



JBoss Cache

Manik Surtani

JBoss, a division of Red Hat

Lead, JBoss Cache

February, 2008

Download demo application used during presentation
here:

http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=102339&release_id=571969

Who is this guy?

What's he going to bore me with today?

What's he going to bore me with today?

- Manik Surtani

Project lead of JBoss Cache

- Agenda:

Data grids and clustering Java, scaling load

Brief overview on JBoss Cache, and how we achieve the above.

- Live demo (hope this works)

Ideas and strategies for making JBoss Cache provide fault tolerance and scalability, and removing bottlenecks

Q & A

Why cache?

- Benefits:
 - Removal of data retrieval or calculation bottlenecks
- Concerns/pitfalls:
 - Cached data validity
 - Cluster-wide coherence
 - “Over-caching”
- Don't cache unless you know you have a bottleneck

Why cluster?

- HA - Fault tolerance is critical
 - 24/7 world
 - Global userbase
 - SLAs
- Scaling horizontally
 - Need to deal with predicted and unpredicted spikes
 - Planned increase of capacity
- Coherence of state and performance of maintaining coherence - two opposing forces

Data grids and clustering

- Hardware is cheap.
 - Both CPUs and memory
- Grid computing is becoming easier to implement
 - Compute grid frameworks like GridGain
 - Work stealing patterns
- Data availability becomes important
 - Data retrieval bottleneck impact increases drastically with grid size
 - Cost of keeping caches coherent increases with cluster size
 - Not as much as data retrieval bottlenecks though

JBoss Cache - the answer!!

- Simple API for caching data

Either locally or cluster-wide

Support for eviction of unused elements

Cache loading/storing to disk to free up memory

Chaining caches, remote caches using TcpCacheLoader

Support for JTA transactions

Listeners and notifications

JMX manageability

Multiple locking schemes (optimistic, pessimistic, MVCC*)

JGroups as a network stack

Flavours

- Core cache
 - for caching simple objects, primitives

- POJO cache

Extension of core cache

Optimised for caching large and complex objects with relationships

- Object relationships maintained, even after replication or persistence

Fine-grained replication

Minimal API

Who uses JBoss Cache?

- JBoss Application Server - Clustering HTTP and EJB sessions, JPA entities
- Hibernate - entity caching
- SEAM - caching JSF generated content
- JBoss Portal
- Lucene/Hibernate Search - cluster-wide indexes
- GridGain - data grid component to complement compute grid
- Coming soon:
 - Drools - clustering rules engines
 - JBoss DNA - clustering metadata repositories
- Many other open source and commercial projects

