Infinispan

The future of open source data grids

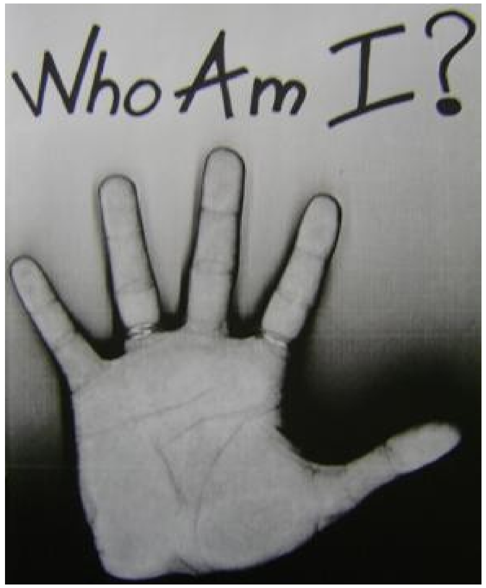
Manik Surtani

manik@jboss.org
Principal Software Engineer
Red Hat

presented by

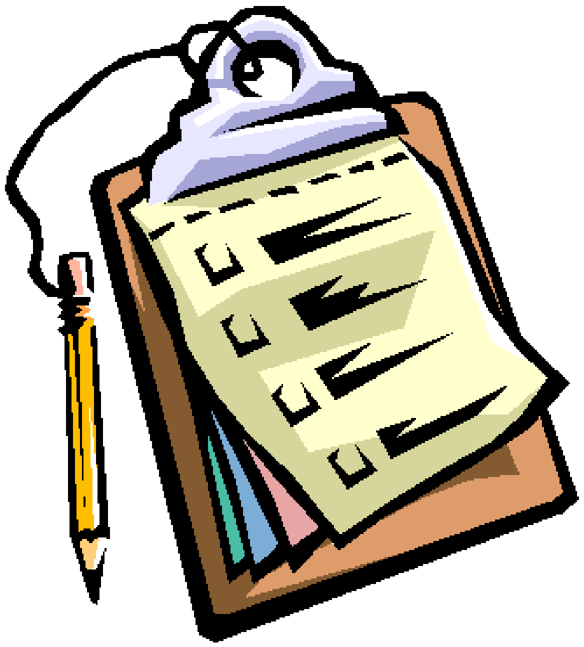
Who is Manik?

- Founder and project lead, Infinispan
- Project lead, JBoss Cache
- Contributor and committer on various OSS projects
  - JBoss AS
  - JGroups
  - Hibernate
  - etc.



Agenda

- Cloud computing and data grids
  - And why YOU should care
- Introducing Infinispan
  - And how this relates to JBoss Cache
- The path ahead for Infinispan
  - Roadmap
  - Featureset



Clouds are today!

- Clouds are happening
  - *aaS: SaaS, PaaS, IaaS
- You cannot escape them!
  - Public: Amazon, Google, GoGrid, Rackspace
  - Private: Eucalyptus, VMWare, IBM
- Traditional datacenters marginalized to niche deployments
- Clouds become mainstream

Why are clouds popular?

- Piecemeal cost
  - Pay for what you use
- Massive, global data centers means high availability, instant backups
- Everyone benefits from economies of scale
- Ability to scale on demand
- Very fast provisioning
- Proven charging model
  - Remember timesharing on mainframes?

So why now?

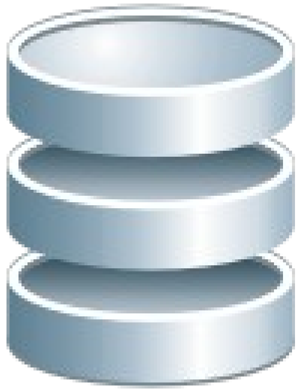
- We’re in a perfect storm
- Bandwidth is cheap and plentiful
- OS virtualization is mature
- ... and we’re in a financial crisis!
  - Everyone wants to cut costs, be more efficient!
  - Making changes is easier now

Why should I care?

