Measuring performance and capacity planning in Java-based data grids

Mircea Markus
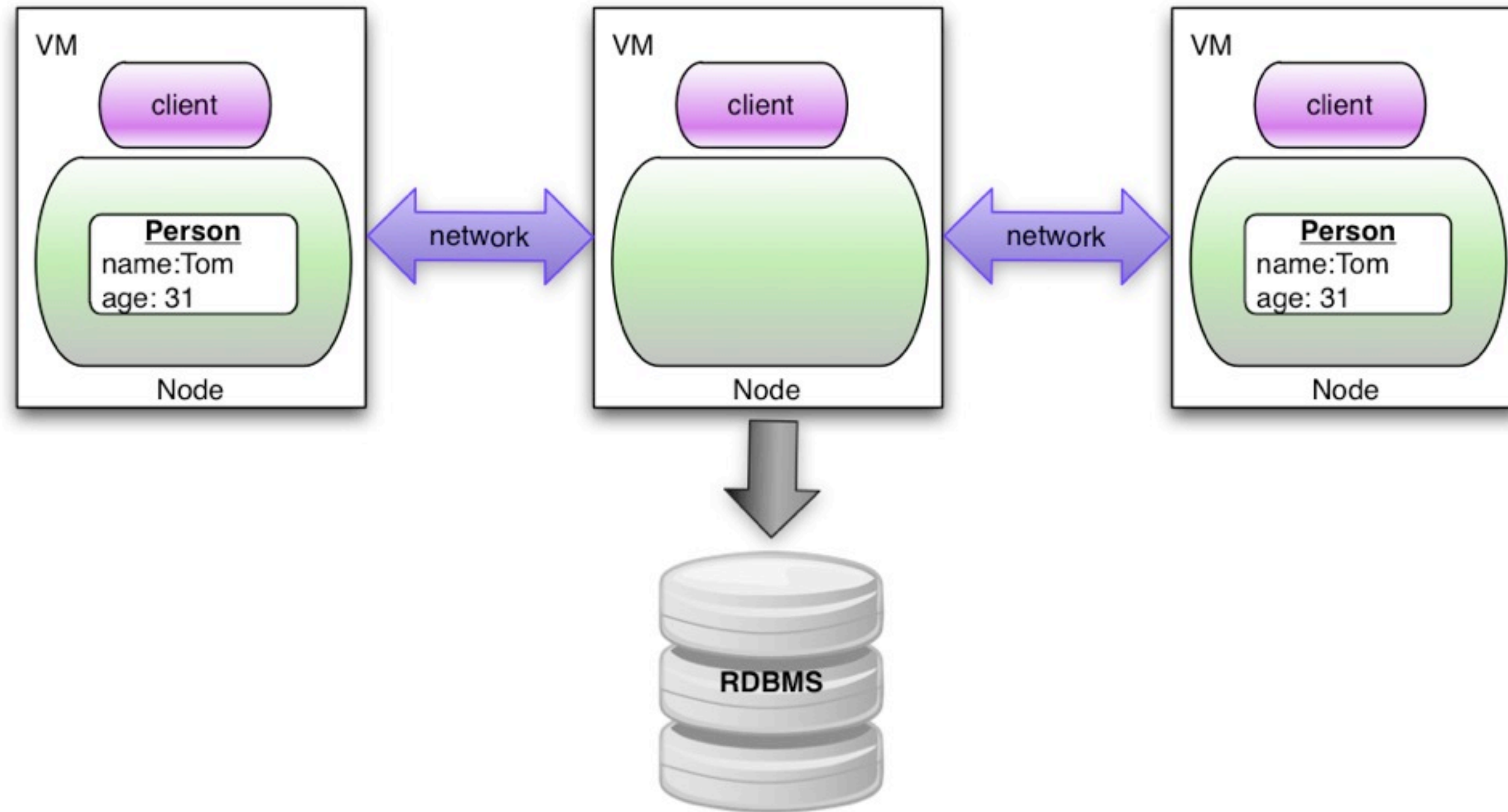Principal Software Engineer
Red Hat

Who's this guy?

