PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Highly Scalable Data Grids and Distributed Caching with Infinispan

Manik Surtani,
founder and project lead, Infinispan

Red Hat, Inc.

Infinispan

SUMMIT JBoss WORLD

PRESENTED BY RED HAT

Rules of the game

#jbw

#infinispan

Who am I?

- R&D Engineer, Red Hat Inc.
- Founder and project lead, Infinispan
- Project lead, JBoss Cache
- Frequent speaker on cloud computing and cloud data storage

http://twitter.com/maniksurtani

http://blog.infinispan.org

The wide world of Infinispan

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

What is Infinispan?

- Open source (LGPL) in-memory Data Grid
- Some concepts from Amazon Dynamo

2 usage modes

- Embedded
- Client-server
  - memcached
  - Hot Rod
  - REST

Consistent hash based distribution

- Self healing
- No single point of failure

Highly concurrent

- MVCC locking

XA Transactions

- 2-phase commit based
- Deadlock detection algorithms
- Coming soon: Atomic Broadcast

Map/Reduce

- Clean, fluent API to deal with an otherwise complex programming model

Querying

- Using Lucene and Hibernate Search

Persistence

- Not *just* in memory!
- Write through and write behind
- Pluggable "*drivers*"

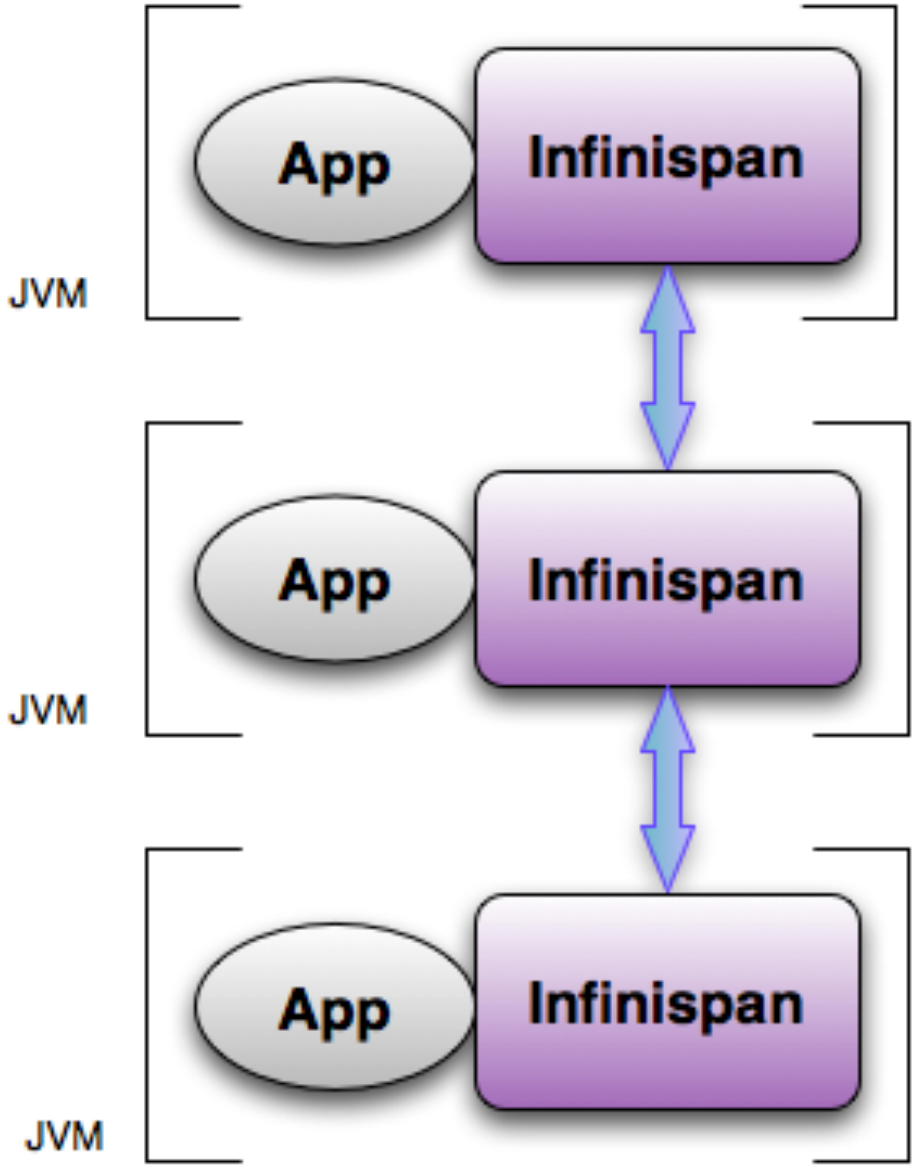
