4 June 2012

Sanne Grinovero, Red Hat

What you get by replicating Lucene indexes on the Infinispan Data Grid

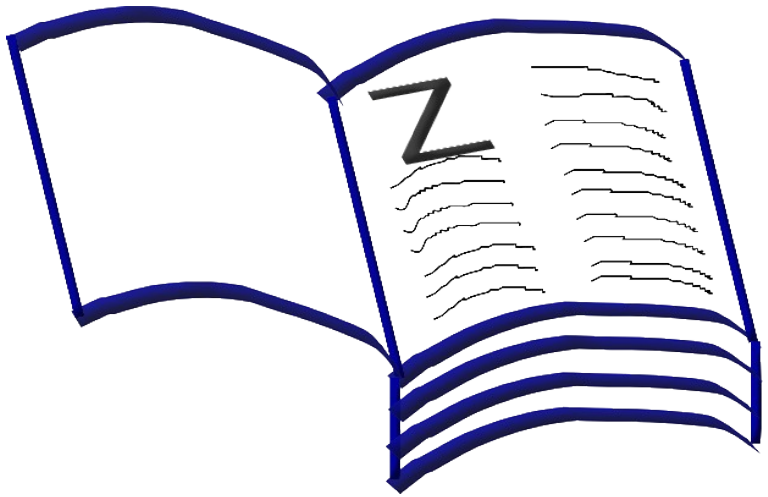
Who is that guy?

- Sanne Grinovero
  - From this planet
- Team Hibernate
  - Hibernate Search
  - Hibernate OGM
- Team Infinispan
  - Infinispan Core
  - Infinispan Query
- Apache Lucene, Netty, HotSpot, ANTLR, JGroups, Byteman, The Jokre

What are we talking about?

- Apache Lucene
- Infinispan
  - Integrations with Lucene
    - Infinispan Lucene Directory

Apache Lucene ?



Infinispan

- An in-memory datagrid
  - Memory of multiple *nodes*
  - Cluster modes
  - CacheLoaders
  - Integrations with Lucene
    - Lucene Directory

Infinispan API?

- Map-like key/value store
  - JSR 107 javax.cache.Cache interface
  - JSR 347 ??
- Asynchronous API

In practice:

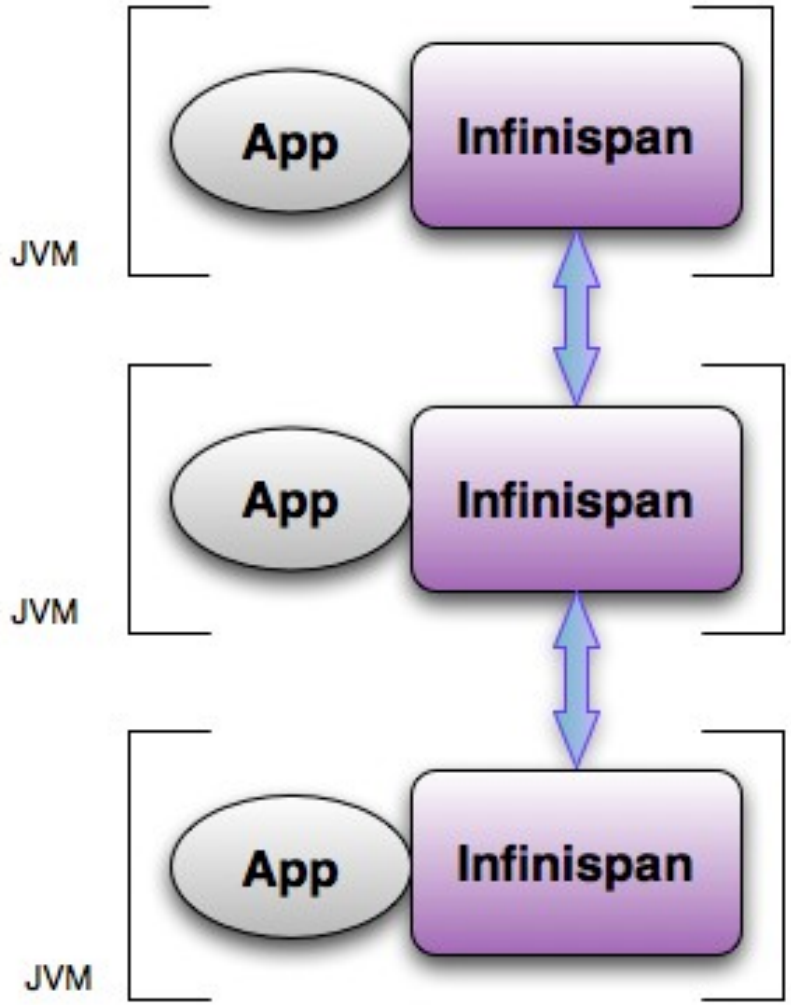
```
cache.put( “user-34”, userInstance );

cache.get( “user-34” );

cache.remove( “user-34” );

cache.putIfAbsent( “user-38”, other );
```