- R&D Engineer, Red Hat Inc.
- AS clustering: JBossCache, JGroups
- Infinispan, Radargun
- Twitter: @mirceamarkus
- Blog: mirceamarkus.blogspot.com

JBoss Community

Agenda

- In-memory data grids
    - characteristics
- Project Radargun
    - why benchmarking?
    - features

In memory data grids

Data grids (continued)

- No standard yet (JSR-347)
  - products share similar characteristics
- API
  - Map based
  - no/less strict schema
- Build with horizontal scaling in mind
  - consistent hashing commonly used
  - available during topology changes
  - elastic
  - transactional

JBoss Community

Access patterns

- Embedded
  - client and node in same VM
  - fast communication
  - less object serialisation
  - supports transactions
- Remote
  - similar to DB
  - client/server
  - transactional or not

Available on the market
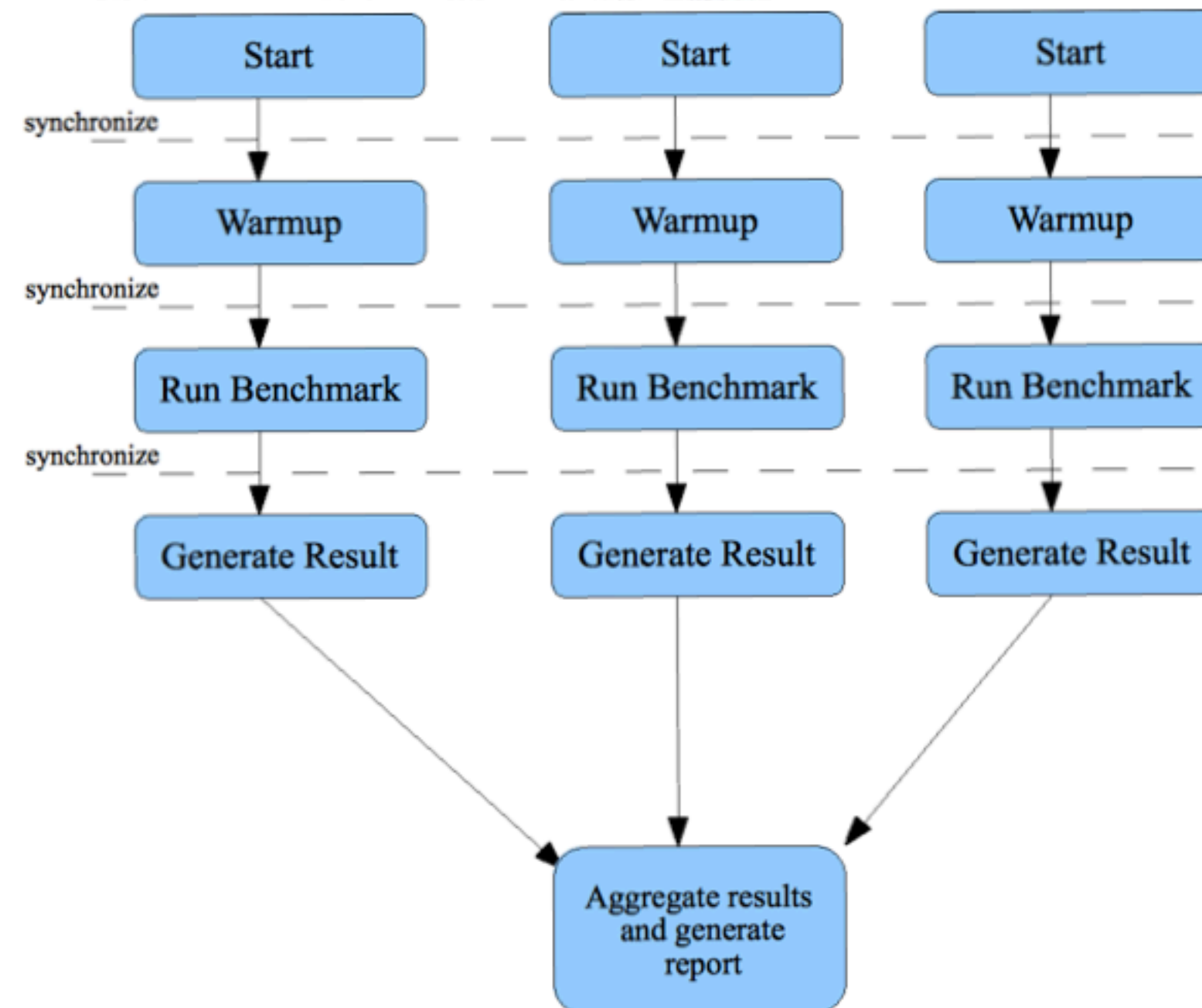
Why Radargun?

- Performance regression
  - an easy way to start a datagrid
- Compare with competitors
- Prototyping
  - easy to write & run

How does it work?

