# Infinispan 8

Jiří Holuša
(jholusa@redhat.com)

# About Infinispan

- distributed in-memory key/value data store
- API: Map-like (ConcurrentMap), JCache (JSR-107) compliant, new Functional Map (see later)
- Library vs. Client-Server mode
- Main cache modes: Distributed, Replicated, Local
- Features:
  - Expiration, eviction
  - Transactions
  - Clustering, failover handling, data replication
  - Querying
  - Map/Reduce, distributed execution
  - Event listeners and many more ...

# News in Infinispan 8

- First based on Java 8
- New APIs leveraging the best Java 8 has to offer
  - Functional Map API
  - Distributed streams
- Querying: continuous, grouping, aggregation, Lucene 5, hybrid
- New web-based admin console
- Expiration and eviction enhancements
- New integrations: Apache Hadoop, Apache Spark

# New APIs

# Functional Map API

- Not a replacement for the current Map/JCache API
- Asynchronous
- Lazy
- Lambda based
- Example

# Distributed streams

- Distributed implementation of java.util.stream.Stream over cache data
- Topology aware
- Parallel
- Lambdas and Collectors must be serializable
- Example

# Querying

# Querying

- When you don't know the key, you have to search in the cache
- New in Infinispan 8
  - Aggregations (min, max etc.)
  - Grouping
  - Continuous queries
- Examples

# Continuous query

- Query is stored once
- Whenever starts/stops to match the query, user receives a notification to attached listener
- Listener has to override two simple methods to handle the notification
- Example

# Management console

# Web base management console

- Angular.js, bootstrap.js and PatternFly based
- Available only for server mode
- Monitoring and configuration of Infinispan cluster without requiring an external console
- Demo

# Core enhancements

# Eviction

- Ability to create a bounded cache
- We can bound it newly by memory size (estimation)
- Works for primitives key/value, byte arrays
- Since storing as byte arrays is default in server mode, requirement above is not a problem
- In library mode, Infinispan can be configured to store key/value as byte arrays
- Pluggable estimator for custom classes is planned

# Expiration

- Ability to set a lifespan of an entry
- When lifespan exceeds, the entry is automatically removed
- Newly in Infinispan , you can receive a notification about this event happening anywhere in the cluster

# Integrations

# Integrations

- Hadoop and Spark integrations
- Allows to use Infinispan cluster as datasource for Hadoop and Spark jobs
- For more information about the Spark integration (live demo and more), visit presentation “From Big Data towards Fast Data” by Vojtěch Juránek later today

# More information

- Demo: https://github.com/Holmistr/DevConfCZ2016Demo
- Website: http://www.infinispan.org
- GitHub: https://github.com/infinispan/infinispan
- JIRA: https://issues.jboss.org/browse/ISPN
- Twitter: https://twitter.com/infinispan
- IRC: #infinispan on Freenode

# Thank you