Distributed Data



Connected via JGroups

A Toolkit for Reliable Multicast Communication

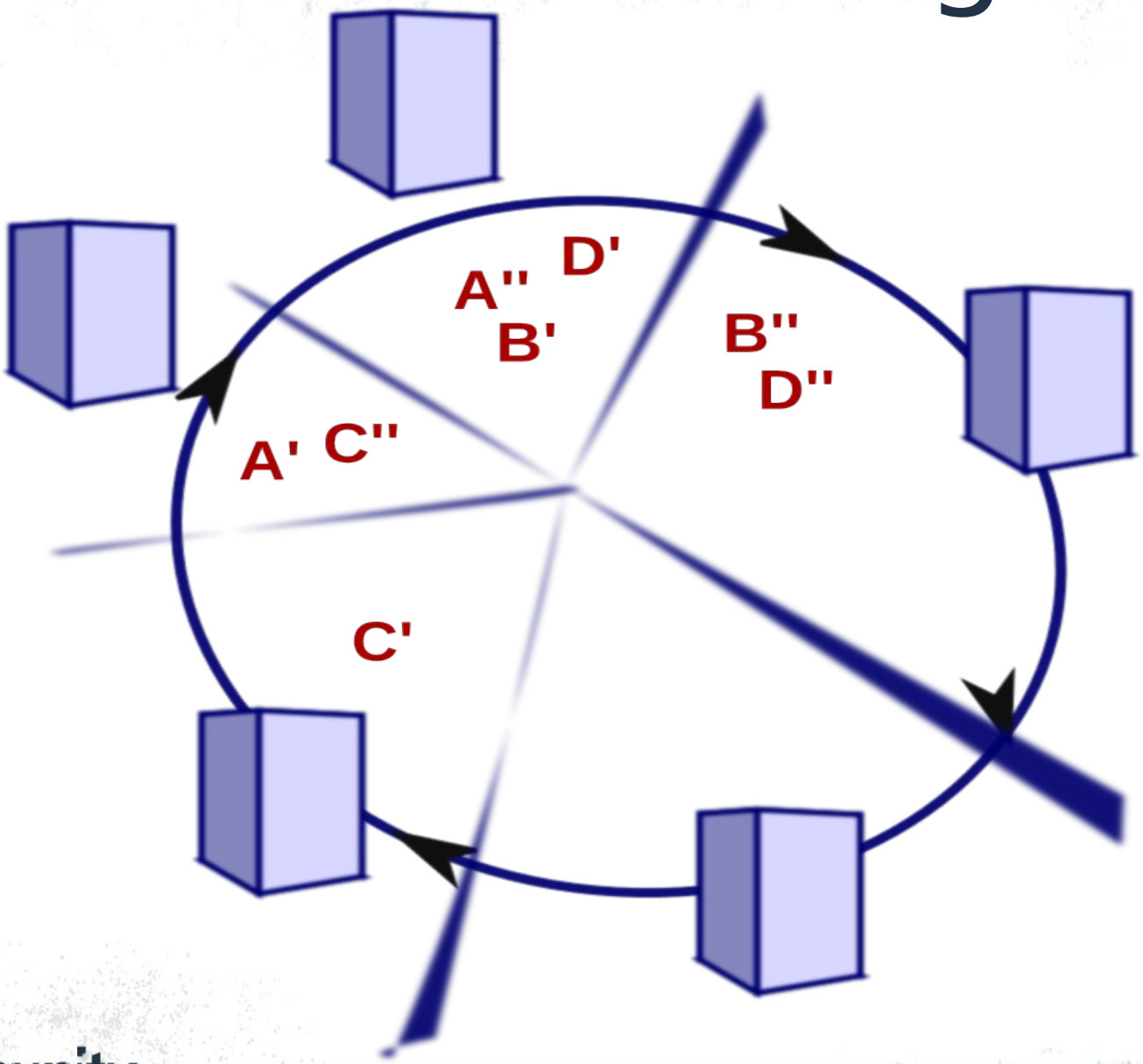
http://jgroups.org

JBoss Community

Or remote clients via:

- Memcached
- REST
- Hot Rod (Ruby, Python, *C, C#, ...*)
  - Netty

Consistent Hashing: DIST



Transactions!

JBoss AS7 core component

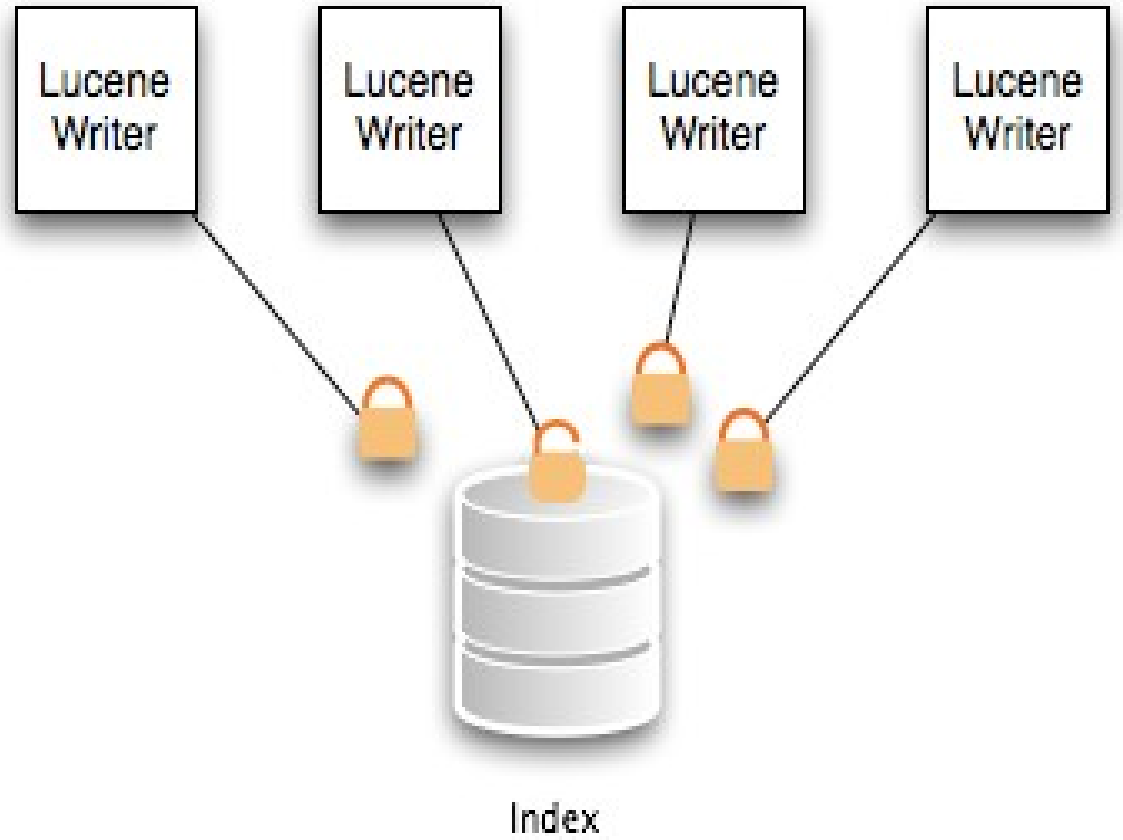
- Cluster nodes autodiscovery
- Session replication / failover
- Hibernate second level cache
- mod_cluster integration

In-memory volatile?