Configuration

```xml
<bench-config>
    <master bindAddress="${127.0.0.1:master.address}" port="${2103:master.port}"/>

    <benchmark initSize="2" maxSize="${4:slaves}" increment="1">
        <DestroyWrapper runOnAllSlaves="true"/>
        <StartCluster staggerSlaveStartup="true"/>
        <ClusterValidation partialReplication="false"/>
        <WebSessionWarmup numberOfRequests="2500" opsCountStatusLog="500" numOfThreads="2"/>
        <ClearCluster/>
        <WebSessionBenchmark numberOfRequests="2500" opsCountStatusLog="500" numOfThreads="2"/>
        <CsvReportGeneration/>
    </benchmark>

    <products>
        <jbosscache3> <config name="mvcc/mvcc-repl-sync.xml"/> </jbosscache3>
        <infinispan4> <config name="repl-sync.xml"/> </infinispan4>
    </products>

    <reports>
        <report name="All" includeAll="true"/>
    </reports>

</bench-config>
```
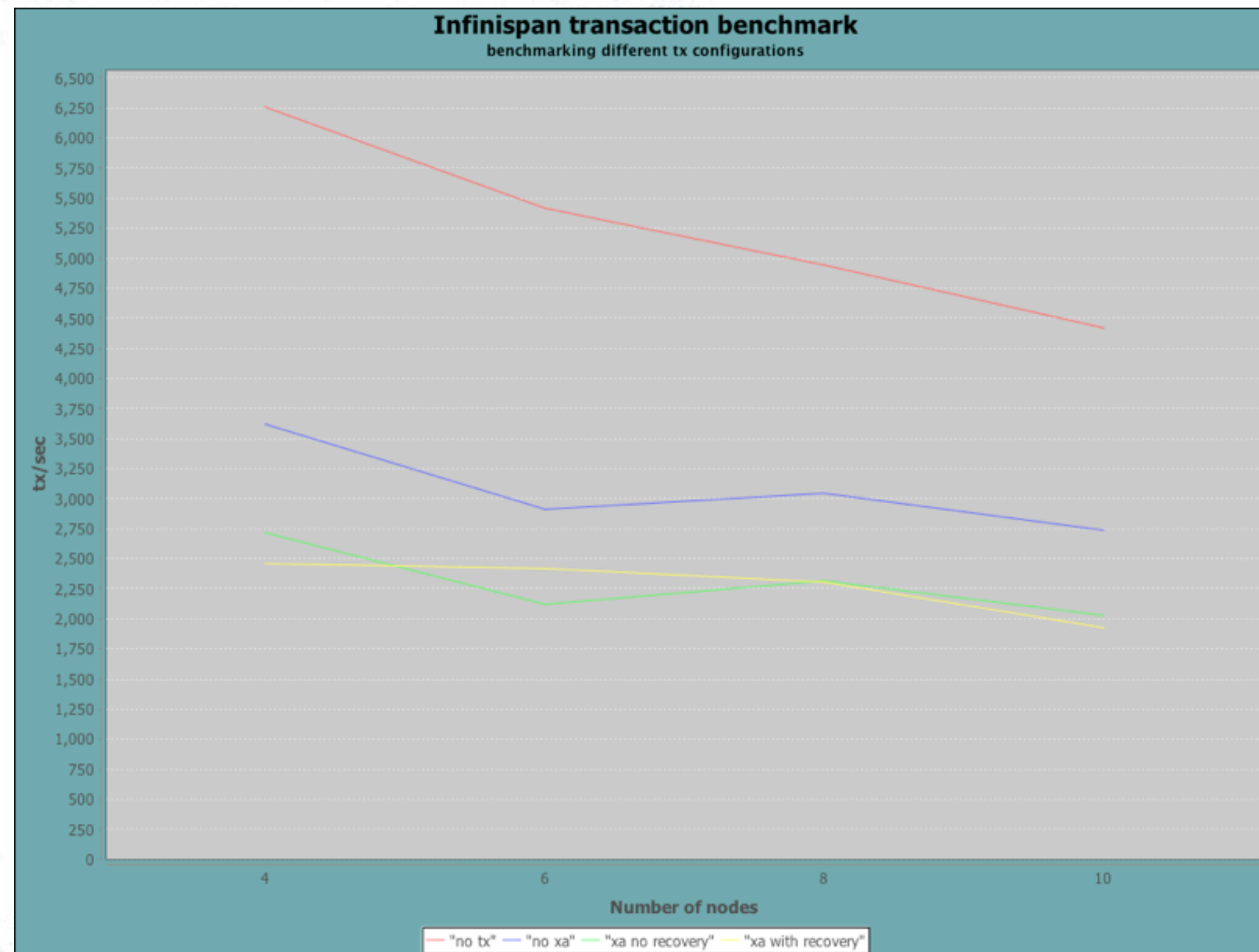
JBoss Community

