
ORACLE®

ORACLE®

Coherence*Extend Best Practices

Jason Howes
Consulting Member Technical Staff, Oracle Coherence

Agenda

- Coherence*Extend Overview
- Best Practices
  - Architecture
  - Configuration
  - Deployment
  - Monitoring
- Questions

A Bit About Me

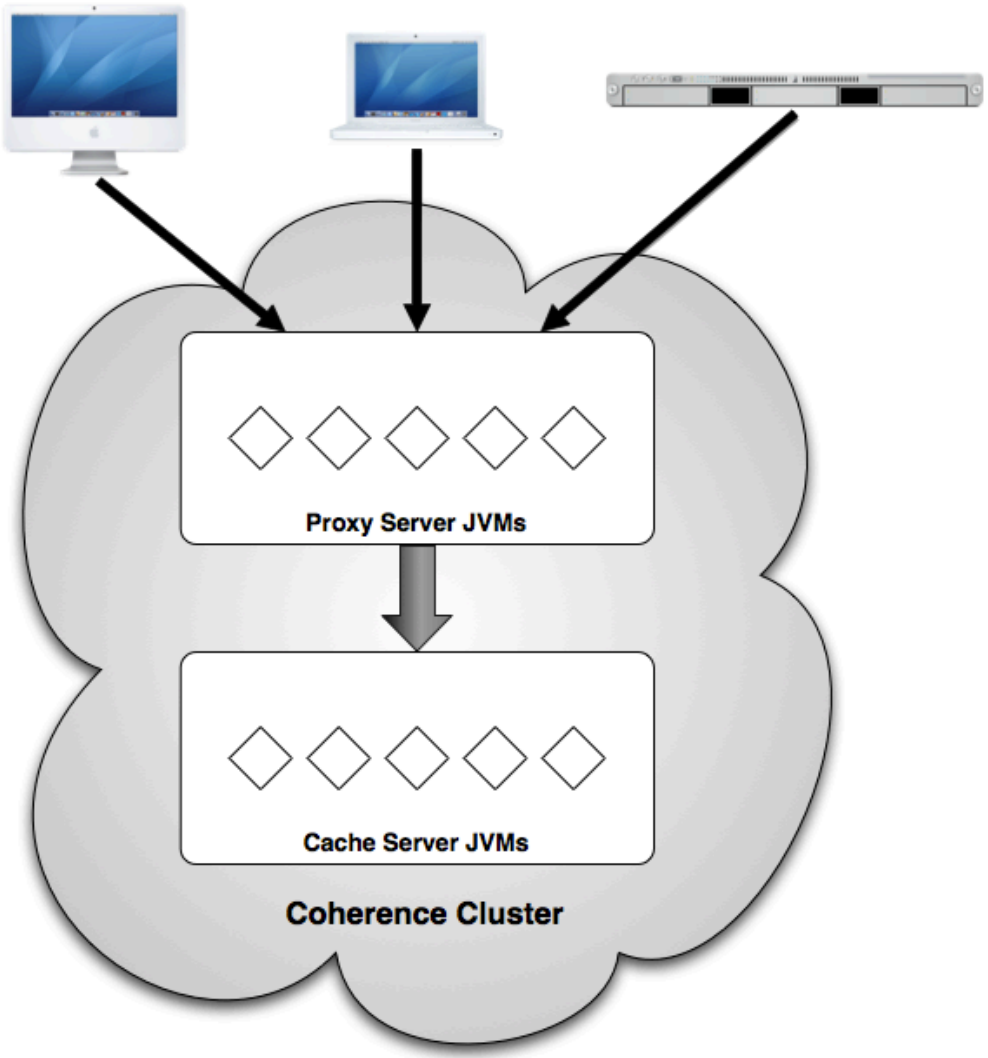
- Lead architect for Coherence*Extend at Tangosol
  - *Coherence*Extend-JMS*
  - Coherence*Extend-TCP
    - Java client
    - .NET client
- Been with Oracle since 2007
  - Coherence*Extend-TCP for C++
  - Coherence*Extend-HTTP (REST API)

Coherence*Extend Overview

What is Coherence*Extend?

- A feature of Coherence that allows non-clustered clients to access clustered services
  - Caching
  - Invocation
  - Aggregation/Processing
- Java/C++/.NET/REST clients
- Portable serialization format (POF)
- Cluster “bridges” (replication, etc.)

The Big Picture



Best Practices: Architecture

When should you use Coherence*Extend?