- The platforms I use will still be relevant:
  - Java, Java EE
  - Python, Ruby, .NET
  - ... whatever!!
- The OS I use will still be relevant
  - Linux
  - Solaris
  - etc.

Data Storage

- Clouds are inherently stateless and ephemeral
- Databases on clouds don't make sense
  - Traditional modes of data storage won't work
- Scalability is crucial
  - Databases <u>still</u> are a bottleneck
  - … <u>and</u> single point of failure!



Trying to make databases work in the cloud

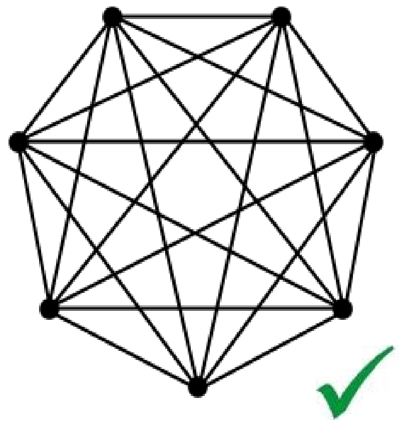
- E.g., with AWS, store data files on:
  - a mounted EBS volume
    - EBS is not guaranteed to be durable
    - Still needs to be backed up
      - Snapshots expose data loss windows
      - Locks the volume from being written to
    - Can only be mounted by one EC2 node at a time
      - Single point of failure
      - Bottleneck

Trying to make databases work in the cloud

- E.g., with AWS, store data files on:
  - a mounted S3 bucket
    - High latency
      - Web service or REST based comms to S3!
- Native database clustering
  - Notoriously slow and non-scalable
  - Unreliable
  - Expensive!
  - Need special hardware, e.g., SAN

The solution: Data Grids!

- Data grids are perfect for clouds
  - Highly scalable
  - No single point of failure
  - Works with ephemeral nodes
  - Very low latency
- Data grids
  - Amazon SimpleDB uses Dynamo
  - Infinispan, etc.
  - Many other commercial and open source offerings



Data Grids - Speed!

- Data grids give you speed!
- Very low latency due to minimal disk lookup
  - Memory 2 orders of magnitude faster than disk
  - Especially for frequently used data
- Far greater concurrency
  - Disk IO is always a concurrency bottleneck
  - Memory offers far greater concurrency

JBoss WORLD CHICAGO 2009

Introducing Infinispan

presented by

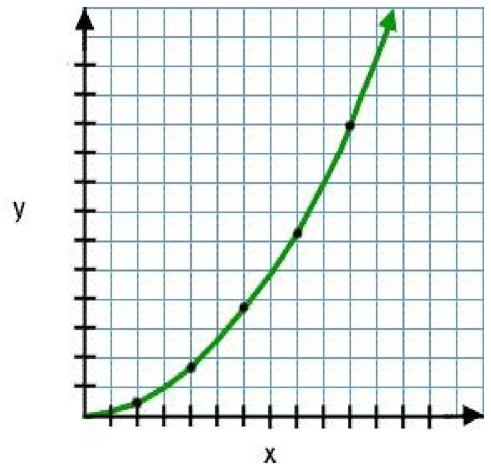
Introducing Infinispan

- Highly scalable data grid platform
  - 100% open source licensed (LGPL)
  - Based on some JBoss Cache code
    - But mostly all-new!
- JBoss Cache is a clustered caching library
  - Infinispan is a data grid platform
- JBoss Cache uses a tree-structured API
  - Infinispan is a Map. Like JSR-107's JCACHE

Infinispan != JBoss Cache 4

- Internal data container design completely different
- APIs completely different
- Not backward-compatible
  - Although an code-level compatibility layer is available

More scalable than JBoss Cache



- Internal structures more memory efficient
  - Data organised in Map-like structures
    - Making use of CAS
    - minimising synchronized blocks, mutexes
  - Containers are naturally ordered
    - Makes eviction much more efficient
- Uses JBoss Marshalling
  - Smaller payloads + poolable streams = faster remote calls

“Borrowed” from JBoss Cache

- JTA transactions
- Replicated data structure
- Eviction, cache persistence
- Notifications and eventing API
- JMX reporting
- Fine-grained replication
- MVCC locking
- Non-blocking state transfer techniques
- Query API
- Custom (non-JDK) marshalling

NEW!