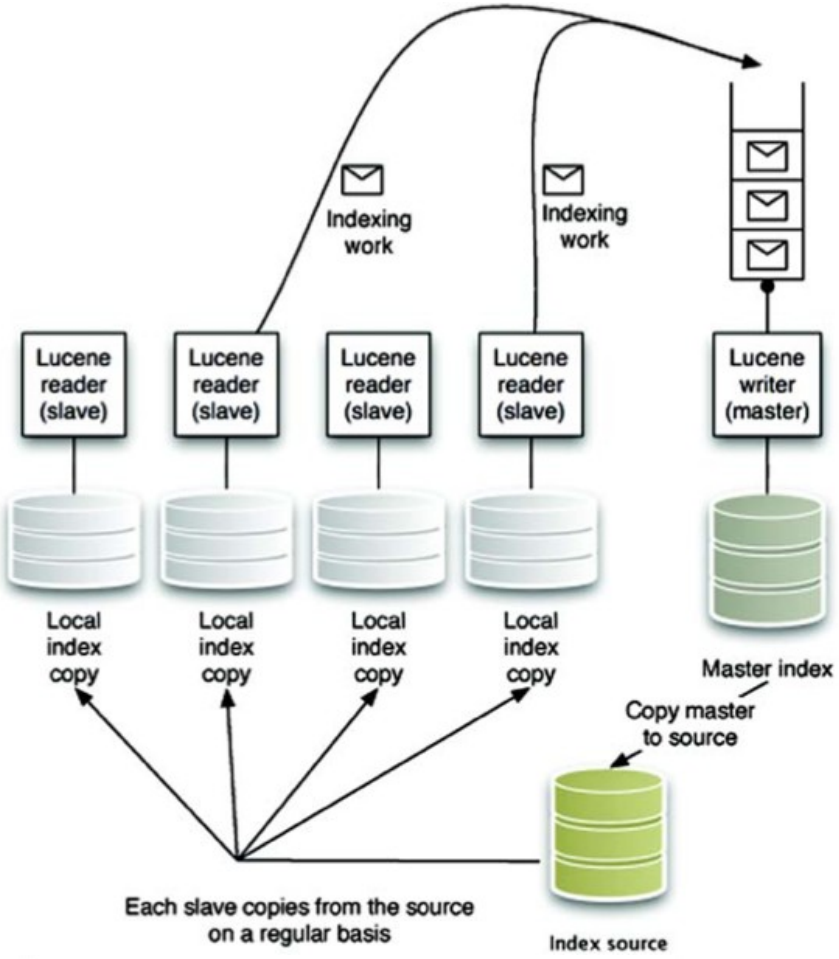
Cache Stores: durability, warm caches, more capacity...

- Cassandra
- HBase
- JDBC
- Clouds (S3, ...)
- Plain Old Files
- Many more + custom

