PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Using Infinispan for High Availability, Load Balancing, & Extreme Performance

Manik Surtani,
founder and project lead, Infinispan
Galder Zamarreño
core engineer, Infinispan

Red Hat, Inc.

Infinispan

SUMMIT JBoss WORLD
PRESENTED BY RED HAT

Rules of the game

#summitjbw

#infinispan

Who are we?

- Manik Surtani
  - Founder and project lead, Infinispan
  - Project lead, JBoss Cache
  - Frequent speaker on cloud computing, cloud data stores
- Galder Zamarreño
  - R&D engineer on Infinispan and JBoss Cache

The wide world of Infinispan

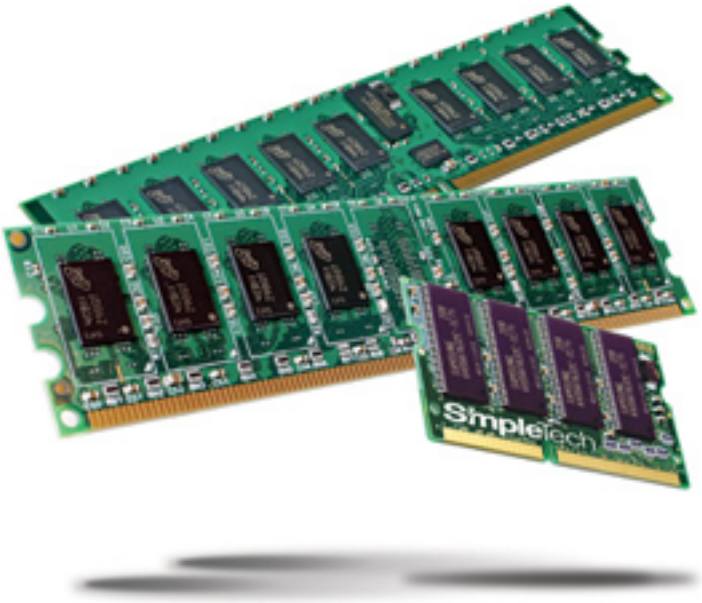
- Categorize major Infinispan uses
- Discuss each one
  - Benefits
  - Typical usage
  - Tuning tricks and tradeoffs

Infinispan as a ...

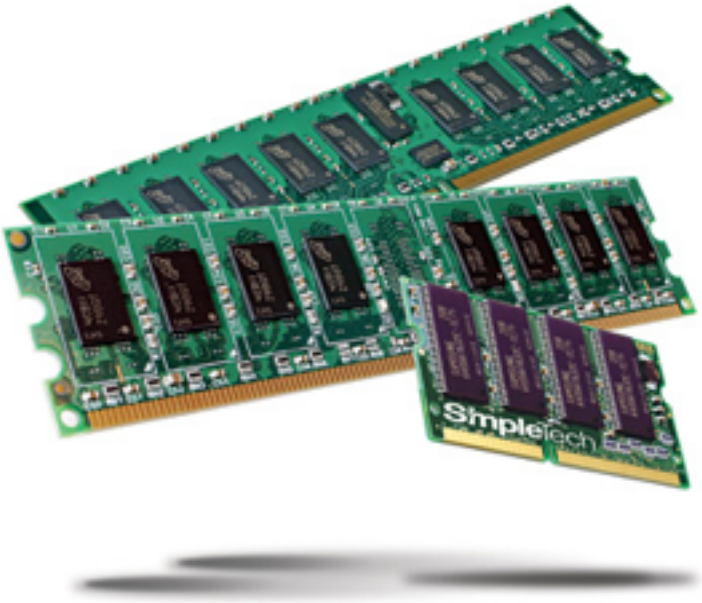
- ... local, in-memory object cache
- ... clustering/high availability toolkit
- ... clustered in-memory cache
- ... in-memory data grid
- ... cloud-ready data store

Local, in-memory object cache

- Performance booster
- Good for data that is:
  - Hard to calculate
  - Expensive to retrieve
    - E.g., from a DB or a Web Service
  - Frequently accessed