Eviction and expiry

- Efficient, adaptive algorithms
- Addresses shortcomings of LRU & FIFO

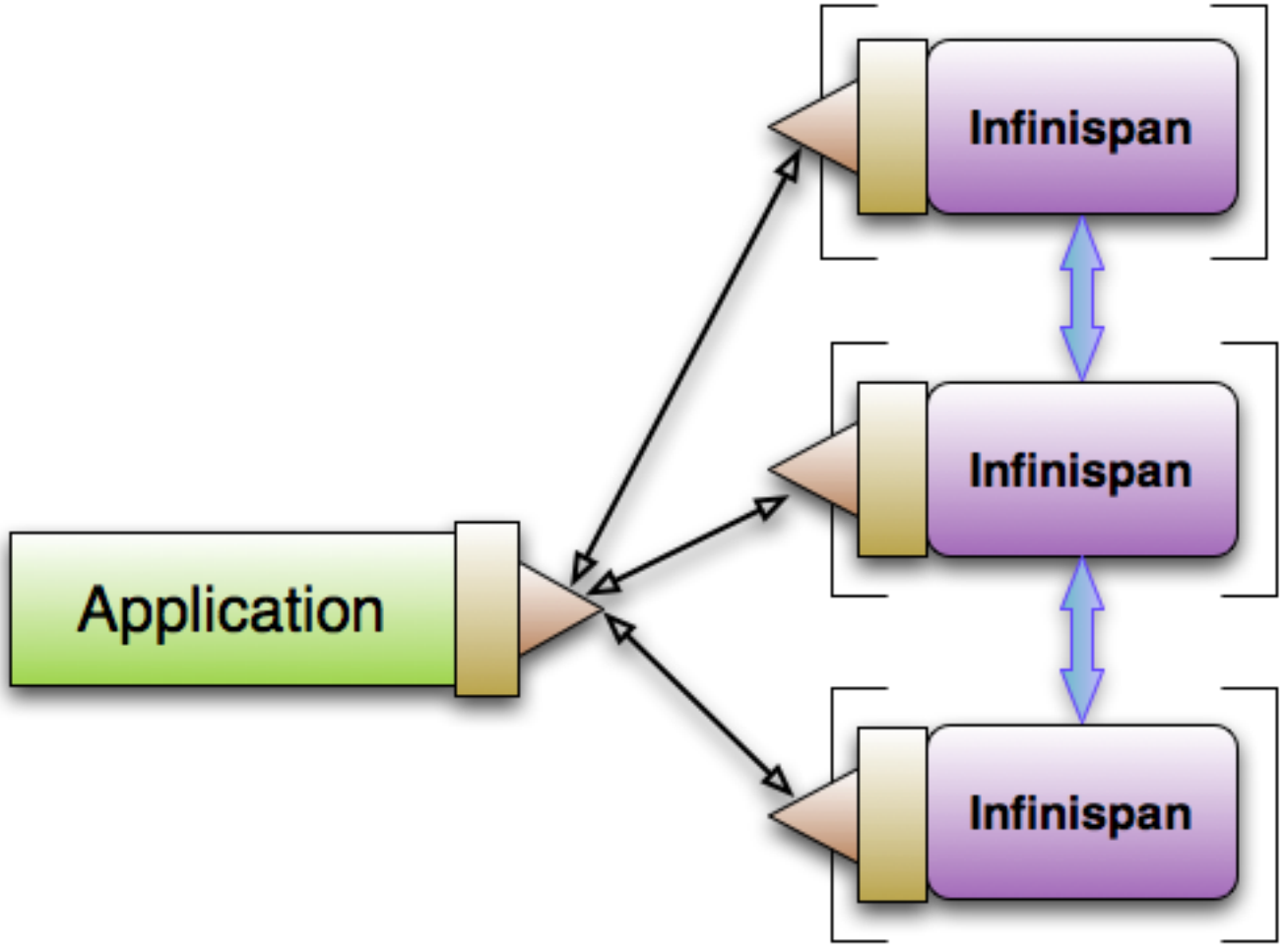
API

- Map-like key/value store (JSR 107)
- Proposed data grid API (JSR 347)
- Upcoming JPA-like layer
  - Hibernate OGM

Peer To Peer Architecture



Client/Server Architecture



Supported Protocols

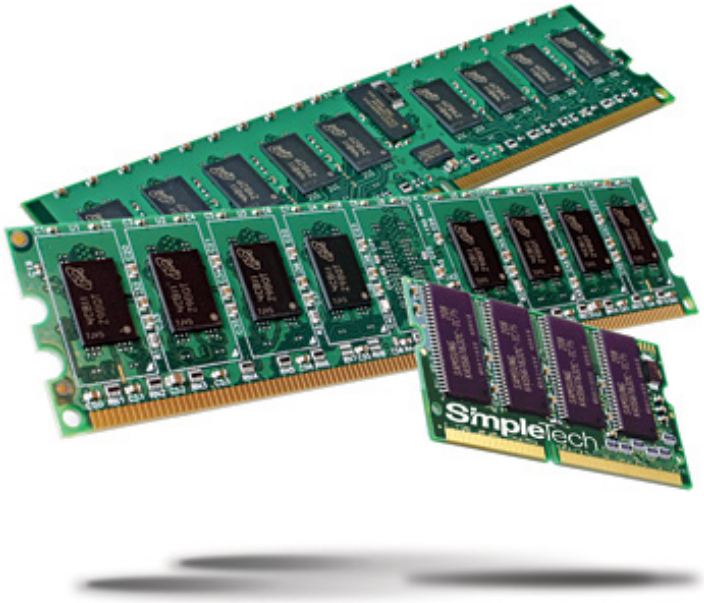
- REST
- Memcached
- Hot Rod

Infinispan as a ...

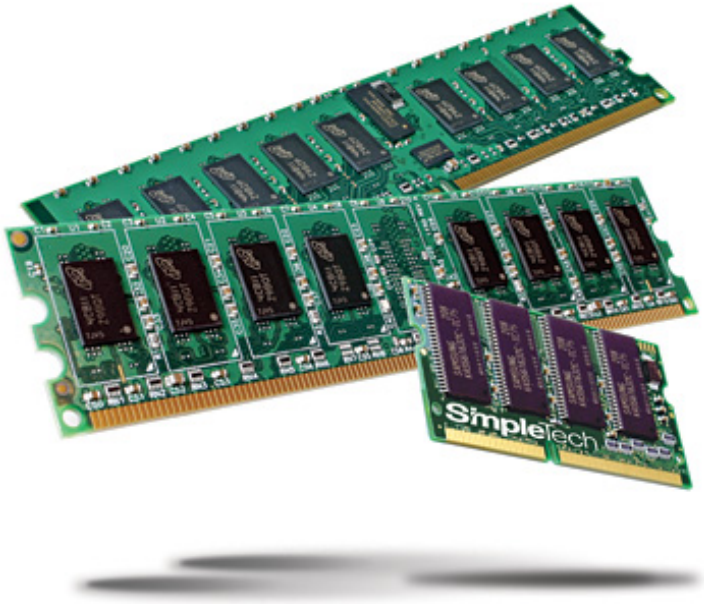
- ... local, in-memory object cache
- ... clustering/high availability toolkit
- ... clustered in-memory cache
- ... in-memory data grid
- ... cloud-ready data store