Scripts

- benchmark.sh
  - runs a whole benchmark
- chart.sh
- dist.sh
  - verify distribution evenness
- local.sh
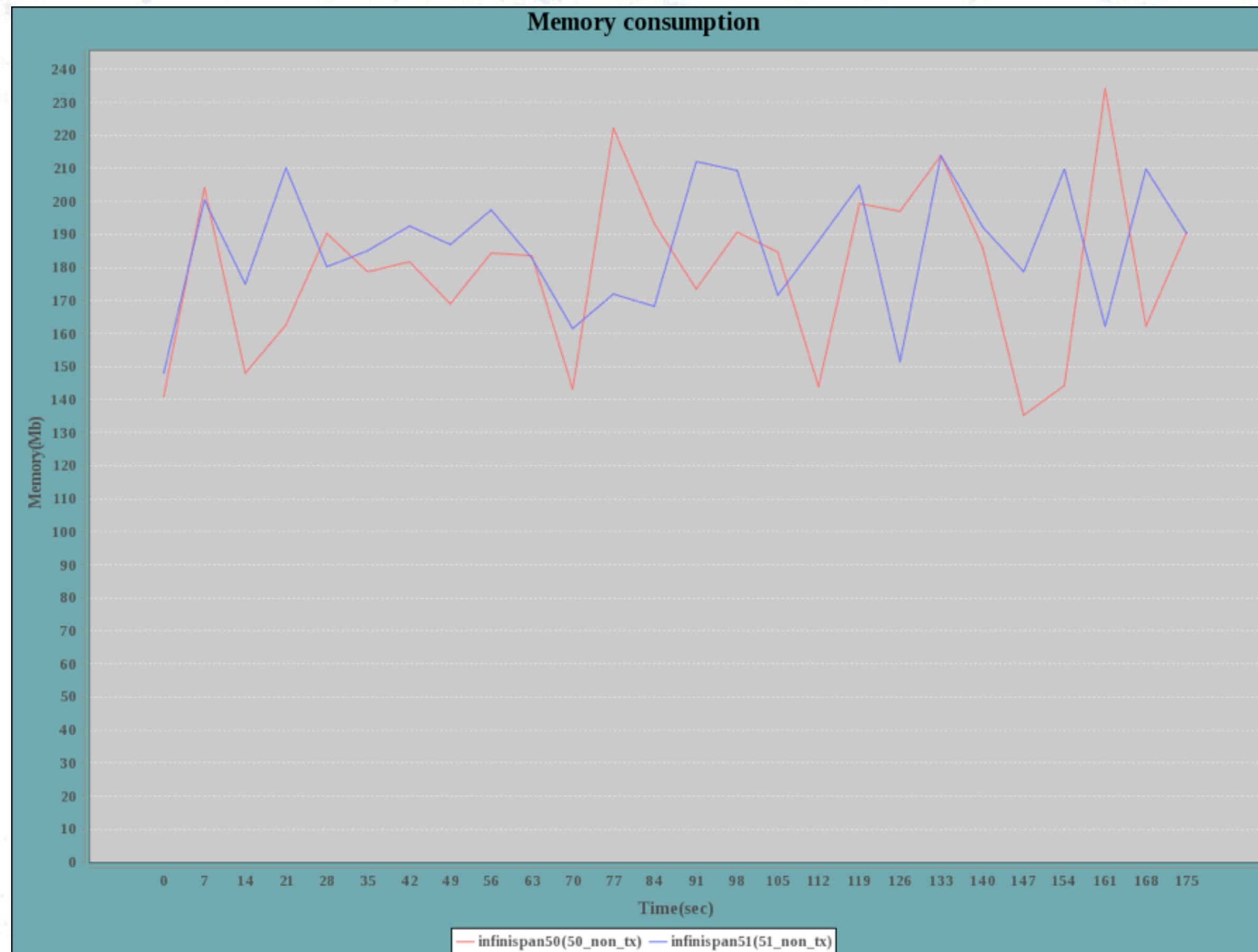 - same as benchmark.sh but for local caches (JSR107)