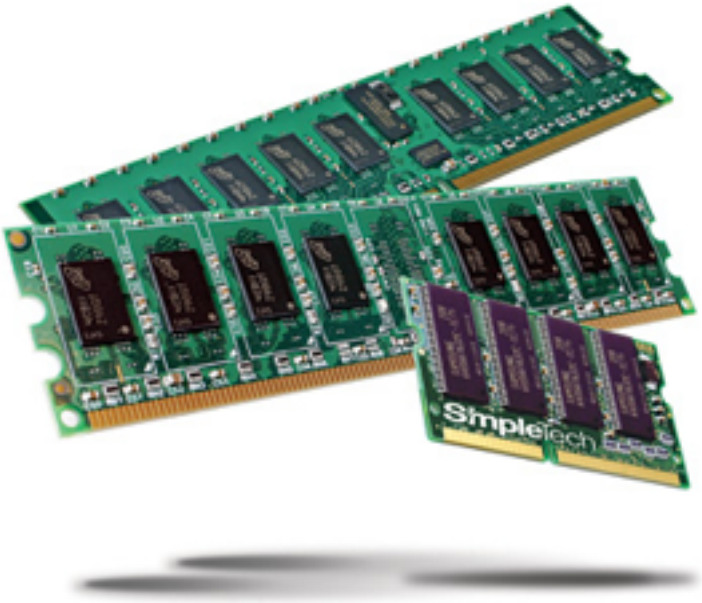
Local, in-memory object cache



- Better than a HashMap
  - Greater concurrency
  - Built-in eviction, prevents OOMs!
  - Overflow to disk
  - Warm starts, preloading
  - Events, notifications
  - Highly configurable locking strategies
  - JTA compatible
  - JMX monitoring

Local, in-memory object cache

- Plugs in to various frameworks to boost performance
  - Hibernate/JPA
  - SEAM
    - Cache JSF fragments
  - Wicket
  - Apache Camel, ESBs
  - ... etc ...



Tips and Tricks

- Make use of eviction
  - Low-cost, bounded container
  - Recency-based eviction: LIRS
- Tune for read-heavy
  - Use lock-striping, minimize lock pool
  - READ_COMMITTED is good enough for most
- Preloading can be expensive
  - Slow startup
  - Is it really needed? Is lazy population good enough?

Clustering toolkit

- Helps cluster your framework
- Helps you add the following features:
  - High Availability
  - Failover
  - Scale-out