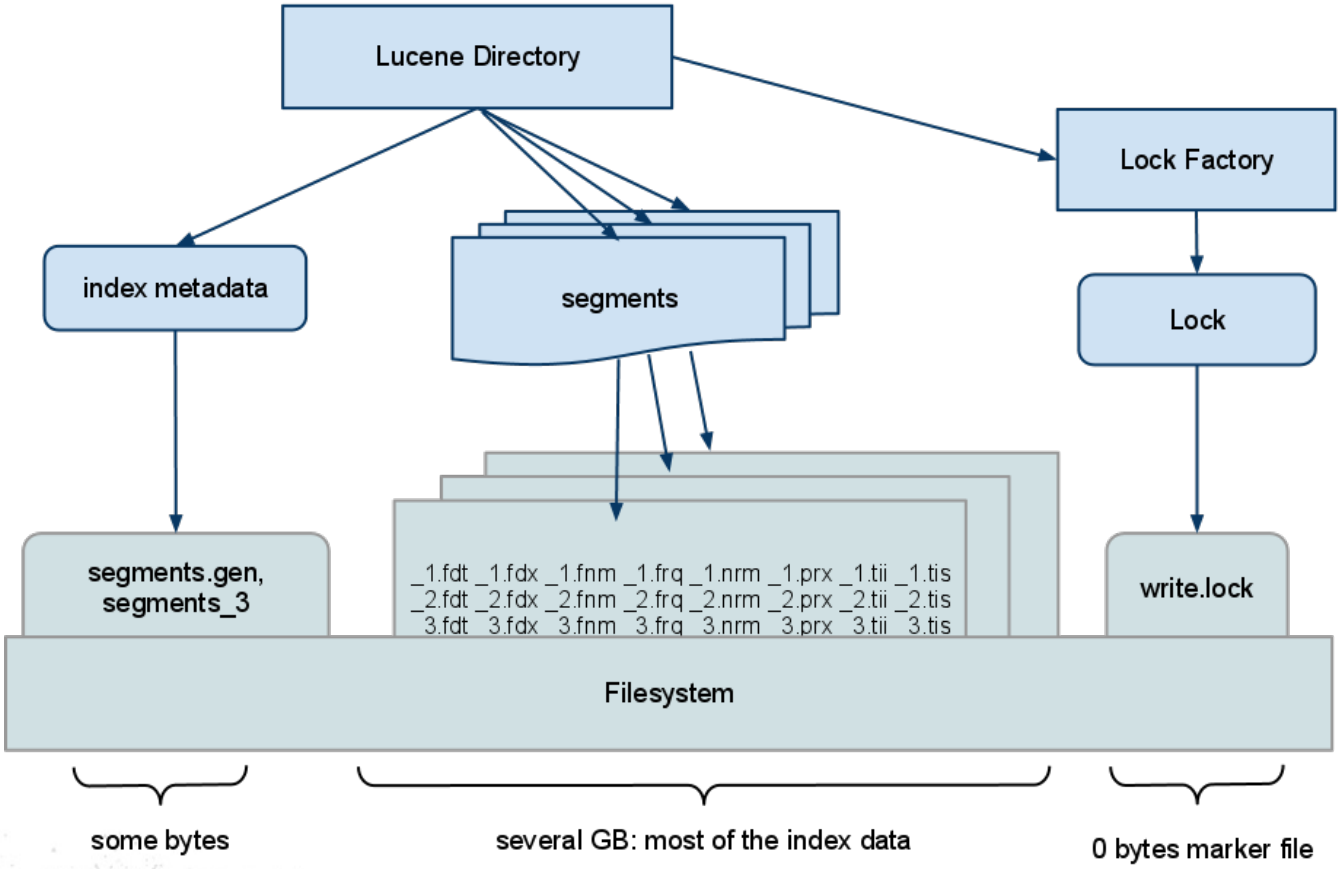
Back on Lucene: Single Writer lock

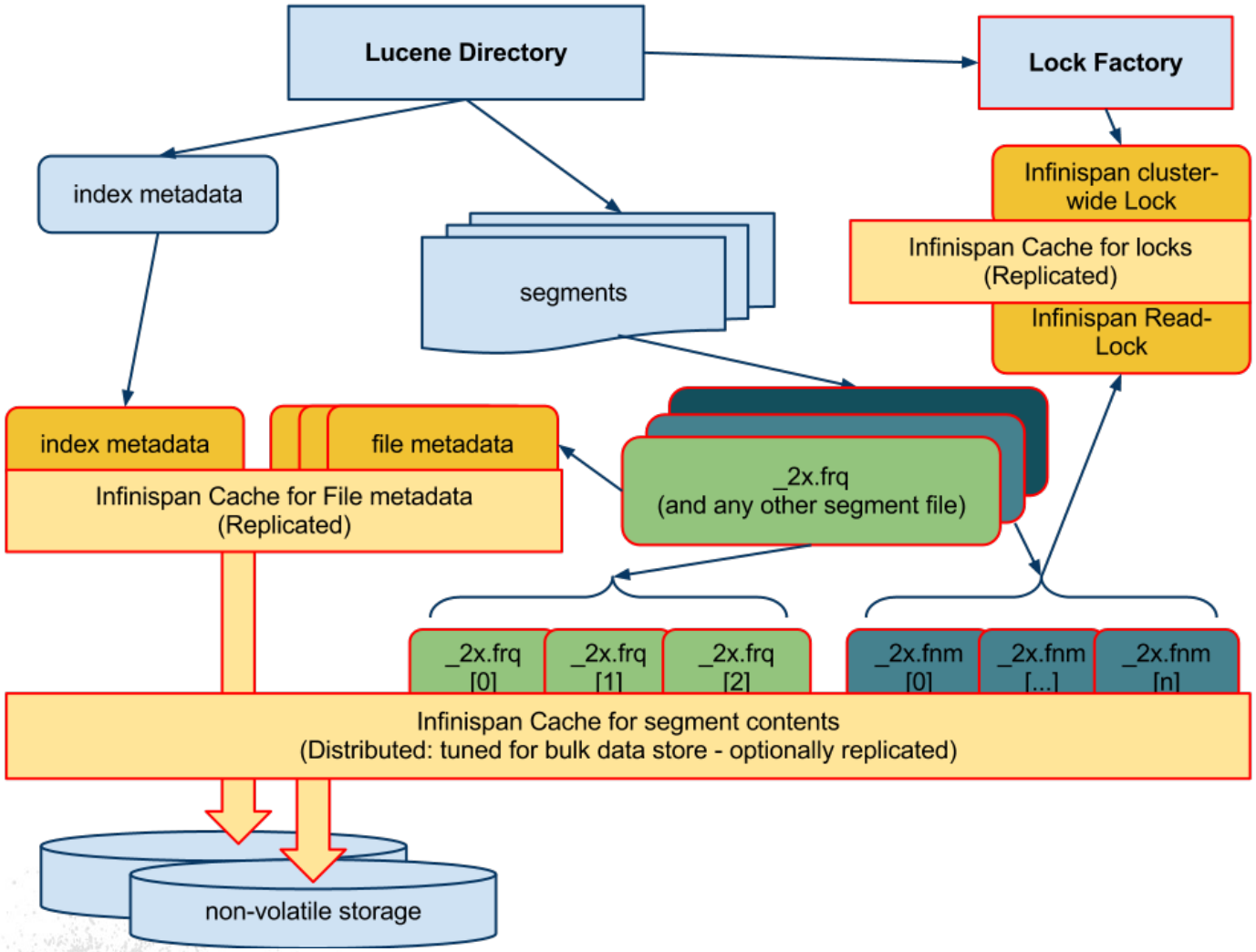


Queue-based clustering
(filesystem index)

