

Uncommon Sense 2

A salmagundi of Coherence development tips and techniques

David Whitmarsh

April 4, 2014

Outline

- 1 Spring Recap
- 2 POF Tricks
- 3 NUMA
- 4 Cache Configuration
- 5 Operational Configuration

Recipe: Spring JMX Exporter

```
public class SpringCoherenceJMXExporter extends MBeanExporter {
    @Override
    protected void doRegister(Object mbean, ObjectName
        objectName)
        throws JMException {
        Registry registry =
            CacheFactory.ensureCluster().getManagement();
        String sname =
            registry.ensureGlobalName(objectName.getCanonicalName());
        registry.register(sname, mbean);
    }
}
```

Export Spring JMX MBeans Coherence JMX Node

- custom-mbeans.xml patterns are scanned once
- scan happens during Coherence startup
- MBeans with a dependency on Coherence will not yet exist
- SpringCoherenceJMXExporter exports everything according to Spring JMX configuration

Test Object Type

- How do you determine the type of a POF serialised object in cache?
- ClassFilter?
- It isn't serializable or Portable

Type ID Filter

```
public class SimpleTypeIdFilter implements EntryFilter {  
  
    public SimpleTypeIdFilter(  
        this.typeId = typeId;  
    }  
  
    public boolean evaluateEntry(Entry entry) {  
        BinaryEntry binEntry = (BinaryEntry) entry;  
  
        PofContext ctx = (PofContext) binEntry.getSerializer();  
  
        PofValue value =  
            PofValueParser.parse(binEntry.getBinaryValue(),  
                ctx);  
  
        int valueType = value.getTypeId();  
  
        return valueType == typeId;  
    }  
}
```

Find The Type ID

```
ConfigurablePofContext ctx = (ConfigurablePofContext)
    cache.getCacheService().getSerializer();

int typeId = ctx.getUserTypeIdentifier(GoPlayer.class);

Filter goFilter = new AndFilter(
    new SimpleTypeIdFilter(typeId),
    new EqualsFilter(new PofExtractor(Integer.class,
        GoPlayer.POF_DAN), 9));
```

Test the Type of a Property

```
public class PofTypeIdFilter implements EntryFilter {

    private int typeId;
    private int target;
    private PofNavigator navigator;

    // Constructors etc

    public boolean evaluateEntry(Entry entry) {
        BinaryEntry binEntry = (BinaryEntry) entry;
        PofContext ctx = (PofContext) binEntry.getSerializer();
        com.tangosol.util.Binary binTarget;
        switch (target) {
            case AbstractExtractor.KEY:
                binTarget = binEntry.getBinaryKey();
                break;
            case AbstractExtractor.VALUE:
                binTarget = binEntry.getBinaryValue();
                break;
            default:
                throw new IllegalArgumentException("invalid
                    target");
        }

        PofValue value = PofValueParser.parse(binTarget, ctx);

        if (navigator != null) {
            value = navigator.navigate(value);
        }

        if (value == null) {
            return false;
        }

        return typeId == value.getTypeId();
    }
}
```


Testing For Null

How much does this deserialise?

```
Set nullvaluekeys = cache.keySet(new  
    IsNullFilter("getSomething"));
```

How about this?

```
Set nullvaluekeys = cache.keySet(new EqualsFilter(  
    new PofExtractor(SOMETHING_POF)), null);
```

Testing For Null Without Deserialising

Null is represented in POF by type id -37

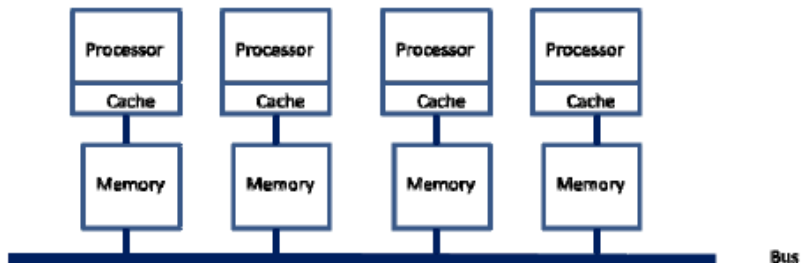
```
Filter isNullFilter = new PofTypeIdFilter(  
    -37,  
    AbstractExtractor.VALUE,  
    new SimplePofPath(GoPlayer.POF_LASTNAME));
```