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



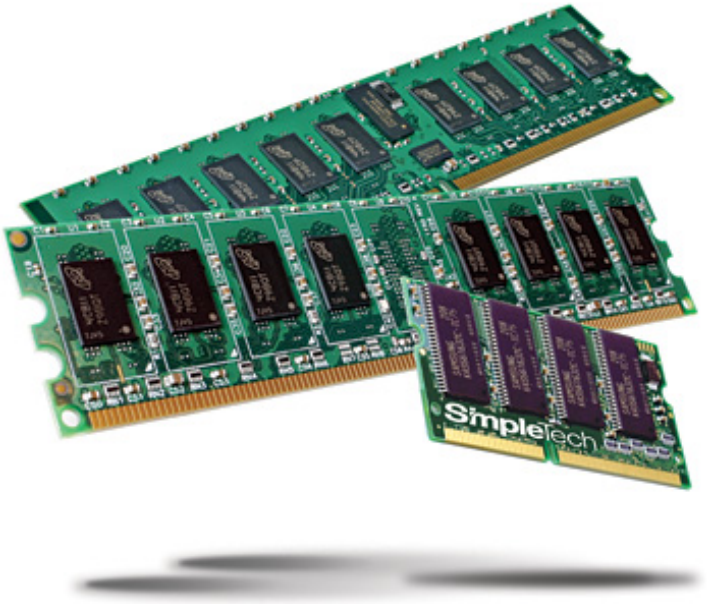
Local, in-memory object cache



- 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 ...



Clustering toolkit

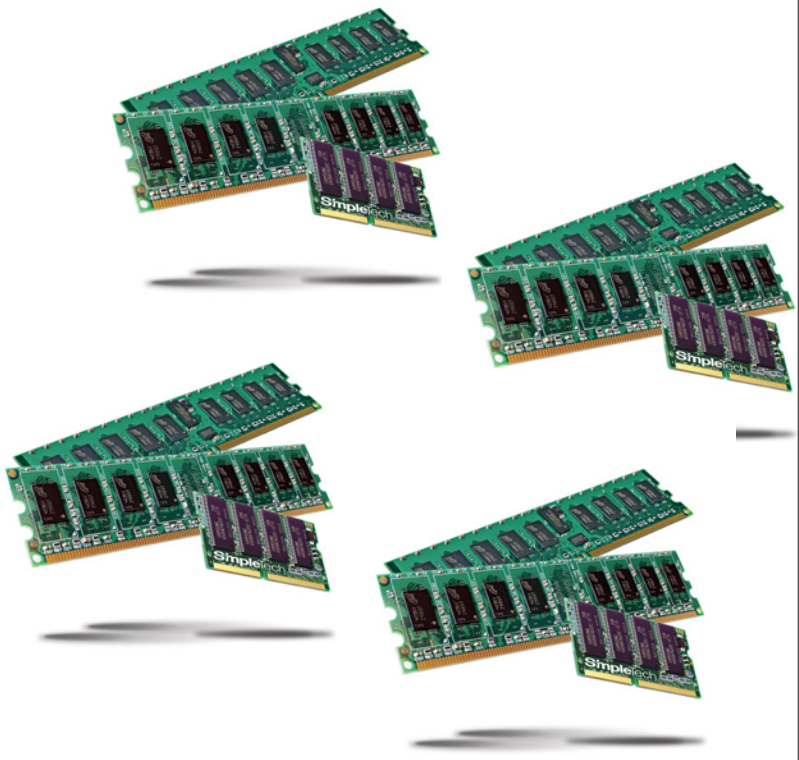
- Helps cluster your framework
- Helps you add the following features:
  - High Availability
  - Failover
  - Scale-out

