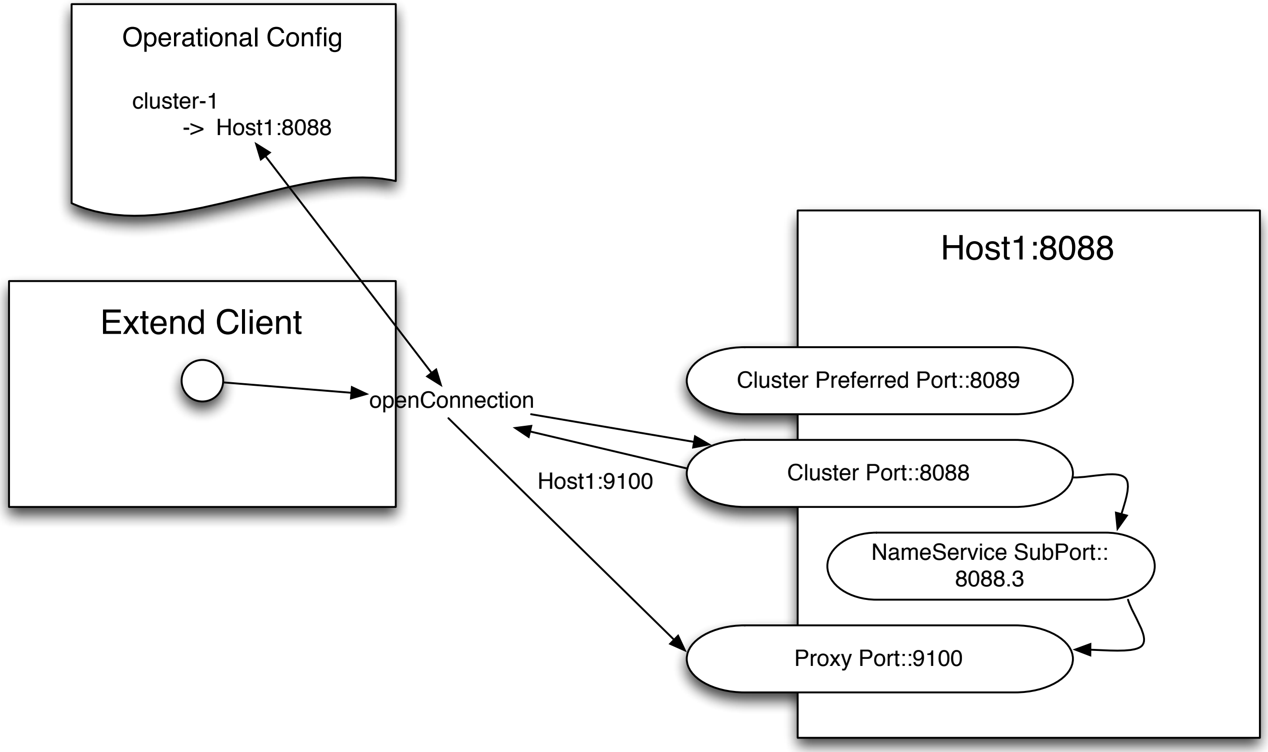


```
<remote-cache-scheme>
  <initiator-config>
    <tcp-initiator>
      <name-service-addresses>
        <address-provider>cluster-1</address-provider>
      </name-service-addresses>
    </tcp-initiator>
  </initiator-config>
</remote-cache-scheme>
```

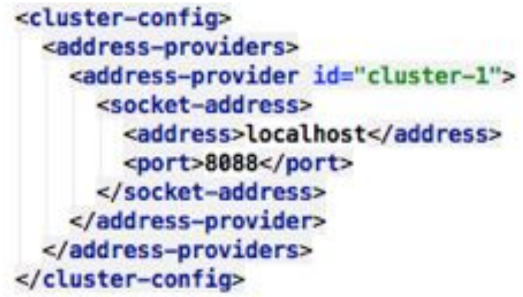
Coherence *Extend - NameService

- Allows discovery of proxies using logical names (service names)
- Handshake takes place during connection establishment
- Uses well known cluster sub-port to discover proxy end points
- Could be considered as ClusterNameService (CNS ≈ DNS)