With Added Syntactic Sugar

```
public class PofNullFilter extends PofTypeIdFilter {  
    public PofNullFilter(int target, PofNavigator navigator) {  
        super(-37, target, navigator);  
    }  
}
```

```
Filter isNullFilter = new PofNullFilter(  
    AbstractExtractor.VALUE,  
    new SimplePofPath(GoPlayer.POF_LASTNAME));
```

NUMA



UseNUMA

```
$ java -XX:+UseNUMA ...
```

- Objects are created in memory attached to the CPU of the thread that created them
- Parallel GC threads collect from local memory

```
$ numactl --cpubind=0 --membind=0 "$JAVA_HOME/bin/java ...".
```

- all threads of the java process will be bound to CPU 0
- memory allocations will be made from the memory connected to CPU 0
- typical throughput gains: 10-25%
- best-case measured improvement: 43%

Default Cache Configuration - coherence-cache-config.xml

```
<cache-mapping>
  <cache-name>*/</cache-name>
  <scheme-name>example-distributed</scheme-name>
</cache-mapping>

<distributed-scheme>
  <scheme-name>example-distributed</scheme-name>
  <service-name>DistributedCache</service-name>

  <backing-map-scheme>
    <local-scheme>
      <scheme-ref>example-binary-backing-map</scheme-ref>
    </local-scheme>
  </backing-map-scheme>
</distributed-scheme>

<local-scheme>
  <scheme-name>example-binary-backing-map</scheme-name>

  <eviction-policy>HYBRID</eviction-policy>
  <high-units>{back-size-limit 0}</high-units>
  <unit-calculator>BINARY</unit-calculator>
  <b>expiry-delay</b>{back-expiry 1h}</expiry-delay>
  <cachestore-scheme></cachestore-scheme>
</local-scheme>
```

On each new Coherence release...

- Download the package
- Extract the jar
- **DELETE coherence-cache-config.xml from the jar!**
- Upload to your Nexus/Artifactory/etc

Default Cache Configuration - coherence-cache-config.xml

```
<cache-mapping>
  <cache-name>*/</cache-name><cache-name>*/</cache-name>
  <scheme-name>example-distributed</scheme-name>
</cache-mapping>

<distributed-scheme>
  <scheme-name>example-distributed</scheme-name>
  <service-name>DistributedCache</service-name>

  <backing-map-scheme>
    <local-scheme>
      <scheme-ref>example-binary-backing-map</scheme-ref>
    </local-scheme>
  </backing-map-scheme>
</distributed-scheme>

<local-scheme>
  <scheme-name>example-binary-backing-map</scheme-name>

  <eviction-policy>HYBRID</eviction-policy>
  <high-units>{back-size-limit 0}</high-units>
  <unit-calculator>BINARY</unit-calculator>
  <expiry-delay>{back-expiry 1h}</expiry-delay>
  <cachestore-scheme></cachestore-scheme>
</local-scheme>
```

QUIZ!

- Three cache configurations
- for each, is there a problem?

Quiz: Configuration A

```
<distributed-scheme>
  <scheme-name>ServiceScheme1</scheme-name>
  <service-name>DistService1</service-name>
  <thread-count>10</thread-count>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>ServiceScheme2</scheme-name>
  <service-name>DistService1</service-name>
  <lease-granularity>member</lease-granularity>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>
```

Quiz: Configuration B

```
<distributed-scheme>
  <scheme-name>ServiceScheme1</scheme-name>
  <service-name>DefaultDistributedService</service-name>
  <thread-count>10</thread-count>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>ServiceScheme2</scheme-name>
  <partition-count>13</partition-count>
  <backing-map-scheme>
    <read-write-backing-map-scheme>
      <scheme-ref>ReadThroughScheme</scheme-ref>
    </read-write-backing-map-scheme>
  </backing-map-scheme>
</distributed-scheme>
```

Quiz: Configuration C

```
<distributed-scheme>
  <scheme-name>ServiceScheme1</scheme-name>
  <scheme-ref>BaseServiceScheme</scheme-ref>
  <thread-count>5</thread-count>
  <backing-map-scheme>
    <read-write-backing-map-scheme>
      <scheme-ref>ReadThroughScheme</scheme-ref>
    </read-write-backing-map-scheme>
  </backing-map-scheme>
</distributed-scheme>
```

```
<distributed-scheme>
  <scheme-name>ServiceScheme2</scheme-name>
  <scheme-ref>BaseServiceScheme</scheme-ref>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>
```

```
<distributed-scheme>
  <scheme-name>BaseServiceScheme</scheme-name>
  <service-name>DistService1</service-name>
  <thread-count>10</thread-count>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>
```