Clustering toolkit

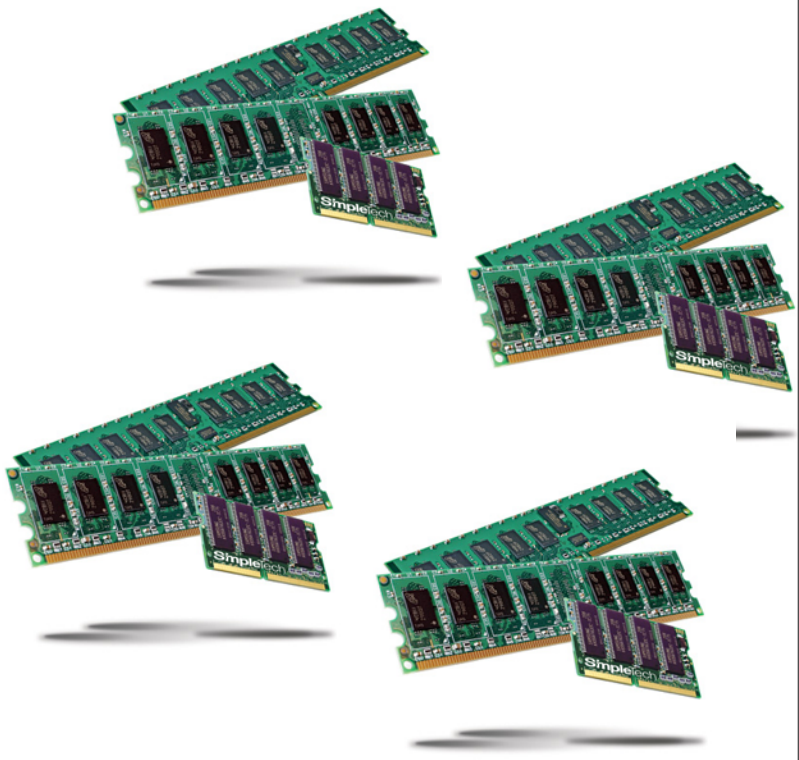
- E.g., clustering EJB and Servlet containers
  - JBoss AS, others
- SIP session state
  - Mobicents
- Lucene directory
- ModeShape, JCR
- Plenty of custom frameworks, servers
  - MMORPG servers
  - Financial trading systems