… and new features!

- Consistent hash based data distribution
- Much simpler Map API (JSR-107 compliant)
- JPA API
- Client/server module with memcached compatibility
- REST API
- Ability to be consumed by non-JVM platforms
- JOPR based GUI management console
- Distributed executors
    - Map/reduce programming model made easy!

Distributed Cache

- Consistent hash based data distribution
  - Will allow us to scale to bigger clusters
  - Goal of efficient scaling to 1000's of nodes
- Lightweight, "L1" cache for efficient reads
  - On writes, "L1" gets invalidated
- Dynamic rebalancing

JPA API and fine-grained replication

- Successor to POJO Cache
- JPA interface: persist, find, remove...
- Will not rely on AOP, javassist, etc.
  - More robust and easier to use/debug
- Familiar JPA interface
- Easy migration from existing, “traditional” datastores!

Management

- Uses JOPR
  - Simple WAR file
  - Rich web-based GUI
  - Open Source (LGPL)
- Infinispan exposes all data, operations in JMX
  - Infinispan-JOPR plugin represents this graphically in JOPR
  - Other plugins can be built for other tools
    - HP OpenView, Hyperic, etc.

So why is Infinispan sexy?

Why is Infinispan sexy?

- Transparent horizontal scalability
  - In both directions
- Fast, low latency data access
- Ability to address a very large heap
- Cloud-ready
- Free and doesn't suck!

JBoss WORLD CHICAGO 2009

The path ahead

presented by

What happens to JBoss Cache?

- JBoss Cache in maintenance mode
  - Currently released as 3.2.0
  - Only critical bug fixes
  - No new development
- And where is Infinispan?
  - Currently in Beta
  - Expecting a final release very soon
  - Very stable already, over 2x as fast as JBC and 4x more memory-efficient

Roadmap

- **Infinispan 4.0.0**
  - New Map API, Async API
  - Distributed cache
  - New marshalling code
  - Management tooling
- **Infinispan 4.1.0**
  - Client/server API, memcached module, language bindings
  - REST API
  - Query API

Roadmap

- **Infinispan 5.0.0**
  - JPA API
  - Fine-grained replication
- **Infinispan 5.1.0**
  - Distributed executors, map/reduce model
  - Dynamic provisioning
- **Infinispan 5.2.0**
  - Distributed querying based on map/reduce

Supported versions

- Productisation roadmap not yet finalised
- Most likely a part of
  - JBoss EAP 6
  - JBoss SOA-P 5.1
  - Possibly even JBoss EAP 5.1

To sum it up

- Clouds are becoming mainstream
  - Developers need to think about challenges involved
- Databases and clouds pose many challenges
- Data grids offer a good alternative
- Infinispan, a new open source data grid
  - Viable cloud data store
  - Also helps remove bottlenecks and single points of failure in non-cloud environments!

How can YOU participate?

- Download and try it out!
- Report bugs. Not just in code, even docs, wikis, etc.
- Suggest new features!
- Test with your own use cases
  - We love to hear how people use our stuff!!
- Lend a hand with development
  - Open and democratic dev process
  - Helps prioritize features you want!
  - Several non-Red Hat core committers already!

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/JBOSSWORLD-SURVEY

Project site:
http://www.infinispan.org

Blog:
http://blog.infinispan.org

Twitter:
http://twitter.com/infinispan
#infinispan