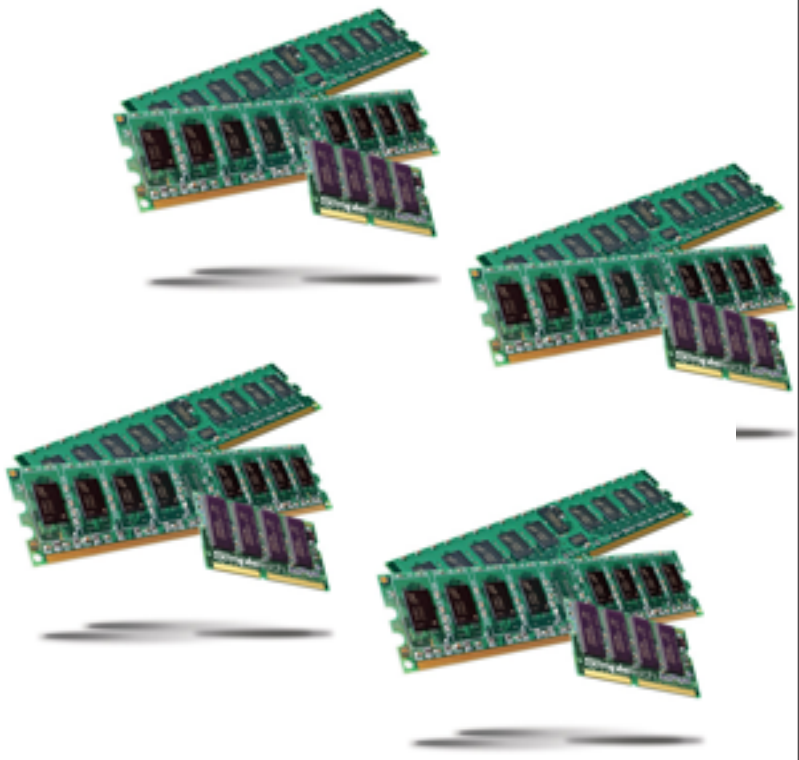
Clustering toolkit

- E.g., clustering EJB and Servlet containers
  - JBoss AS, others
- SIP session state
  - Mobicents
- Lucene directory
- ModeShape, JCR
- Plenty of custom frameworks, servers
  - MMORPG servers
  - Financial trading systems

Tips and Tricks

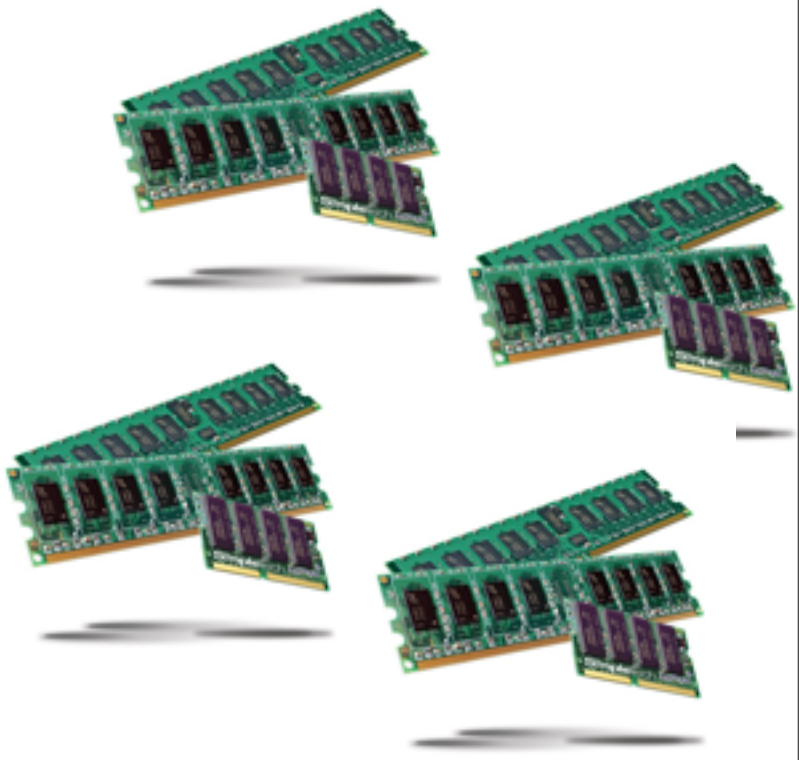
- Strive for session affinity
  - A very valid optimization
  - Will allow for async comms
- Replicated mode vs distribution?
  - Depends on cluster size
- Distribution: co-locate related state
- Preloading, state transfer not necessary, use a `ClusterCacheLoader`

Clustered in-memory cache



- Performance booster
- Similar to a local cache
- Cluster-aware
- More shared-cache space!

Clustered in-memory cache

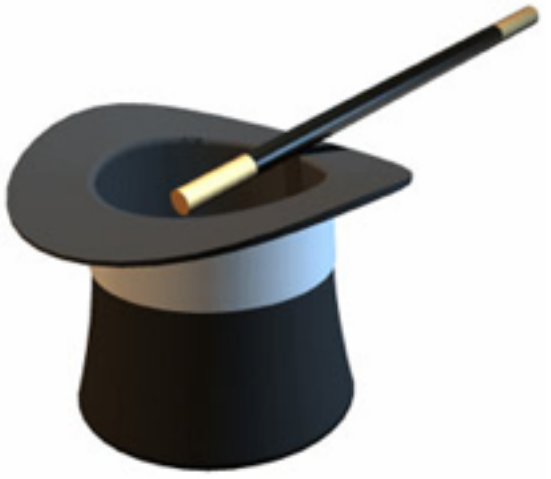


- Just as an in-memory cache:
  - Cache data that is costly to:
    - Calculate (e.g., deserialize)
    - Retrieve (e.g., RDBMS, Web Svc)
- Hibernate/JPA
  - cluster-aware 2nd-level cache
- Apache Camel, ESBs