Result

	×	✓
Configuration A	×	
Configuration B	×	
Configuration C	×	

Quiz: Configuration A

```
<distributed-scheme>
  <scheme-name>ServiceScheme1</scheme-name>
  <service-name>DistService1</service-name>
  <thread-count>10</thread-count><thread-count>10</thread-count>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>
```

```
<distributed-scheme>
  <scheme-name>ServiceScheme2</scheme-name>
  <service-name>DistService1</service-name>
  <lease-granularity>member</lease-granularity><lease-granularity>member</lease-granularity>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>
```

Quiz: Configuration B

```
<distributed-scheme>
  <scheme-name>ServiceScheme1</scheme-name>
  <service-name>DefaultDistributedService</service-name>
  <thread-count>10</thread-count><del>thread-count>10</del>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>ServiceScheme2</scheme-name>
  <partition-count>13</partition-count><del>partition-count>13</del>
  <backing-map-scheme>
    <read-write-backing-map-scheme>
      <scheme-ref>ReadThroughScheme</scheme-ref>
    </read-write-backing-map-scheme>
  </backing-map-scheme>
</distributed-scheme>
```


Quiz: Configuration C

```
<distributed-scheme>
  <scheme-name>ServiceScheme1</scheme-name>
  <scheme-ref>BaseServiceScheme</scheme-ref>
  <thread-count>5</thread-count><b><thread-count>5</thread-count></b>
  <backing-map-scheme>
    <read-write-backing-map-scheme>
      <scheme-ref>ReadThroughScheme</scheme-ref>
    </read-write-backing-map-scheme>
  </backing-map-scheme>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>ServiceScheme2</scheme-name>
  <scheme-ref>BaseServiceScheme</scheme-ref>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>BaseServiceScheme</scheme-name>
  <service-name>DistService1</service-name>
  <thread-count>10</thread-count><b><thread-count>10</thread-count></b>
  <backing-map-scheme>
    <local-scheme/>
  </backing-map-scheme>
</distributed-scheme>
```

What is a caching-scheme?

- Template for configuration of caches
 - backing-map-scheme
 - listener
- Service Configuration
 - thread-count
 - backup-count
 - ... everything else

Avoiding Trouble - by Convention

- Distinguish between cache-template schemes and service schemes
- Cache template schemes
 - Have a scheme-ref but no service-name
 - Have a backing-map-scheme, and maybe a listener
 - Have *no other elements*
 - Reference (possibly indirectly) a service scheme with a service name
- Service schemes
 - Have no backing-map-scheme or listener
 - May have a service name, or be "abstract", referenced by a service scheme with a name
 - No two service schemes have the same name

Avoiding Trouble - By Convention - 1

```
<キャッシング-schemes>

  <!-- Cache template schemes - no service parameters -->

  <distributed-scheme>
    <scheme-name>SlowCacheScheme</scheme-name>
    <scheme-ref>ManyThreadServiceScheme</scheme-ref>

    <backing-map-scheme>
      <read-write-backing-map-scheme>
        <internal-cache-scheme><local-scheme/></internal-cache-scheme>
      </read-write-backing-map-scheme>
    </backing-map-scheme>

  </distributed-scheme>

  <distributed-scheme>
    <scheme-name>FastCacheScheme</scheme-name>
    <scheme-ref>FewThreadServiceScheme</scheme-ref>

    <backing-map-scheme>
      <read-write-backing-map-scheme>
        <internal-cache-scheme><local-scheme/></internal-cache-scheme>
      </read-write-backing-map-scheme>
    </backing-map-scheme>

  </distributed-scheme>
```

Avoiding Trouble - By Convention - 3

```
<!-- Service schemes - no cache template parameters -->

<distributed-scheme>
  <scheme-name>ManyThreadServiceScheme</scheme-name>
  <scheme-ref>AbstractServiceScheme</scheme-ref>
  <service-name>ManyThreadDistributedService</service-name>
  <thread-count>100</thread-count>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>FewThreadServiceScheme</scheme-name>
  <scheme-ref>AbstractServiceScheme</scheme-ref>
  <service-name>FewThreadDistributedService</service-name>
  <thread-count>5</thread-count>
</distributed-scheme>

<distributed-scheme>
  <scheme-name>AbstractServiceScheme</scheme-name>
  <autostart>true</autostart>
</distributed-scheme>

</caching-schemes>
```