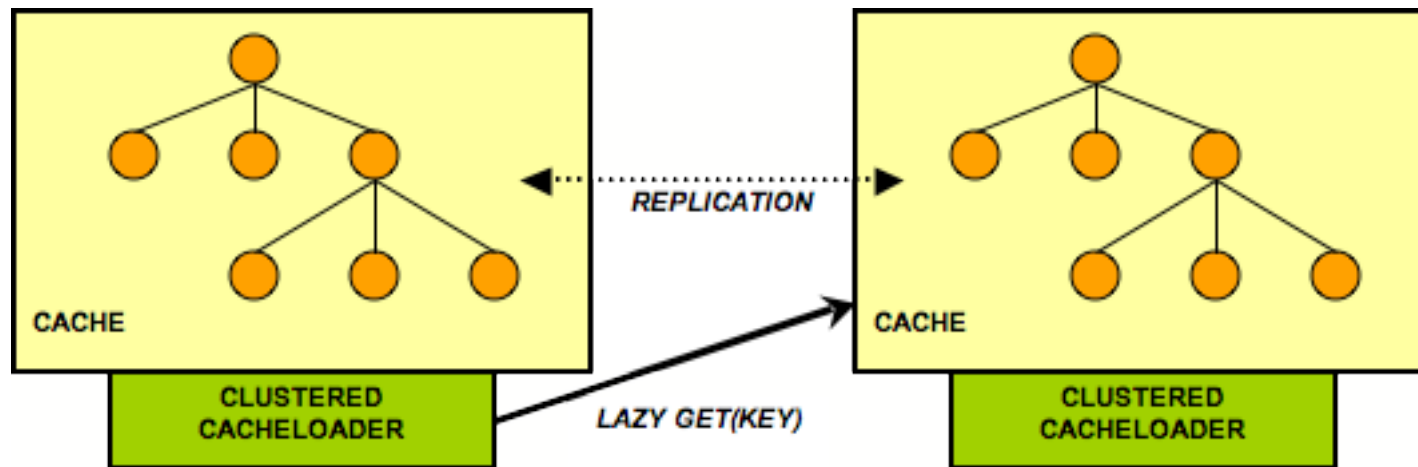
Using JBoss Cache

- Licensing
 - LGPL, truly open source, business-friendly
- 720K core jar
- 17MB ZIP download with documents, srcs, tests, optional extras
- No need for an App Server - just a JVM!
- API - Not going to waste your time with this. :-)
 - Read Javadocs or the User Guide instead
 - www.jbosscache.org
- Demo instead!