Cache.putForExternalRead()

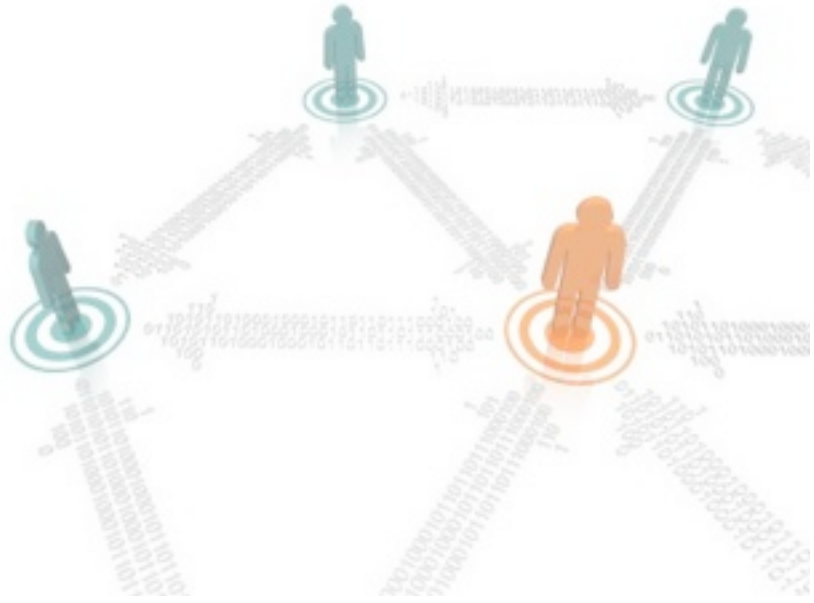
`put()`	`putForExternalRead()`
Use for updating state	Use to cache state read from external source
Regular lock acquisition timeout	Fail-fast
Could throw an exception	Fails quietly
Could cause existing transaction to fail	Will never affect existing transactions

Tips and Tricks



- Similar to an in-memory cache
- putForExternalRead() API
- Use invalidation clustered mode
  - Very efficient: only keys on wire
  - async comms help even further
- Replication can be used as well
  - if the cluster size is small
  - the data cached is small
  - overall data volume can be contained in a single node

In-memory data grid

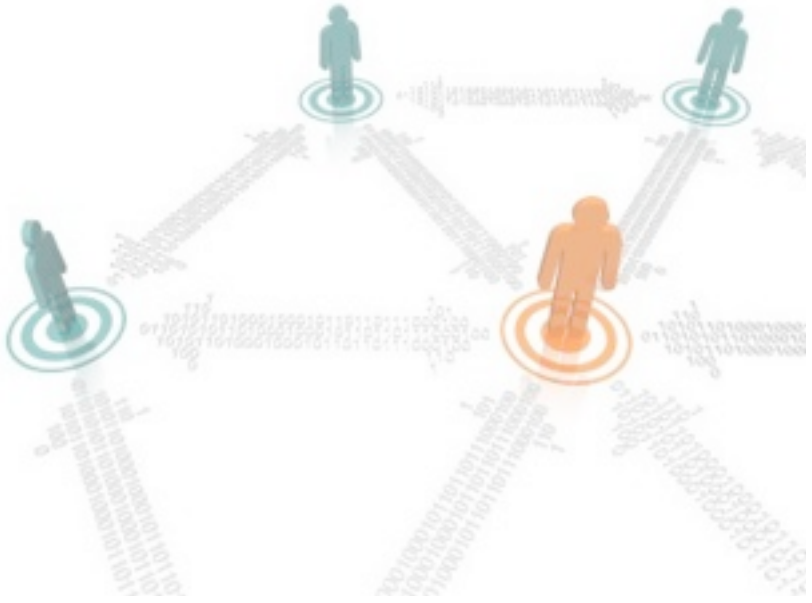


- Alternate data store
- Not just a cache
  - An authoritative data sink
- Highly scalable, low latency
- Accessed in a P2P *or* client/server manner

"Memory is the new disk, disk is the new tape!"