Coherence *Extend - NameService

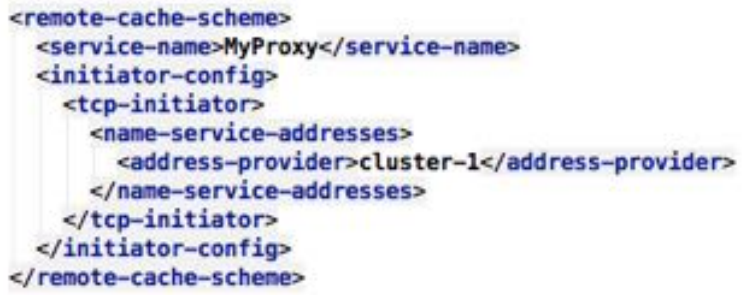


Coherence *Extend - NameService

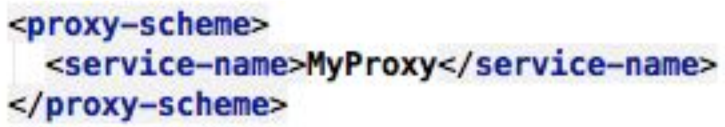


```
<cluster-config>
  <address-providers>
    <address-provider id="cluster-1">
      <socket-address>
        <address>localhost</address>
        <port>8088</port>
      </socket-address>
    </address-provider>
  </address-providers>
</cluster-config>
```

- Service name must be the same
- Discovers all proxy addresses for the same service name
- Proxy service can use ephemeral ports



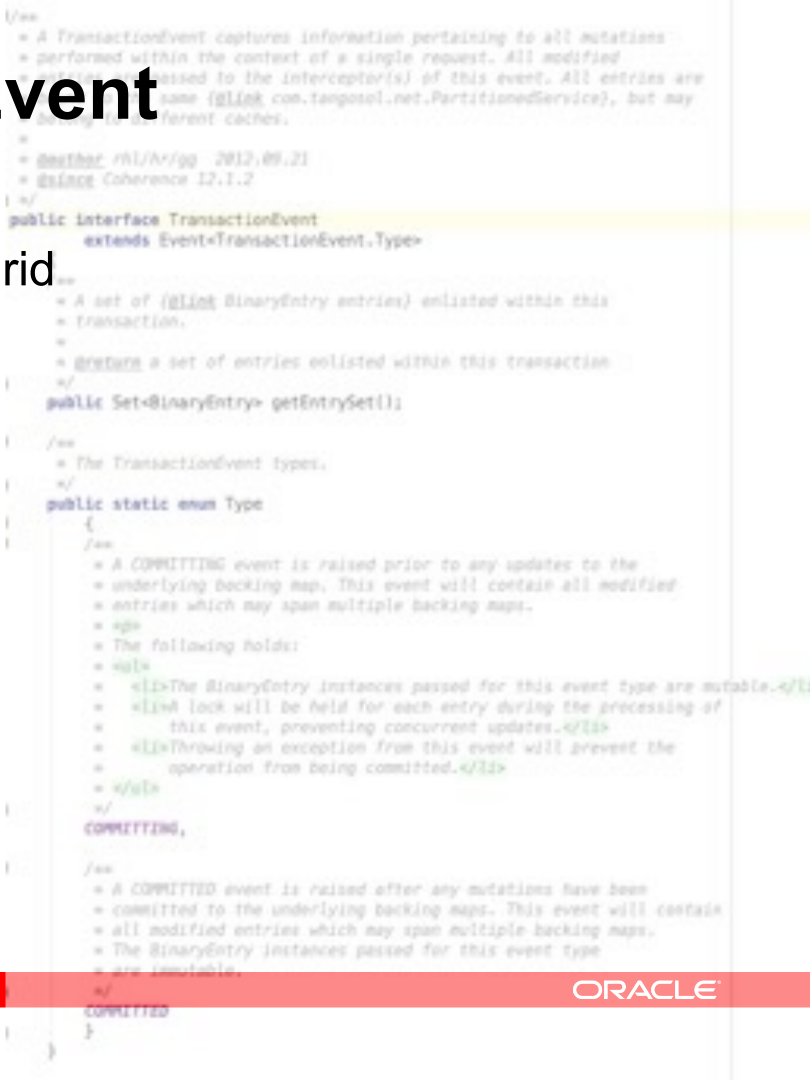
```
<remote-cache-scheme>
  <service-name>MyProxy</service-name>
  <initiator-config>
    <tcp-initiator>
      <name-service-addresses>
        <address-provider>cluster-1</address-provider>
      </name-service-addresses>
    </tcp-initiator>
  </initiator-config>
</remote-cache-scheme>
```



```
<proxy-scheme>
  <service-name>MyProxy</service-name>
</proxy-scheme>
```