Avoiding Trouble - Programmatically - 1

```
<?xml version="1.0"?>

<cache-config
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-cache-config"
  xmlns:scheme="class://org.cohbook.configuration.cache.CacheConfigNamespaceHandler"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/coherence-cache-config
    coherence-cache-config.xsd">

  <caching-scheme-mapping>
    <cache-mapping>
      <cache-name>slow-cache</cache-name>
      <scheme-name>SlowCacheScheme</scheme-name>
    </cache-mapping>
    <cache-mapping>
      <cache-name>fast-cache</cache-name>
      <scheme-name>FastCacheScheme</scheme-name>
    </cache-mapping>
  </caching-scheme-mapping>
```

Avoiding Trouble - Programmatically - 2

```
<キャッシング-schemes>

  <!-- Cache template schemes - no service parameters -->

  <distributed-scheme scheme:scheme-type="cache">
    <scheme-name>SlowCacheScheme</scheme-name>
    <scheme-ref>ManyThreadServiceScheme</scheme-ref>

    <backing-map-scheme>
      <read-write-backing-map-scheme>
        <internal-cache-scheme><local-scheme/></internal-cache-scheme>
      </read-write-backing-map-scheme>
    </backing-map-scheme>

  </distributed-scheme>

  <distributed-scheme scheme:scheme-type="cache">
    <scheme-name>FastCacheScheme</scheme-name>
    <scheme-ref>FewThreadServiceScheme</scheme-ref>

    <backing-map-scheme>
      <read-write-backing-map-scheme>
        <internal-cache-scheme><local-scheme/></internal-cache-scheme>
      </read-write-backing-map-scheme>
    </backing-map-scheme>

  </distributed-scheme>
```

Avoiding Trouble - Programmatically - 3

```
<!-- Service schemes - no cache template parameters -->

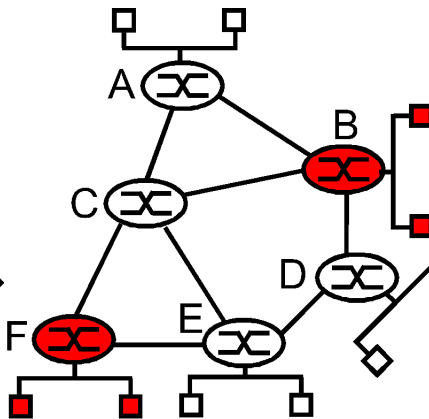
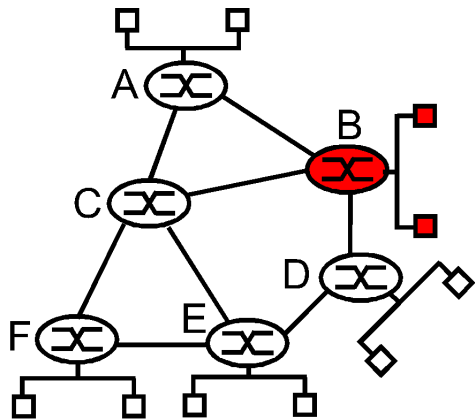
<distributed-scheme scheme:scheme-type="service">
  <scheme-name>ManyThreadServiceScheme</scheme-name>
  <scheme-ref>AbstractServiceScheme</scheme-ref>
  <service-name>ManyThreadDistributedService</service-name>
  <thread-count>100</thread-count>
</distributed-scheme>

<distributed-scheme scheme:scheme-type="service">
  <scheme-name>FewThreadServiceScheme</scheme-name>
  <scheme-ref>AbstractServiceScheme</scheme-ref>
  <service-name>FewThreadDistributedService</service-name>
  <thread-count>5</thread-count>
</distributed-scheme>

<distributed-scheme
  scheme:scheme-type="abstract-service">
  <scheme-name>AbstractServiceScheme</scheme-name>
  <autostart>true</autostart>
</distributed-scheme>

</caching-schemes>
```


Multicast Routing 1



Multicast Routing 2

- multicast packets only travel on routes where they are needed
- shared multicast addresses, even with different ports:
 - increase traffic on the network
 - increase load on the hosts (as they receive and discard packets not for their ports)
- routers discard multicast packets even if they have a high TTL
- some routers are extremely inefficient at discarding packets with TTL=zero

Multicast Configuration Rules

- 1 Use a separate multicast address for each set of cluster hosts
- 2 Set a high TTL; let the routers manage discards
- 3 Turn *off* multicast for local testing on PCs, build hosts etc

Finally...

david.whitmarsh@sixwhits.com

<http://www.coherencecookbook.org/downloads>