

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

How to Stop Worrying and Start Caching in Java

SriSatish Ambati , Azul Systems
Manik Surtani, RedHat Inc

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



The Trail

Examples

Elements of Cache Performance

Theory

Metrics

200GB Cache Design

Whodunit

Overheads in Java

Serialization

Sockets/Acceptor Threads

Queues (SEDA)

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Some example caches

Homegrown caches – Surprising work well.

(do it yourself! It's a giant hash, all said & done)

JBoss Cache, Coherence, Gemstone, GigaSpaces,
TerraCotta, EhCache, JCache JSR107.

NoSQL stores (Apache Cassandra)

Non java alternatives: MemCached & clones.

Wie die wahl hat, hat die Qual!

He who has choices has hell

SUMMIT

**JBoss
WORLD**

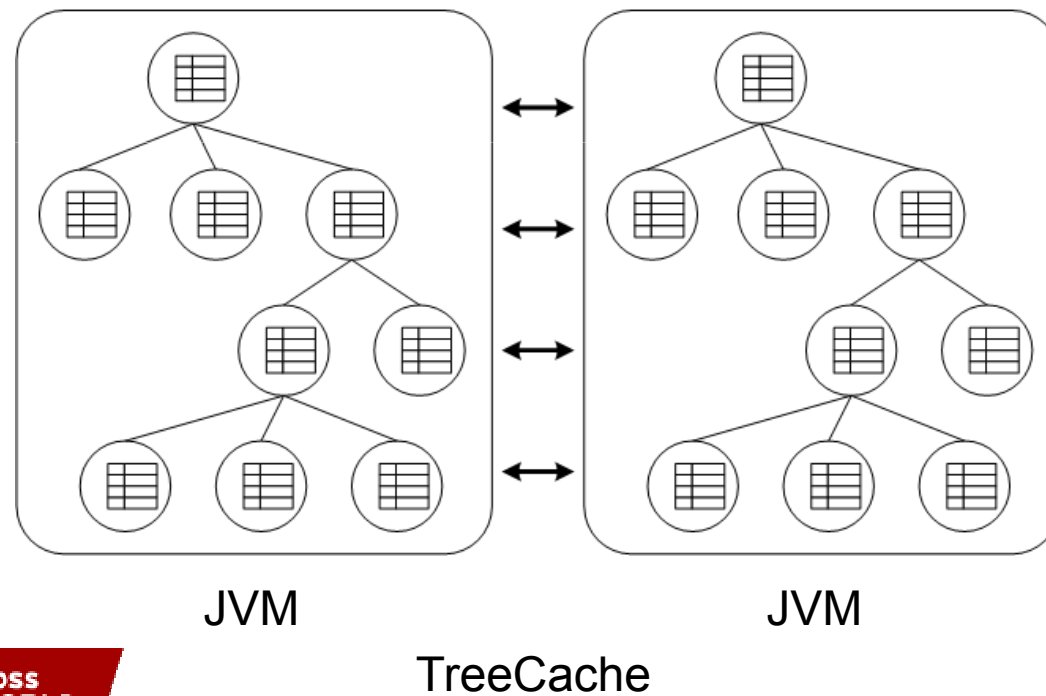
PRESENTED BY RED HAT



Visualize Cache

Simple example - JBoss Cache

Visualize Cache



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Elements of Cache Performance

Hot or Not: The 80/20 rule.

A small set of objects are very popular!

Hit or Miss: Hit Ratio

How effective is your cache?

LRU, LFU, FIFO.. Expiration.

Long-lived objects, better locality

Spikes happen

Cascading events: Node load, Node(s) dead.

Cache Thrash: Full Table scan.

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Elements of Cache Performance : Metrics

Inserts: Puts/sec

Reads: Gets/sec, Indexing

Updates: mods/sec (Locate, Modify & Notify)

Replication, Consistency, Persistence

Size of objects

Size of Cache

of CacheServer Nodes

of clients (readonly, read write)

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Partitioning

Near cache

Put Data close to Logic

Read-only nodes

Read-Write nodes

Communication Costs

Balancing (buckets)

Serialization Costs

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



I/O considerations

Asynchronous

Sockets

Queues & Threads serving the sockets

Bandwidth

Persistence

File

DB

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Buckets– Number of Partitions, Hashing function

Birthdays, Hashmaps & Prime Numbers

Collisions, Chaining

Unbalanced Hash behaves like a list

$O(n)$ retrieval

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



How many nodes to get a 200G cache?

Who needs a 200G cache?

Disk is the new Tape!