Clustered in-memory cache

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

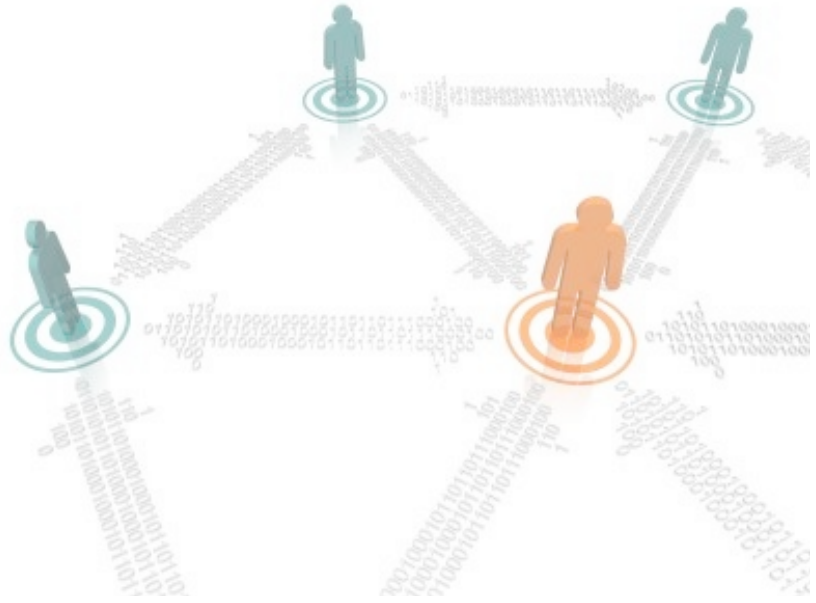


Clustered in-memory cache



- 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

In-memory data grid

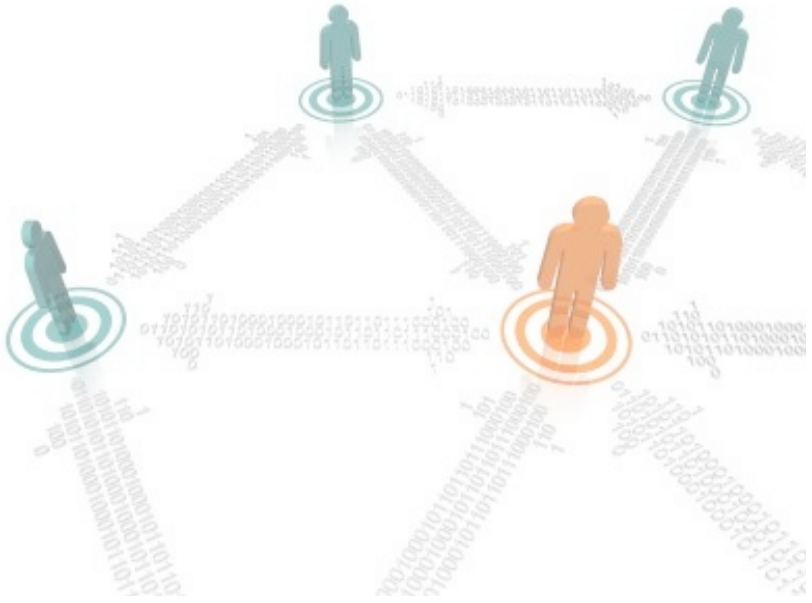


- Alternate data store
- Not just a cache
  - An authoritative data sink
- Highly scalable, low latency
- Accessed in a P2P *or* client/server manner