- C++/.NET applications
- Short-lived processes
- Unmanaged or under-provisioned hardware
- Access to multiple clusters from a single process
- Access to clustered services:
  - From outside a firewall
  - From 1000s of application instances
  - From clients that use different Coherence versions
  - Across a high latency, unreliable, or untrusted network

Leverage Grid Aggregation and Processing

- Use EntryAggregators to aggregate large data sets
- Don’t pull the data set to the client!
- Abuse case: pull 10 MB of order data to the client to calculate an average price

- Use EntryProcessors to update large data sets
- Don’t update the data set on the client!
- Abuse case: pull 10000 orders to the client, change one property of the orders, and then push them back

Leverage Near and Continuous Query Caching

- Local cache of frequently requested clustered data (key or query-based)
- Leverage Near and Continuous Query caches on your clients whenever appropriate
- Reduces Proxy and Cache Server CPU utilization
- Reduces network utilization
- In general, use either “none” or “present” Near Cache invalidation strategy

Leverage POF Serialization

- Avoids deserialization in the grid for most operations
- Eliminates the need to deploy data classes in the grid
- Reduces memory consumption, CPU utilization, and request latency
- Helps future proof your application
  - Different languages
  - Data class evolution

Best Practices: Configuration

Client Configuration

- Configure more than one Proxy Server address
- Enable heartbeats
- Set a request timeout
- Configure an identity for the client
- Leverage system properties

```
<remote-addresses>
  ...
  <socket-address>
    <address system-property="tangosol.coherence.extend.address">localhost</address>
    <port system-property="tangosol.coherence.extend.port">9099</port>
  </socket-address>
  ...
</remote-addresses>
```

Proxy Server Configuration

- Disable local storage
  - `-Dtangosol.coherence.distributed.localstorage=false`
  - In general, no Near or Continuous Query Caches
- Enable JMX
- Enable heartbeats
- Set `SO_REUSEADDR` to true
- Configure an appropriate size worker thread pool
- Configure the same type of serializer used by clients for clustered cache services

Proxy Server Configuration

- On some version of Windows, configuring a 128kb TCP/IP send buffer improves performance*
- Leverage system properties

```
<proxy-scheme>
  ...
  <thread-count system-property="tangosol.coherence.extend.threads">20</thread-count>
  <acceptor-config>
    <tcp-acceptor>
      <local-address>
        <address system-property="tangosol.coherence.extend.address">0.0.0.0</address>
        <port system-property="tangosol.coherence.extend.port">9099</port>
      </local-address>
      <reuse-address>true</reuse-address>
    </tcp-acceptor>
  ...
  <autostart system-property="tangosol.coherence.extend.enabled">true</autostart>
</proxy-scheme>
```