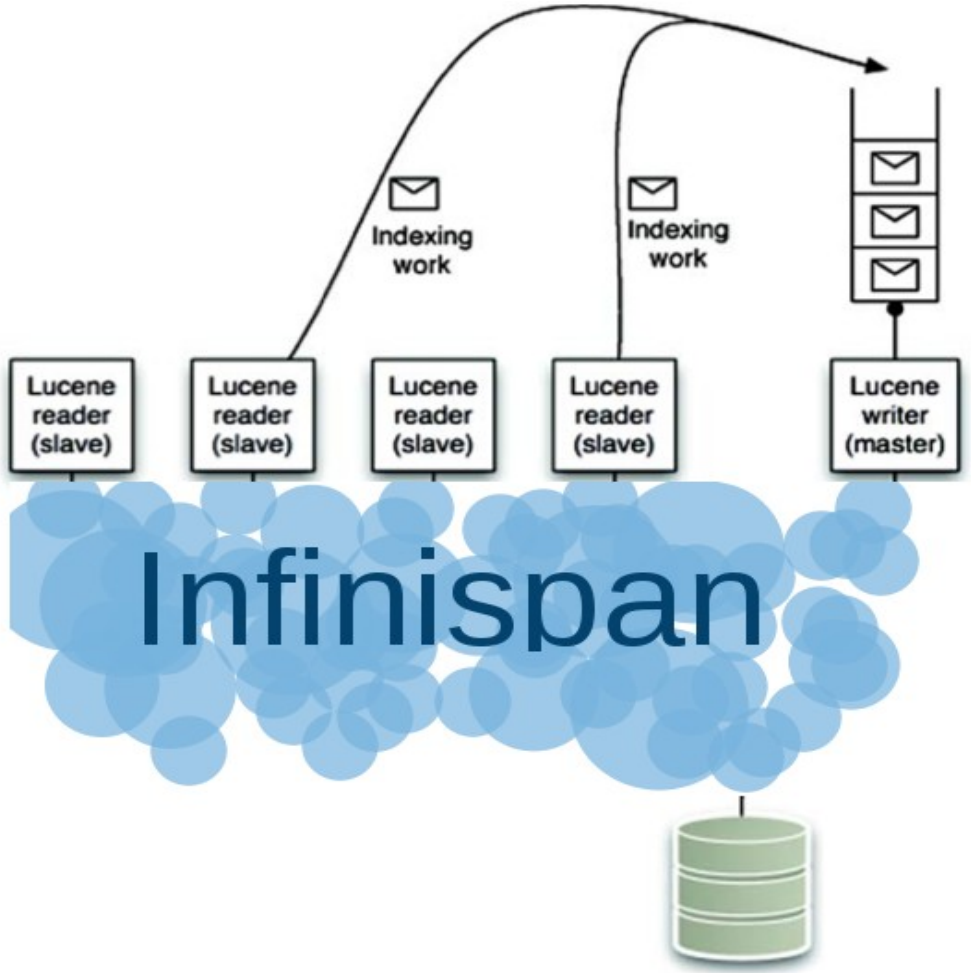


Lucene index storage

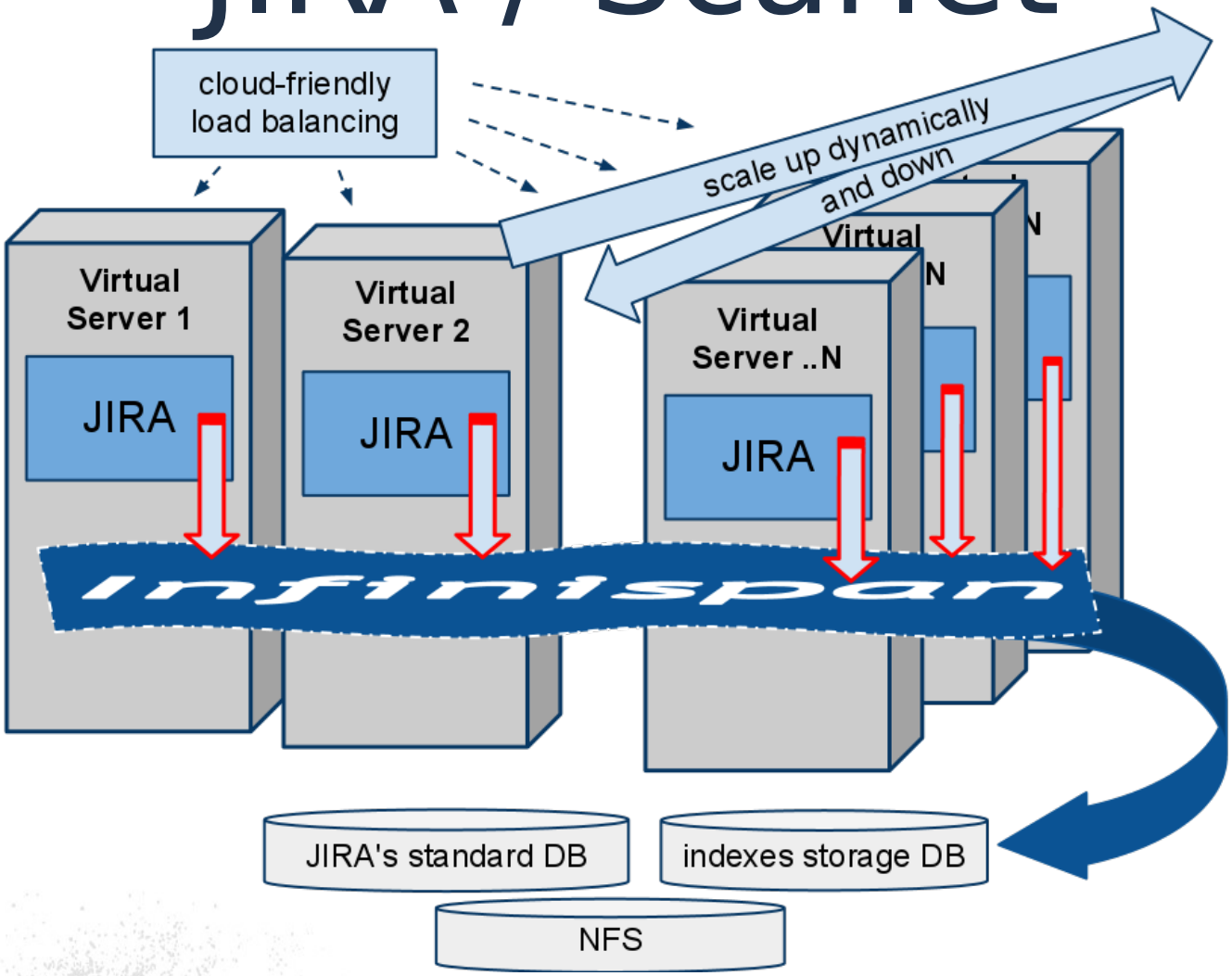



Lucene Directory
Lock Factory
index metadata
segments
Infinispan cluster-wide Lock
Infinispan Cache for locks (Replicated)
Infinispan Read-Lock
index metadata
file metadata
Infinispan Cache for File metadata (Replicated)
_2x.frq (and any other segment file)
_2x.frq [0]
_2x.frq [1]
_2x.frq [2]
_2x.fnm [0]
_2x.fnm [...]
_2x.fnm [n]
Infinispan Cache for segment contents (Distributed: tuned for bulk data store - optionally replicated)
non-volatile storage