LiveEvents – TransactionEvent

- Ability to intercept events as they occur in the grid
  - Fine grained events with logical causality
- 12.1.2 introduces a TransactionEvent
  - Partition Lite transaction event
- Receives all enlisted entries in a single event
- Can enlist more entries



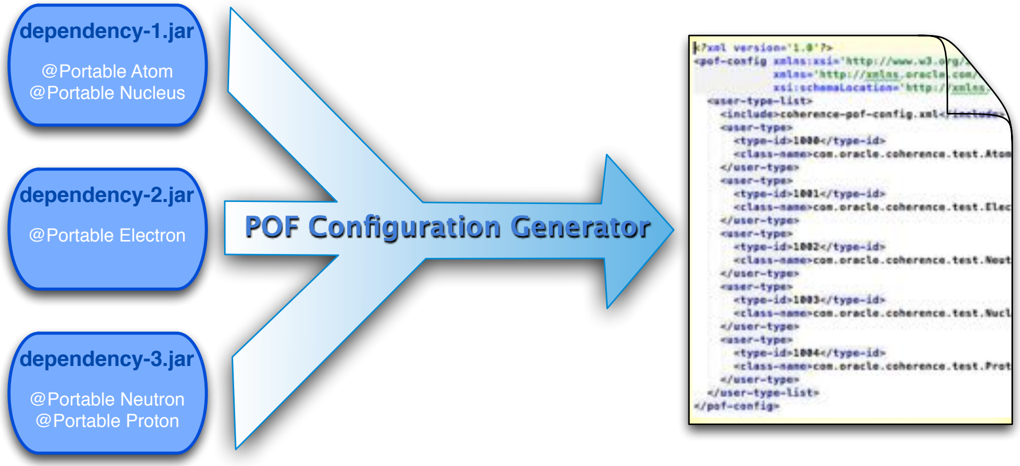
```
/**
 * A TransactionEvent captures information pertaining to all mutations
 * performed within the context of a single request. All modified
 * entries are passed to the interceptor(s) of this event. All entries are
 * bound to the same {@link com.tangosol.net.PartitionedService}, but may
 * belong to different caches.
 *
 * @author rhl/hr/gg  2012.09.21
 * @since Coherence 12.1.2
 */
public interface TransactionEvent
        extends Event<TransactionEvent.Type>
    {
    /**
     * A set of {@link BinaryEntry entries} enlisted within this
     * transaction.
     *
     * @return a set of entries enlisted within this transaction
     */
    public Set<BinaryEntry> getEntrySet();

    /**
     * The TransactionEvent types.
     */
    public static enum Type
        {
        /**
         * A COMMITTING event is raised prior to any updates to the
         * underlying backing map. This event will contain all modified
         * entries which may span multiple backing maps.
         * <p>
         * The following holds:
         * <ul>
         *   <li>The BinaryEntry instances passed for this event type are mutable.</li>
         *   <li>A lock will be held for each entry during the processing of
         *       this event, preventing concurrent updates.</li>
         *   <li>Throwing an exception from this event will prevent the
         *       operation from being committed.</li>
         * </ul>
         */
        COMMITTING,

        /**
         * A COMMITTED event is raised after any mutations have been
         * committed to the underlying backing maps. This event will contain
         * all modified entries which may span multiple backing maps.
         * The BinaryEntry instances passed for this event type
         * are immutable.
         */
        COMMITTED
        }
    }
```

POF Configuration Generator

- Generates POF configuration file based on `@Portable` classes
- Predictable type-id generation
- Generational
  - Accepts previous POF configuration file
- Operates against a GAR and supported by maven GAR plugin



