Benchmarking Infinispan

Manik Surtani & Mircea Markus

twitter.com/maniksurtani
twitter.com/mirceamarkus
Who we are?

A pair of core Infinispan engineers.

Who are you?
Agenda

• Infinispan – 30,000 ft overview
• The need for benchmarking
• Challenges in benchmarking data grids
• The Cache Benchmark Framework
• Live demo!
What is Infinispan?

- Open source, in-memory data grid
- Written for the JVM
- Embedded (P2P) as well as client/server mode
- Client/server endpoints: Hot Rod, Memcached, REST
  - Consumable by non-JVM systems
- Successor to JBoss Cache
- See other talks during the week
Why benchmarking?

“Benchmarks are like assholes. Everyone has one.” – Turk, Mladen
Why benchmarking?

- “Measure – don’t guess”
- Monitor performance
- against JBoss Cache, and other competing products
- between Infinispan releases
- different configurations, access patterns
- exceptional conditions
- while scaling out
- Cornerstone for performance improvements
Challenges

• Multiple access patterns
• Web session replication
• Fronting a database
• Durable data store
• (your custom access pattern here)
Challenges

- Multiple cluster modes
  - Replicated
  - Distributed
  - Invalidated
- Network stacks
  - TCP
  - UDP
  - Both!
Challenges

- Multiple products/versions/configs
- Infinispan
- JBoss Cache
- (your favourite data grid here)
- MxN problem
- If not comparative, what’s the point?
Challenges

• Benchmarking scalability

```java
do {
    runBenchmark()
    clusterSize ++
} while (clusterSize < maxNodes)
```

• Measure throughput and TPS
Challenges

- Complex setup
- Coordinate between multiple distributed benchmark processes
- Result collation
  - Centralised reporting
- Deal with anomalies and node failure
  - Including transient network hiccups
The tools we had
Cache Benchmark Framework

- OSS, hosted on SourceForge
- Inspect the code, no smoke and mirrors!
- Java-based
- Vendor-neutral. Contributors include:
  - Greg Luck/EHCache
  - Talip Ozturk/Hazelcast
Cache Benchmark Framework

- Currently used for benchmarking Infinispan
- Can run comparative benchmarks
  - Against different products
  - Against same product, for regressions
  - Against different configurations
- Ships with simple access patterns
- Benchmark for scalability
Cache Benchmark Framework

- Extensible
- Add your own data grid products to test against
- Add your own data access patterns
  - JUnit-like test cases
- Access patterns can be combined with products
  - No more M x N
- And tested as you scale out!
Cache Benchmark Framework

- Script-based
- Currently only for POSIX
  - BASH scripts for launching remote processes
  - No Windows support yet
- Only for bootstrap
Cache Benchmark Framework

- Each process a finite state machine
- Master node to coordinate state change
- And to collate results ...
- ... and generate reports
  - Including graphical ones!
Basic idea

- Start
  - Warmup
    - Run Benchmark
      - Generate Result
  - Synchronize

- Start
  - Warmup
    - Run Benchmark
      - Generate Result
  - Synchronize

- Start
  - Warmup
    - Run Benchmark
      - Generate Result

Aggregate results and generate report
Status

• 1.0 GA released
• Web session replication access pattern
• Plugins for JBoss Cache, Infinispan, Coherence, EHCache
• Heavily used for tuning Infinispan’s performance
• Currently used on a 128 node cluster
Road map

• Additional measurements
• Memory & CPU overhead over time/scale-out
• New data access patterns
• Control via JMX
• Standardise!
• SPEC?
Want to see it?

- Web session replication
- Benchmark Infinispan vs. JBoss Cache
- Replication for both
- Distribution for Infinispan
Q&A

- http://cachebenchfwk.sourceforge.net
- http://www.infinispan.org
- http://blog.infinispan.org
- http://twitter.com/infinispan
- #infinispan
Learn more about Infinispan!

- Storing Data on Cloud Infrastructure in a Scalable, Durable Manner - Wed 23rd
- Using Infinispan for High Availability, Load Balancing, & Extreme Performance - Thu, 24th
- How to Stop Worrying & Start Caching in Java - Thu 24th
- Why RESTful Design for Cloud is Best - Fri 25th