200 nodes @ 1GB heap each

2 nodes @ 100GB heap each

(plus overhead)

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



SIDE ONE

Join together with the band

I don't even know myself

SIDE TWO

Let's see action

Relay

Don't happen that way at all

The seeker



SUMMIT

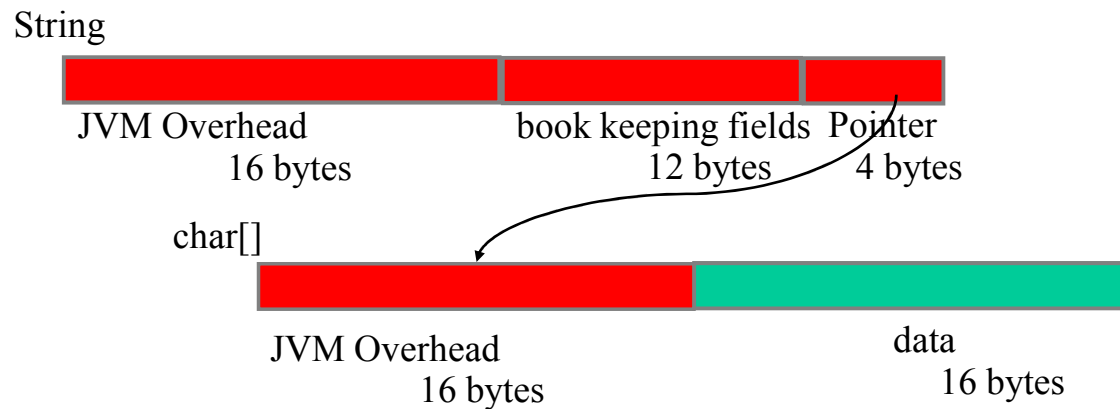
**JBoss
WORLD**

PRESENTED BY RED HAT



Java Limits: Object overhead

How many bytes for a 8 char String ?



How many objects in a Tomcat idle instance?

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JEE is not cheap either!

alloc live
 (million objects)
JBoss 5.1 ~20 ~4
Apache Tomcat 6.0 ~0.25 ~0.1

JBoss 5.1
Allocated

Class name	Size (B)	Count	Avg (B)
Total	1,410,764,512	19,830,135	71.1
char[]	423,372,528	4,770,424	88.7
byte[]	347,332,152	1,971,692	176.2
int[]	85,509,280	1,380,642	61.9
java.lang.String	73,623,024	3,067,626	24
java.lang.Object[]	64,788,840	565,693	114.5
java.util.regex.Matcher	51,448,320	643,104	80
java.lang.reflect.Method	43,374,528	301,212	144
java.util.HashMap\$Entry[]	27,876,848	140,898	197.9
java.util.TreeMap\$Entry	22,116,136	394,931	56
java.util.HashMap\$Entry	19,806,440	495,161	40
java.nio.HeapByteBuffer	17,582,928	366,311	48
java.nio.HeapCharBuffer	17,575,296	366,152	48
java.lang.StringBuilder	15,322,128	638,422	24
java.util.TreeMap\$EntryIterator	15,056,784	313,683	48
java.util.ArrayList	11,577,480	289,437	40
java.util.HashMap	7,829,056	122,329	64
java.util.TreeMap	7,754,688	107,704	72

Apache Tomcat 6.0
Allocated

Class name	Size (B)	Count	Avg (B)
Total	21,580,592	228,805	94.3
char[]	4,215,784	48,574	86.8
byte[]	3,683,984	5,024	733.3
Built-in VM methodKlass	2,493,064	16,355	152.4
Built-in VM constMethodKlass	1,955,696	16,355	119.6
Built-in VM constantPoolKlass	1,437,240	1,284	1,119.30
Built-in VM instanceKlass	1,078,664	1,284	840.1
java.lang.Class[]	922,808	45,354	20.3
Built-in VM constantPoolCache	903,360	1,132	798
java.lang.String	753,936	31,414	24
java.lang.Object[]	702,264	8,118	86.5
java.lang.reflect.Method	310,752	2,158	144
short[]	261,112	3,507	74.5
java.lang.Class	255,904	1,454	176
int[][]	184,680	2,032	90.9
java.lang.String[]	173,176	1,746	99.2
java.util.zip.ZipEntry	172,080	2,390	72

Live

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Java Limits: Garbage Collection

Constraints for Cache Configuration

Pause Times/Size

If Stop_The_World_Pause > Time_To_Live

→ node is declared dead

Allocation Rate: Per node Writes

Size of Heap, Live Heap (residency)

Residency, r

if residency > 50%. GC overheads dominate.

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Locks: Why Amdahl's law trumps Moore's!

Schemes

Optimistic, Pessimistic

Consistency

Eventually vs. ACID

Contention, Wait on Objects

`java.util.concurrent`, critical sections

Transactions are expensive

→ Stop JTA abuse

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Inter-node communication

TCP for mgmt & data

TCP for mgmt, UDP for data

- Coherence

UDP for mgmt, TCP for data

- Cassandra

Instrumentation

- Terracotta

Bandwidth considerations

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Sockets, Queues, Threads, Stages

How many sockets?