Index *stored* in Infinispan



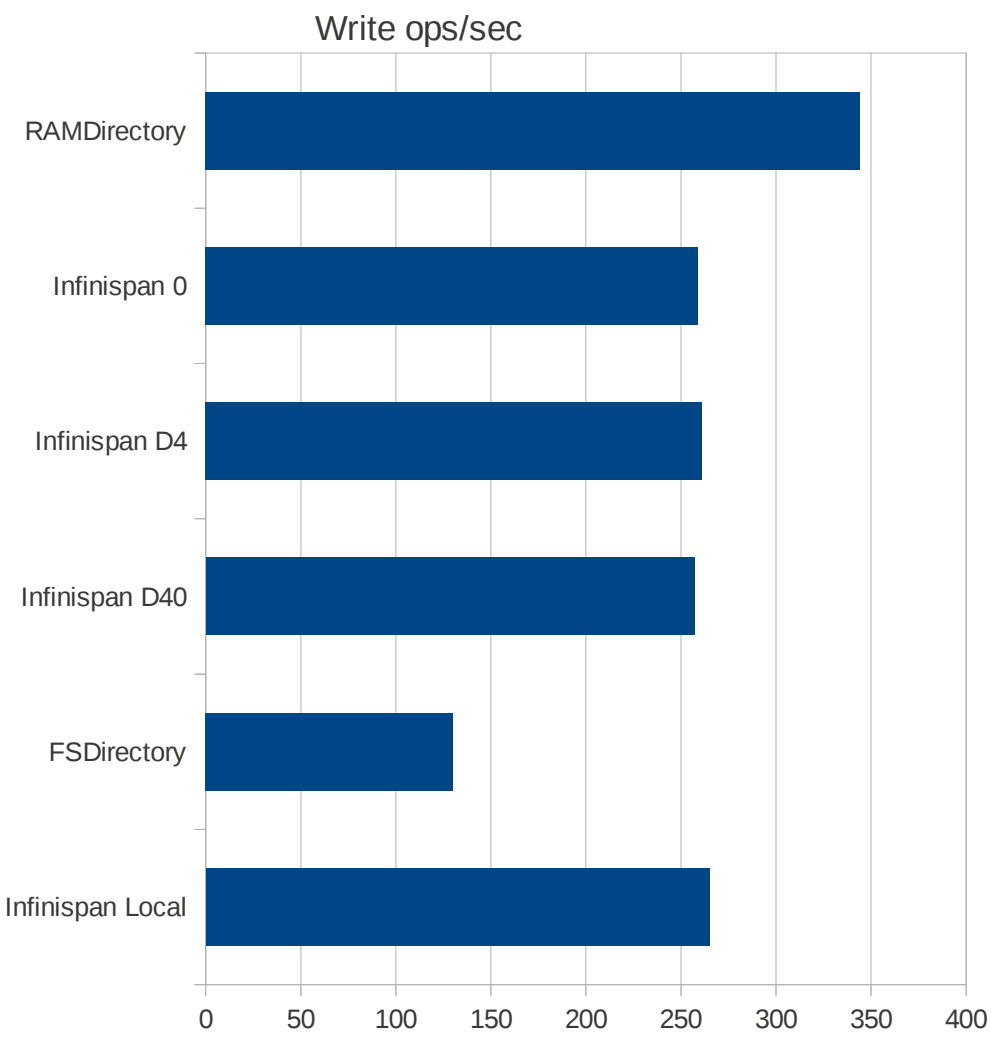
Example architecture : JIRA / Scarlet

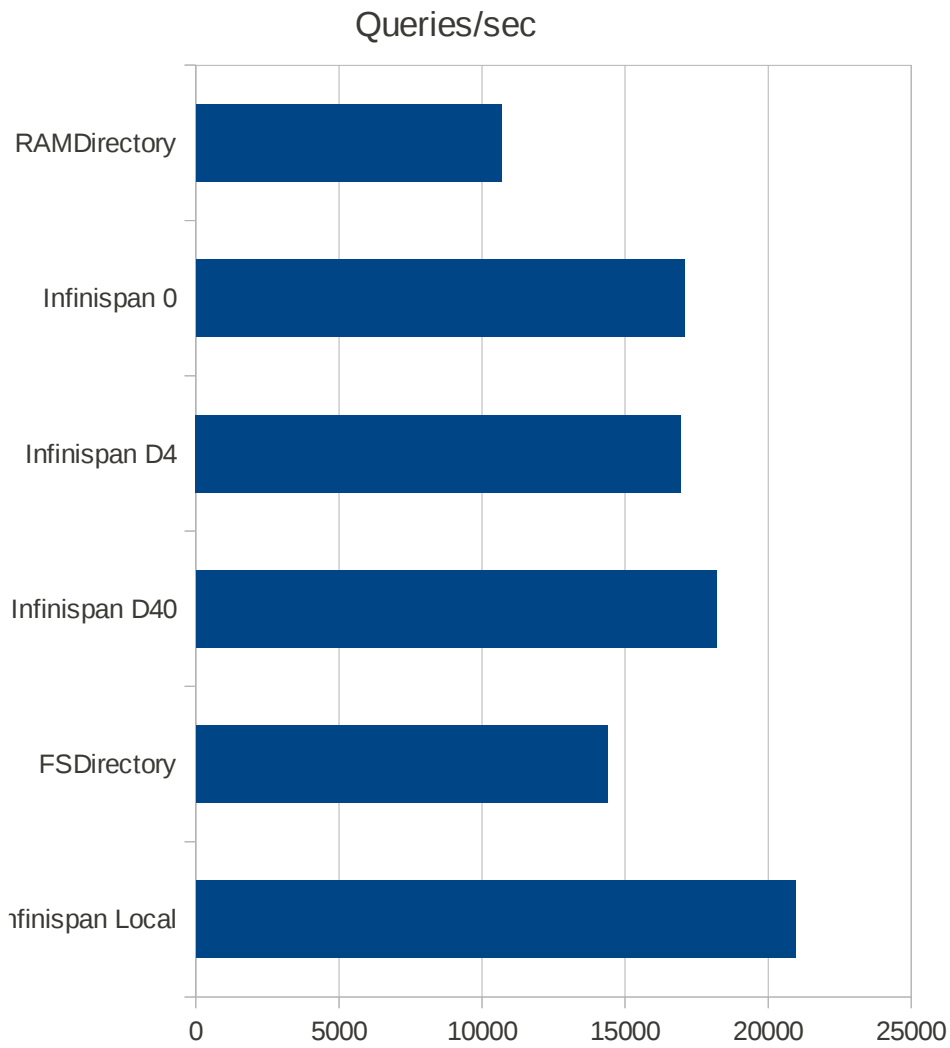


Hints

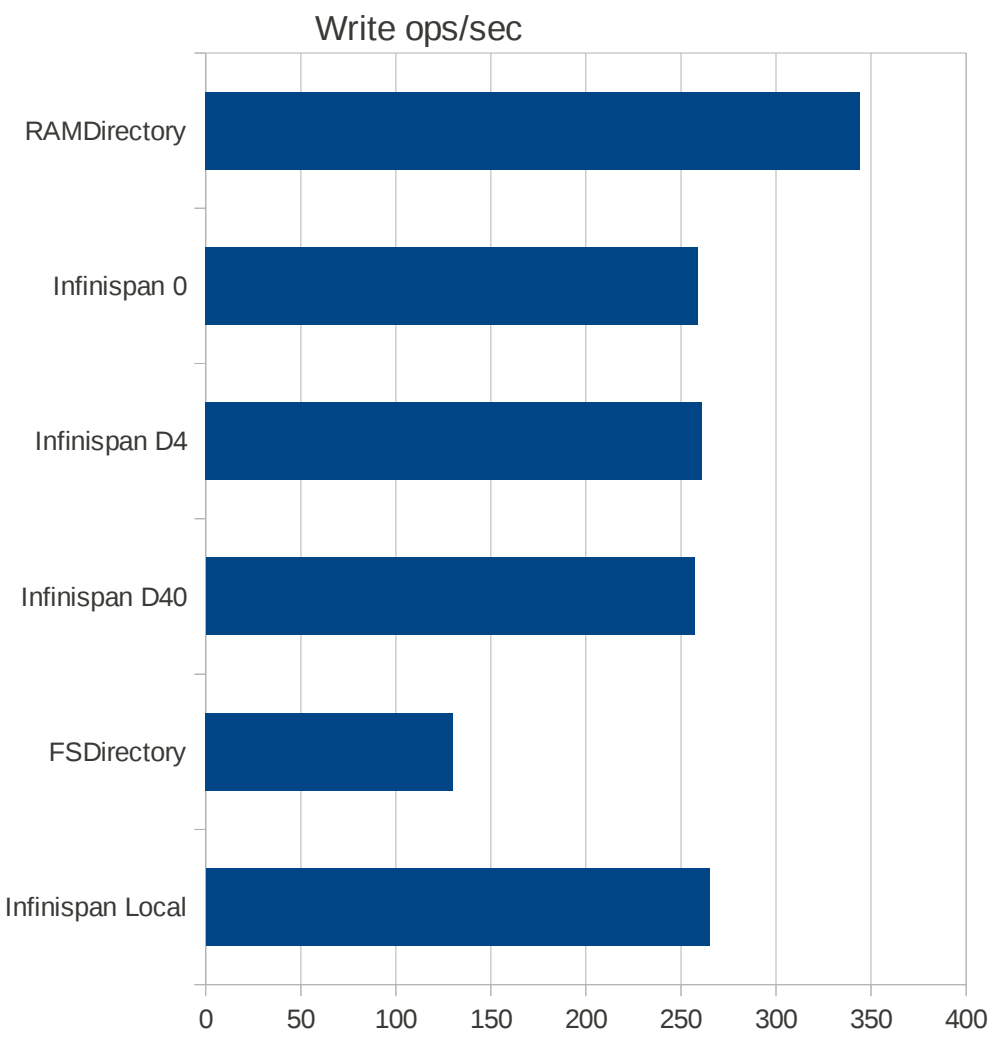
- Some tuning options might have different effects than what you're used
- Network is *orders of magnitude* faster than disk (YMMV)
  - But data locality helps
  - Balance resources
- Get mergers to avoid segment chunking, or readlocks will engage