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

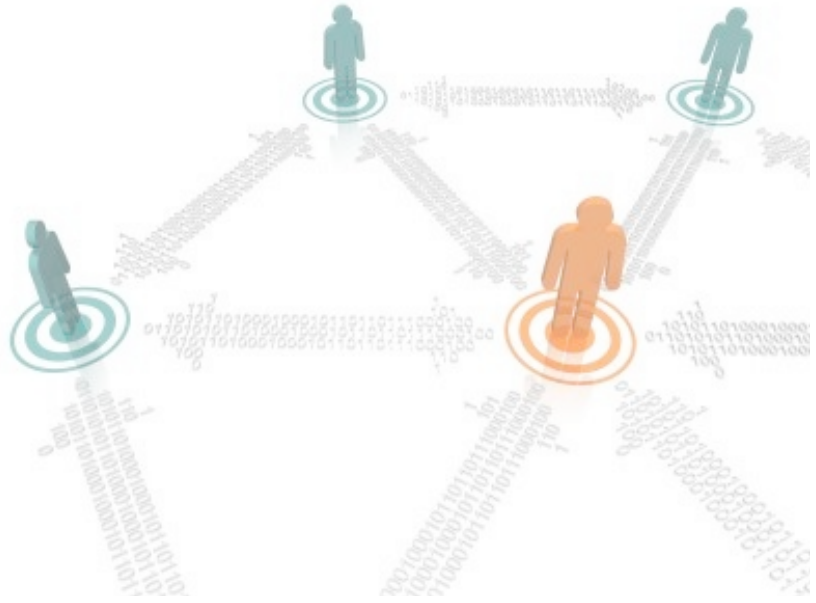
- Tim Bray

In-memory data grid



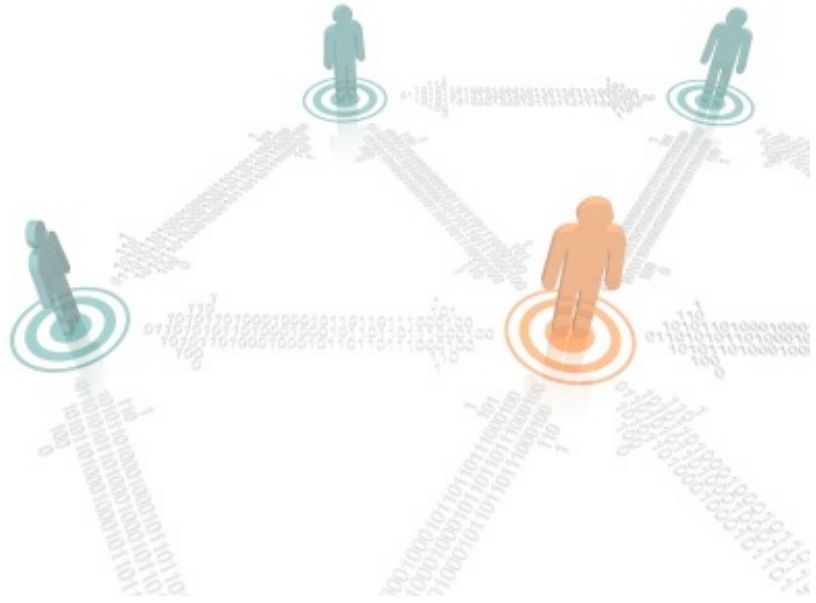
- Multiple access mechanisms
  - Embedded: P2P
  - Client/Server
    - REST
    - Memcached
    - Hot Rod
- Familiar APIs
  - Cache API (JSR 107)
  - JPA-like API (Hibernate OGM)
- Queryability

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

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

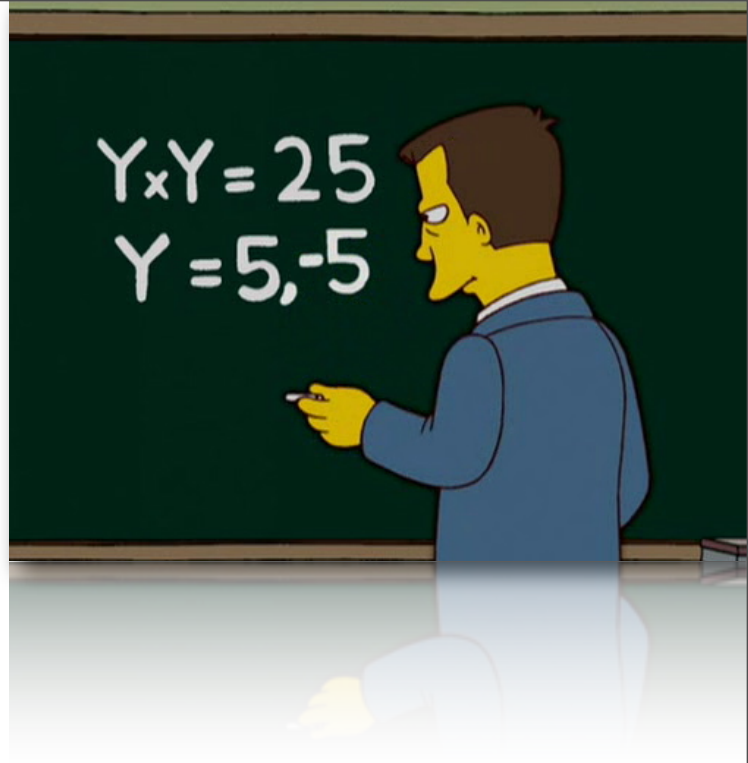
- Google App Engine
  - BigTable
- Amazon SimpleDB
  - Dynamo
- Facebook
  - Cassandra, HBase
- etc.

Cloud-ready data store

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

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

Rate this talk! http://spkr8.com/t/7408

JOIN THE CONVERSATIONS AND STAY IN TOUCH WITH JBOSS YEAR ROUND!

LIKE US ON FACEBOOK

facebook.com/jboss

FOLLOW US ON TWITTER

@JBossDeveloper
@JBossOperations
@JBossNews

READ THE BLOG

redhat.com/about/news/blog/
jboss.org/feeds/