Benchmarks

- Web Session Benchmark
  - no contention
  - configurable: nrOfRequest, writePercentage, sizeOfAttribute..
- TPC-C
  - adaption of the www.tpc.org
  - designed for contention
- Pluggable
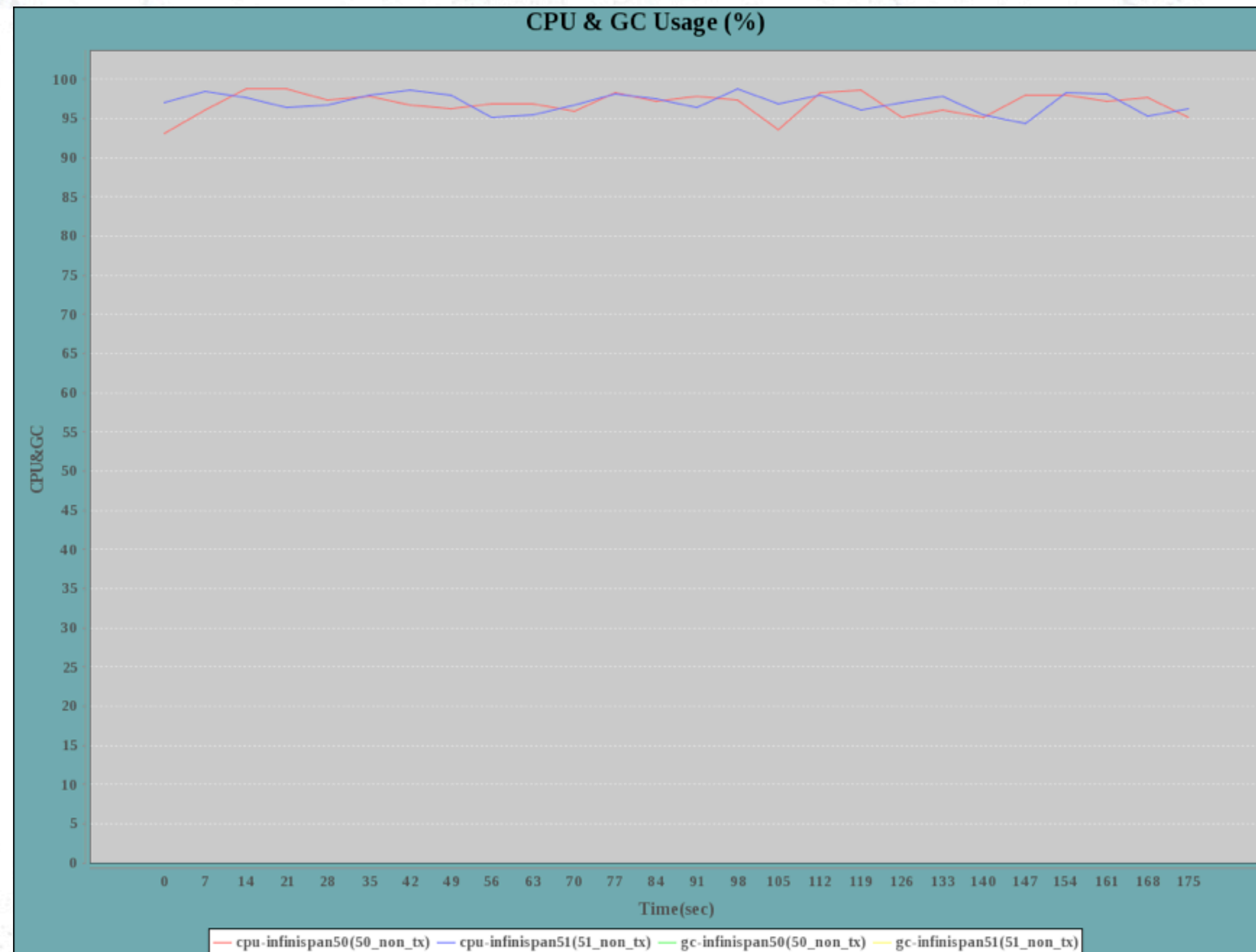 - add your own! (and contribute back)

Results

More results - memory

More results - CPU

Plugins

- Infinispan
- JBossCache
- Oracle Coherence
- EHCache
- Teracotta
- JGroups
- Custom
  - easy to write..

Custom plugin

```java
public interface CacheWrapper {
    void setUp(String config, boolean isLocal, int nodeIndex, TypedProperties confAttributes) throws Exception;

    void tearDown() throws Exception;

    void put(String bucket, Object key, Object value) throws Exception;

    Object get(String bucket, Object key) throws Exception;

    void empty() throws Exception;

}
```

JBoss **Community**

Want to know more?

- http://radargun.sf.net
- mirceamarkus.blogspot.co.uk
- @mirceamarkus

JBoss Community