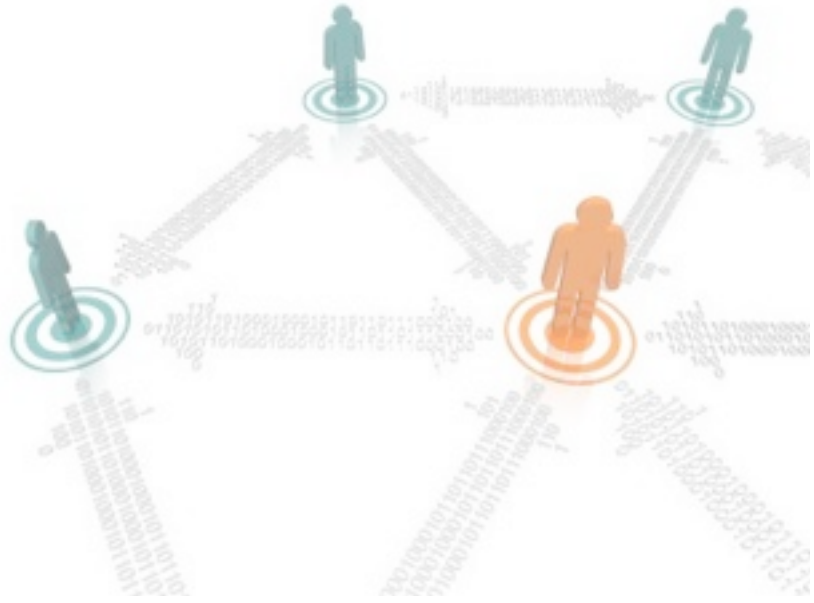
- Tim Bray

In-memory data grid



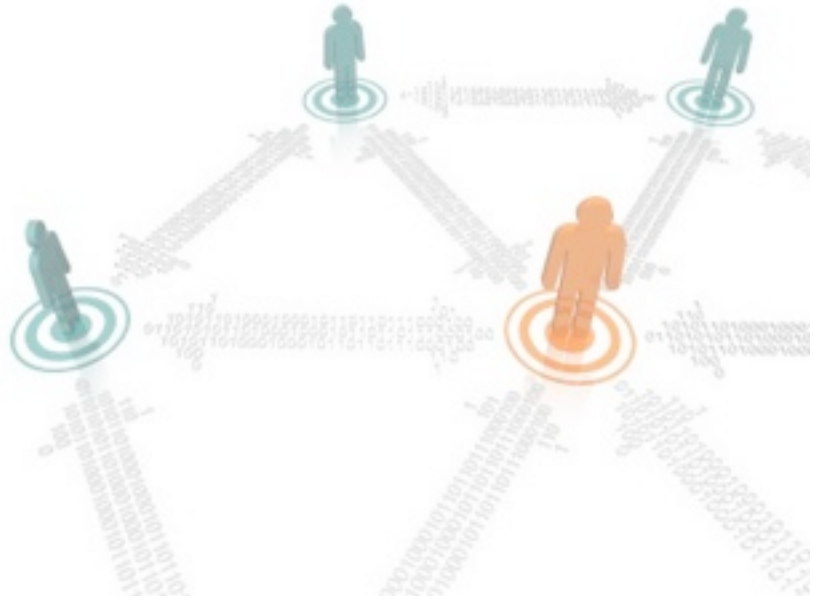
- Multiple access mechanisms
  - Embedded: P2P
  - Client/Server
    - REST
    - Memcached
    - HotRod
- Familiar APIs
  - Cache API, upcoming JPA-like API
- Queryability

In-memory data grid



- Four major characteristics
  - **F**ast
    - In-memory, low latency
    - high concurrency
  - **A**vailable
    - Inherent redundancy
  - **D**istributed
    - Data locality
  - **E**lastic
    - Scalable. Out and back in again.