Gemstone (Multi-socket implementation)

of nodes, clients

How many threads per socket?

Asynchronous IO/Events

Curious Case of a single threaded Queue Manager

→ Context switching

→ SEDA

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Marshal Arts: Ser/Deserialization

java.io.Serializable is S..L....O.....W

- + Use “transient”
- + Roll your own, jserial, avro, etc
- + Google Protocol Buffers, PortableObjectFormat
 - Coherence
- + Externalizable + byte[]

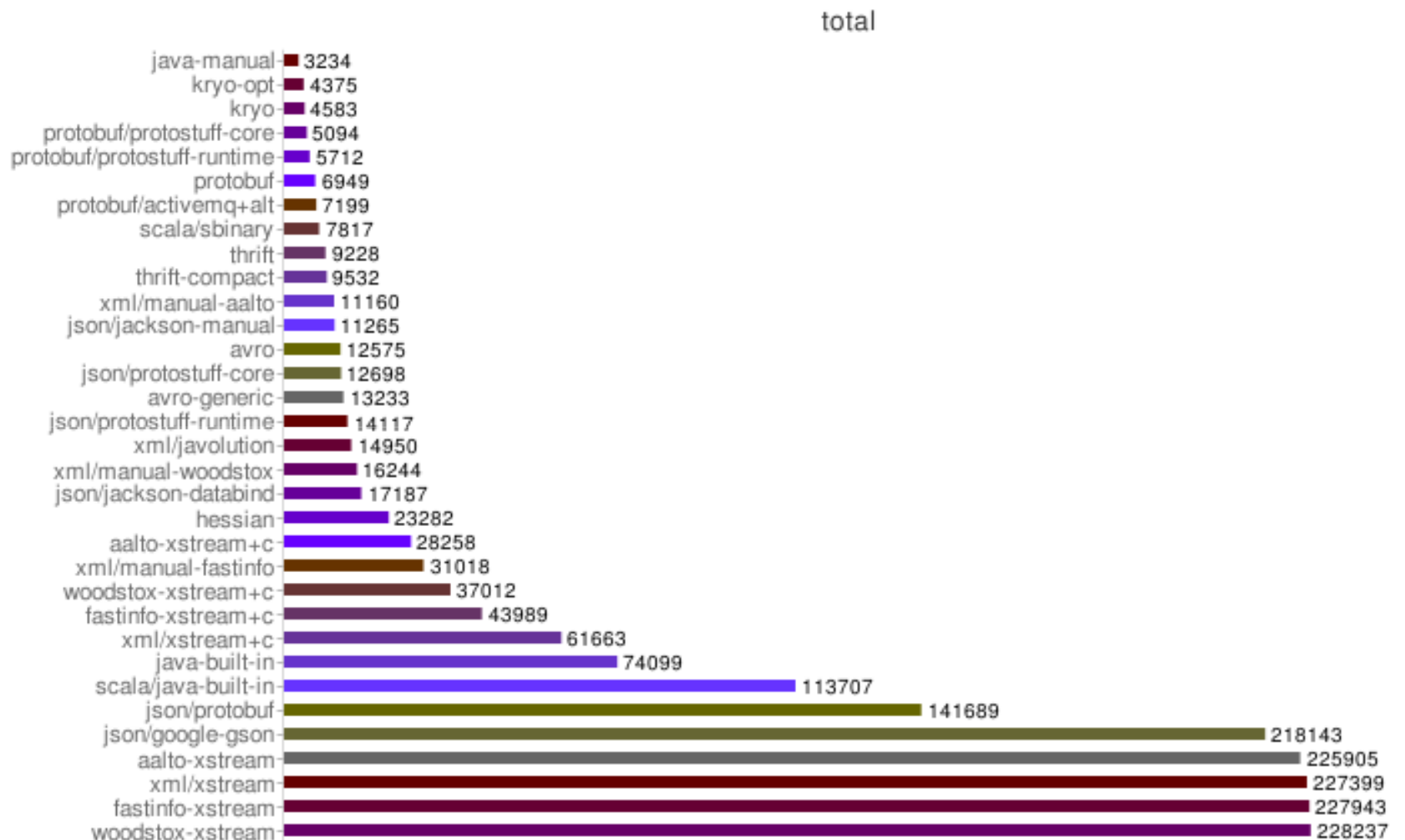
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Serialization + Deserialization uBench



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Q&A

References:

Making Sense of Large Heaps, Nick Mitchell, IBM

Oracle Coherence 3.5, Aleksandar Seovic

JVM Serialization Benchmarks, <http://code.google.com/p/thrift-protobuf-compare/wiki/BenchmarkingV2>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

[#summitjbw](https://twitter.com/summitjbw)

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