* Your mileage may vary

Best Practices: Deployment

Deployment

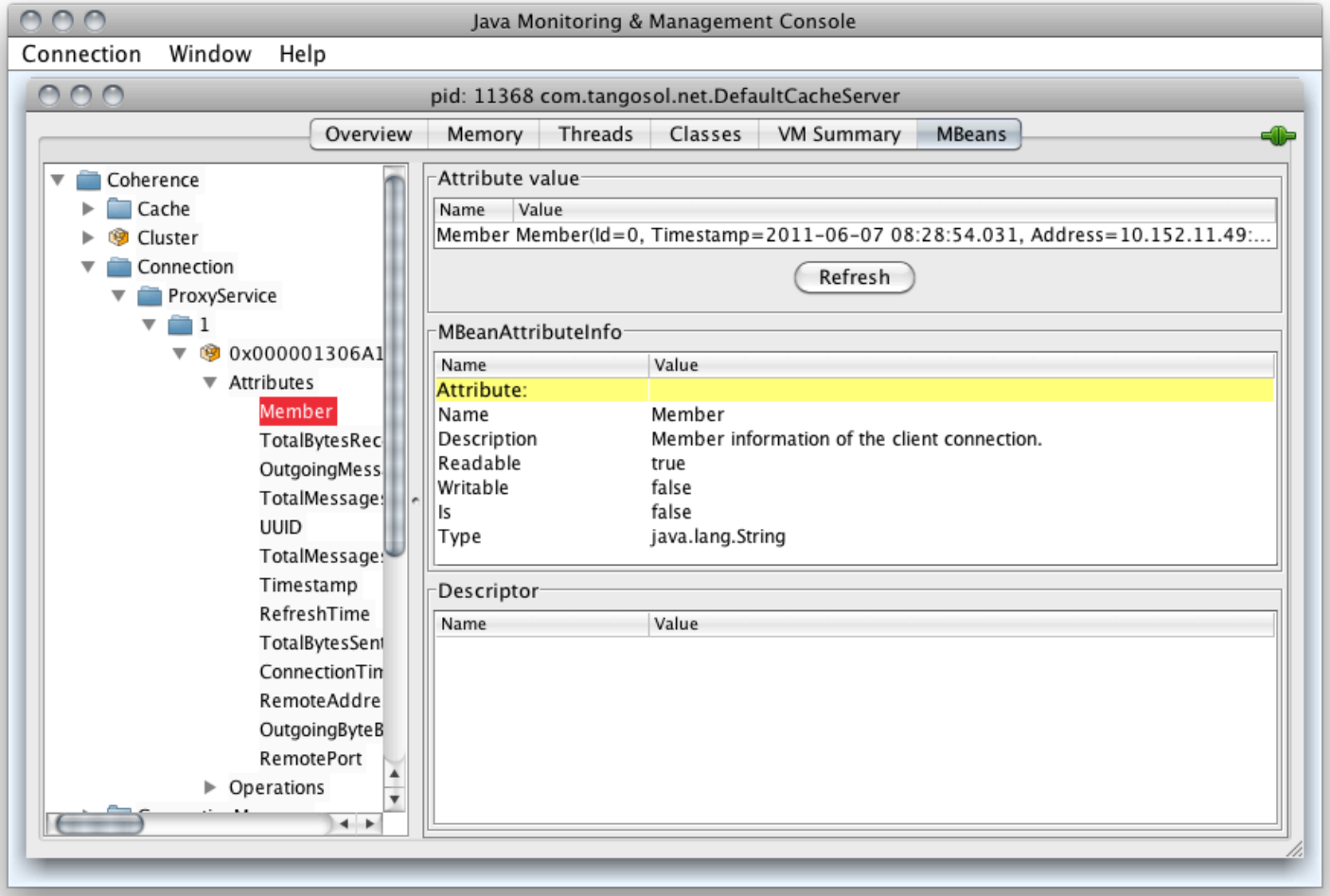
- Leverage a TCP/IP load balancer
  - Built in Proxy Server software LB
  - Hardware LB such as F5
- Scale your Proxy Service tier appropriately
  - Number of clients
  - Size and frequency of requests
  - Size and frequency of updates (passive clients)
  - Horizontal and vertical scale out

Deployment

- Collocate a Proxy Server with server-class C++ and .NET applications
  - Removes one network hop
  - Consider running multiple per machine
  - Use “client” load balancer policy
- Leverage the backwards compatibility of the Coherence*Extend protocol
  - Older clients can connect to newer Proxy Servers
  - Supports a mix of client versions
  - Allows you to upgrade your clients incrementally

Best Practices: Monitoring

JMX and JConsole



JMX and JConsole

- ServiceMBean for the ProxyService:
  - TaskAverageDuration
  - TaskBacklog and TaskMaxBacklog
  - ThreadAverageActiveCount
- ConnectionManager for the ProxyService:
  - OutgoingByteBacklog
  - OutgoingMessageBacklog
- ConnectionMBean for an individual client connection:
  - Member
  - OutgoingByteBacklog
  - OutgoingMessageBacklog

Log Messages

- Indicates a misconfigured serializer:

```
The serializer used by cache "…" (…) is incompatible with the
serializer configured for service "…" (…); therefore, cached keys and
values will be converted via serialization. This will result in
increased CPU and memory utilization. If possible, consider
reconfiguring either serialize
```

- Indicates the use of a cache that doesn't support the "pass-through" serialization optimization:

```
The cache "…" does not support pass-through optimization for objects
in internal format. If possible, consider using  a different cache
topology.
```

- Indicates a "rogue" client or over utilized Proxy Server (CPU, network, etc.):

```
Extend*TCP has determined that TcpConnection(…) must be closed to
maintain system stability: ...
```

Questions?

For More Information

Coherence:

http://www.oracle.com/technology/products/coherence/

Coherence Discussion Forums:

http://forums.oracle.com/forums/forum.jspa?forumID=480

Coherence Examples:

http://coherence.oracle.com/display/EXAMPLES

The Coherence Incubator:

http://coherence.oracle.com/display/INCUBATOR