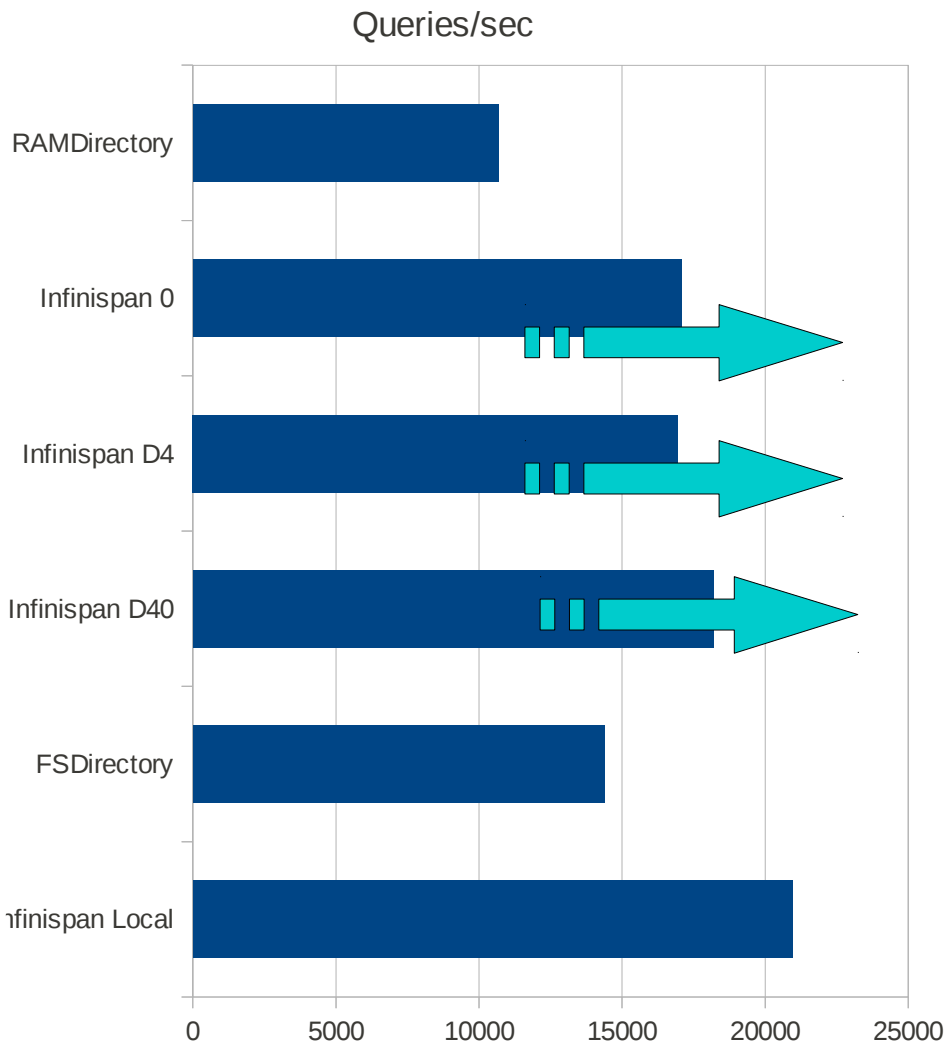
“benchmarks”, stats and more lies





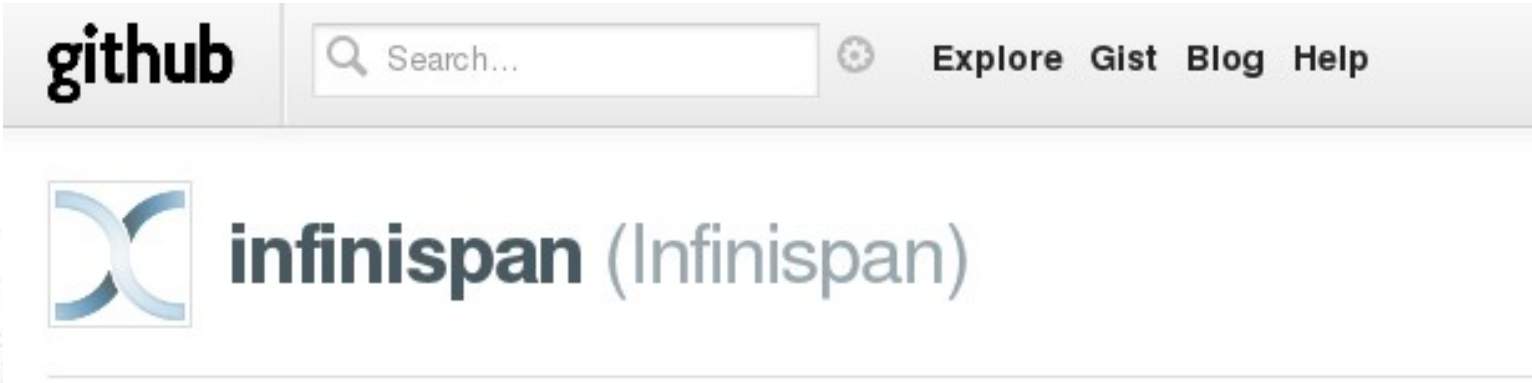
It's not about the figures





What's next?

- Infinispan (core) 5.2 and 6
- Lucene 4.x
- Dynamic chunk sizes
- Ad-hoc “Lucene native” CacheStore
  - NIO byte buffers?



Conclusions

- Quick index replication
- Transactions
- Not a replacements for shards
- Cloud-friendly
  - Delegates to any storage

Q&A

http://infinispan.org
http://in.relation.to
http://jboss.org

@Infinispan
@Hibernate
@SanneGrinovero