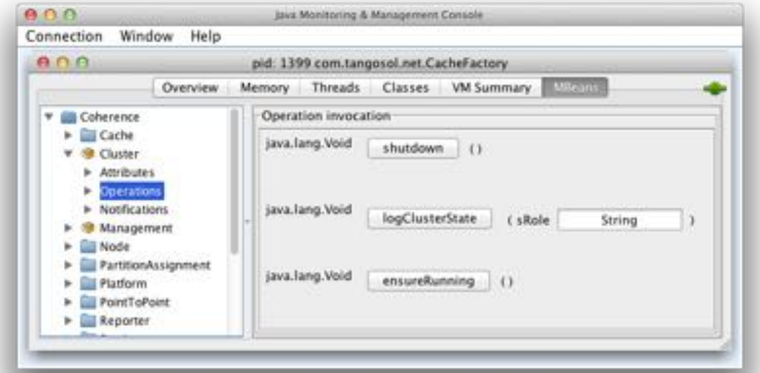
```
<?xml version='1.0'?>
<pof-config xmlns:xsi='http://www.w3.org/
            xmlns='http://xmlns.oracle.com/
            xsi:schemaLocation='http://xmlns.
  <user-type-list>
    <include>coherence-pof-config.xml</include>
    <user-type>
      <type-id>1000</type-id>
      <class-name>com.oracle.coherence.test.Atom
    </user-type>
    <user-type>
      <type-id>1001</type-id>
      <class-name>com.oracle.coherence.test.Elec
    </user-type>
    <user-type>
      <type-id>1002</type-id>
      <class-name>com.oracle.coherence.test.Neut
    </user-type>
    <user-type>
      <type-id>1003</type-id>
      <class-name>com.oracle.coherence.test.Nucl
    </user-type>
    <user-type>
      <type-id>1004</type-id>
      <class-name>com.oracle.coherence.test.Prot
    </user-type>
  </user-type-list>
</pof-config>
```

12c - Loose Change

- BinaryEntry synthetic operations
  - updateBinaryValue(Binary, boolean) // new
  - setValue(Object, boolean)
  - remove(boolean)
- SLF4J logger
  - SLF4J libraries must be on classpath and logger destination be slf4j
- WKA host name lookup now on a separate thread
- NearCache default invalidation strategy is now PRESENT

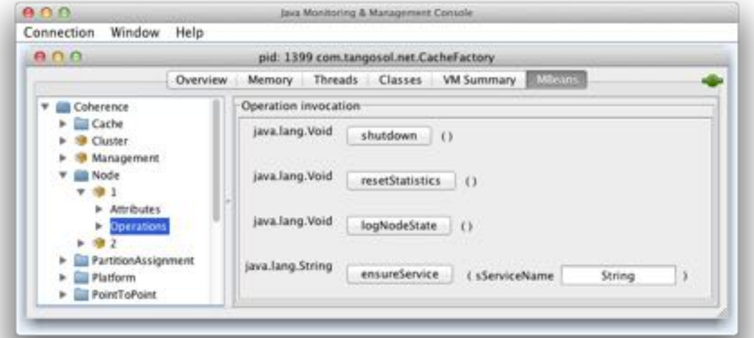
12c - Loose Change

- Log Cluster State
  - Role based
  - Distributed Thread Dump
  - Includes outstanding Polls



Java Monitoring & Management Console
Connection Window Help
pid: 1399 com.tangosol.net.CacheFactory
Overview Memory Threads Classes VM Summary MBeans
Coherence
Cache
Cluster
Attributes
Operations
Notifications
Management
Node
PartitionAssignment
Platform
PointToPoint
Reporter
Operation invocation
java.lang.Void shutdown ()
java.lang.Void logClusterState (sRole String)
java.lang.Void ensureRunning ()

- Log Node State
  - Node Thread dump
  - Includes outstanding Polls



Java Monitoring & Management Console
Connection Window Help
pid: 1399 com.tangosol.net.CacheFactory
Overview Memory Threads Classes VM Summary MBeans
Coherence
Cache
Cluster
Management
Node
1
Attributes
Operations
2
PartitionAssignment
Platform
PointToPoint
Operation invocation
java.lang.Void shutdown ()
java.lang.Void resetStatistics ()
java.lang.Void logNodeState ()
java.lang.String ensureService (sServiceName String)

- Responsibility MBeans

12c - Loose Change

- TcpRing improvements
  - Death broadcast is communicated across the cluster
- Message preprocessing where possible

Demo

Q & A

Hardware and Software

ORACLE

Engineered to Work Together

ORACLE®