High availability strategies

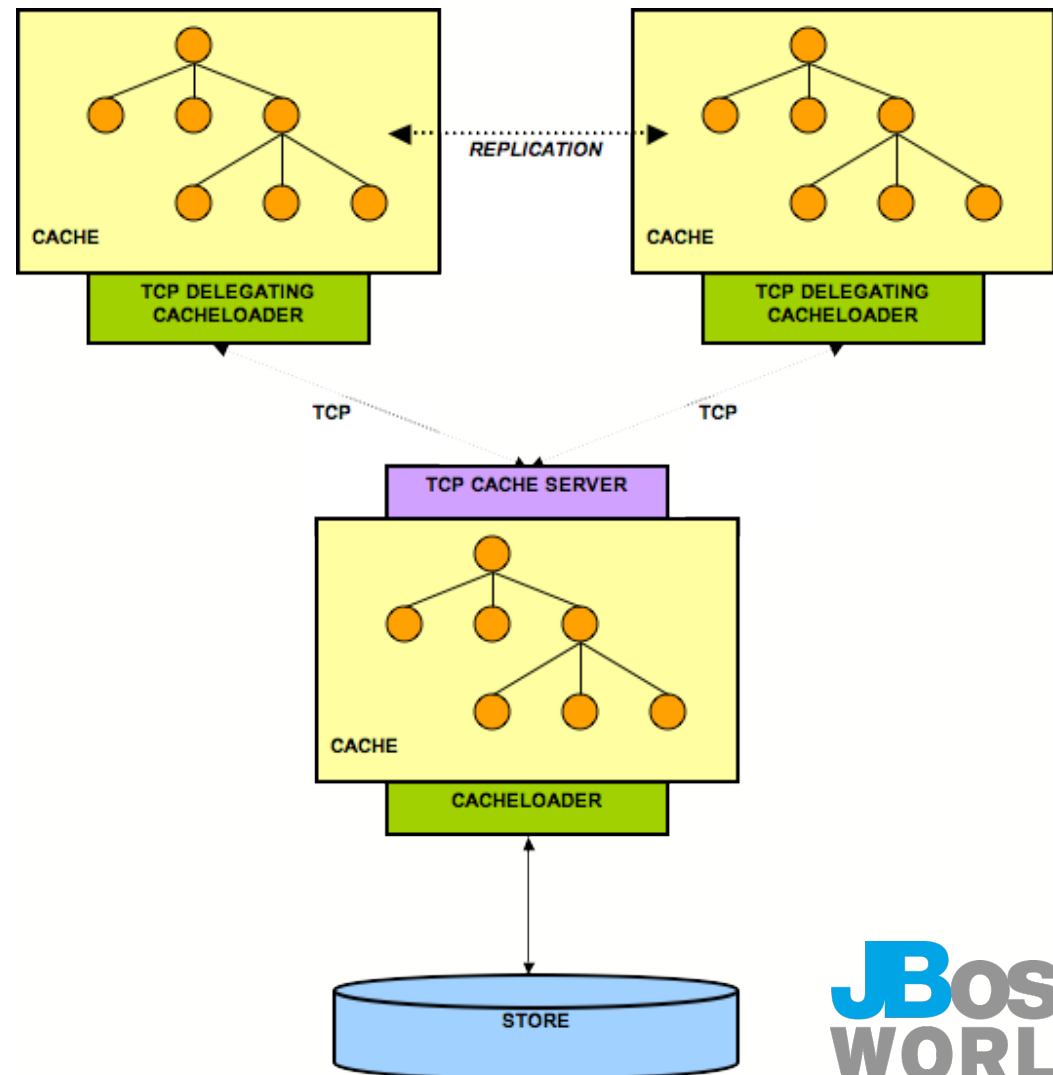
- Lazy state transfer

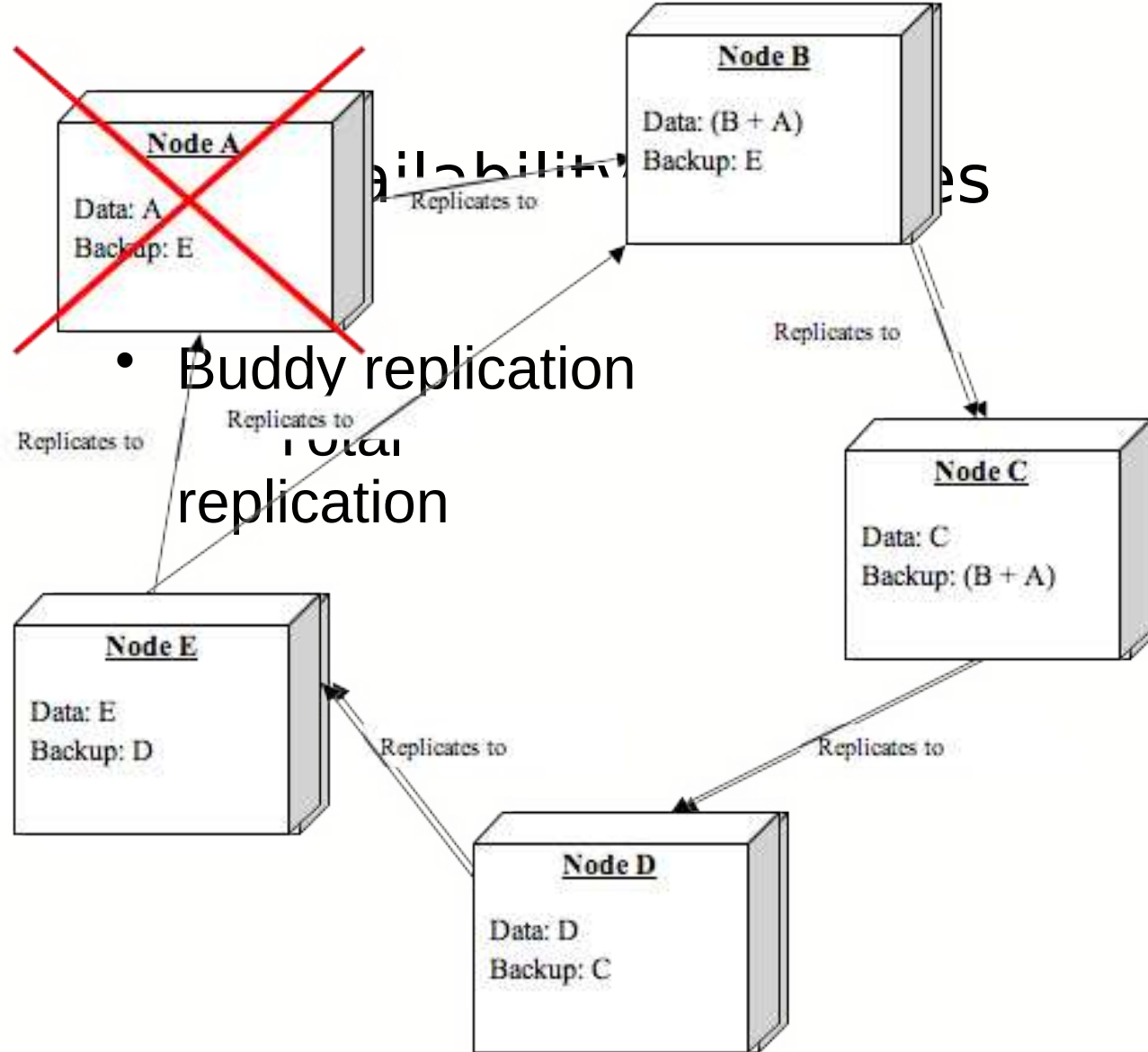
Involves the use of the ClusteredCacheLoader



High availability strategies

- Tiered caches
- Involves the use of TCP delegating cache loader and TCP cache server components





- Buddy replication



Q & A

Download, Use, Contribute, Participate!

www.jbosscache.org

Download demo application used during presentation here:

http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=102339&release_id=571969