In-memory data grid



- Used in HPC environments
  - Latency is key
  - High performance web
  - Consumer-facing systems
- Massive compute grids
  - Distributing computation needs distributed data
  - Financial systems, risk engines, arbitrage systems
  - Scientific use

Tips and Tricks

- Session affinity still a nice to have
- Async comms and eventual consistency
- Use distributed mode
  - Replicated will work to a limit
- Tune performance vs durability
  - numOwners
- Dedicated data tier helps you build stateless, elastic app tier
- HotRod: most efficient endpoint. Smart clients.

Cloud-ready data store

- RDBMS in clouds suck
  - Ephemeral cloud nodes
  - Inelastic
  - single point of failure
- Data grids a much better solution
  - Deals with transience
  - Elastic, scalable
  - Distributed

Cloud-ready data store

- Google App Engine
  - BigTable
- Amazon SimpleDB
  - Dynamo
- Facebook
  - Cassandra
- etc.

Cloud-ready data store

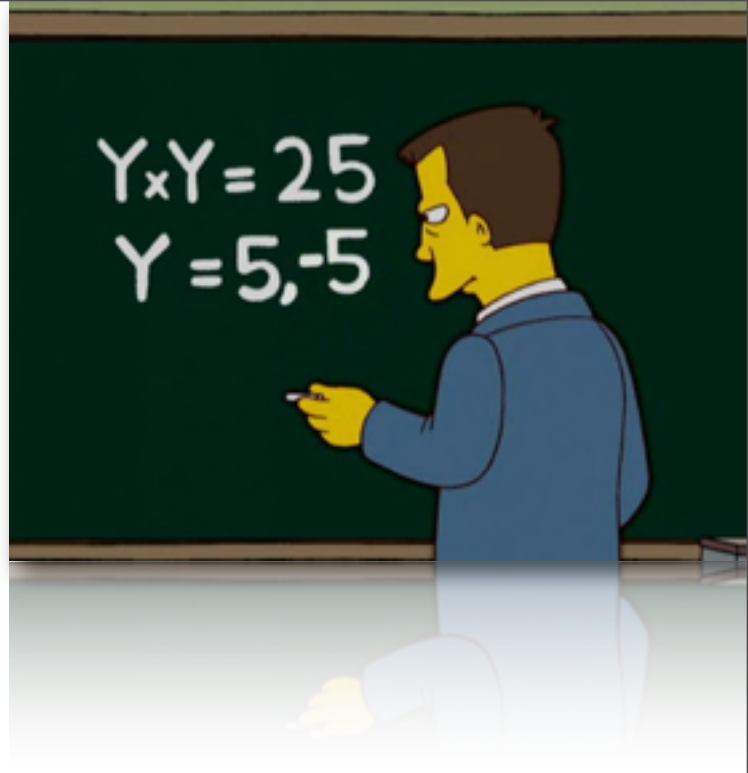
- Like Data Grids, API is key
  - Cache API
  - JPA-like API
  - REST
  - HotRod
- Dedicated data grid tier in the cloud
  - Allows for scalable, stateless app tier

Tips and Tricks

- Same as with Data Grids
- Multiple endpoints: a variety of clients
  - REST
    - Fronted by a load balancer
  - HotRod
    - Built-in load balancing
- Use a transport that doesn't rely on Multicast
  - Not supported by some cloud infrastructure

To sum it up...

- Infinispan can be used in a variety of ways
  - From a simple in-memory cache
  - ... to a cloud-ready data store
- To increase performance
- ... or to provide high availability, elasticity



Questions?

- http://www.infinispan.org
- http://blog.infinispan.org
- http://twitter.com/infinispan
  - #infinispan

Learn more about Infinispan!

- *How to Stop Worrying & Start Caching in Java* - Thu 24th
- *Why RESTful Design for Cloud is Best* - Fri 25th

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#summitjbw

READ THE BLOG

http://summitblog.redhat.com/