

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

Caching with Seam for Improved Application Performance

Will Dinyes

GLS Curriculum Manager, Red Hat

June 23, 2010

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Objectives

- Leverage JBoss Cache
- Understand Pre-Caching

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Hit The Cache

Need for speed

- eValanche Airlines website performance issue
- Cache the airports!

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Why Does Caching Work....

Understanding Caching

- “A store of things that will be required in the future, and can be retrieved rapidly.”

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Understanding Caching (Continued)...

- Could an application benefit from caching?
- Application Scalability Increased
- I/O focused Applications
- CPU focused Applications
- How much will caching speed an application?



Caching with Seam

Making Cache Efficient

- Persistent ORM Caching
 - EJB3
 - Java Persistence API
 - Hibernate
- Web Page Caching
- Web Page Fragment Caching

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Seam's cacheProvider

Batteries Not Included

- cacheProvider component
- Supports
 - JBoss Cache 1.x
 - JBoss Cache 2.x
 - JBoss POJO Cache 1.x (built in)
 - EHCache
- BYO JARs

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

[#summitjbw](https://twitter.com/summitjbw)

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

Caching with Seam for Improved Application Performance

Will Dinyes
GLS Curriculum Manager, Red Hat
June 23, 2010

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Objectives

- Leverage JBoss Cache
- Understand Pre-Caching

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



This session reflects a later chapter in our JB295 course, JBoss Enterprise Application Development. By now, students in the course have seen a lot of what JBoss has to offer as a JEE platform, and have entirely redesigned the application to leverage JBoss Seam. Students may not be familiar with caching at all, so there is a brief introduction to the concept.

For this session, we will cover the module which connects Seam with JBoss Cache for greatly improved access to the reservation engine's list of supported airports. We'll spend about 15-20 minutes on a lecture, then the remainder of the time on the practical lab.

Hit The Cache

Need for speed

- eValanche Airlines website performance issue
- Cache the airports!

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



The eValanche Airlines QA team (yes, believe it or not, they have one) has discovered a potential issue with the application. It seems that loading the airports on the main page is taking forever. We have been asked to speed up this process somehow. In order to do this, we have decided to leverage caching, to load airports before any of our customers actually needs to search for them.

Why Does Caching Work....

Understanding Caching

- “A store of things that will be required in the future, and can be retrieved rapidly.”

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



In terms of computer science, a cache is a collection of temporary data which either duplicates data located elsewhere (such as in the database), or is the direct result of a computation (such as

Once in the cache, the data can be repeatedly accessed inexpensively.

In most business applications, the database represents the single largest bottle neck; any enhancements used to reduce access are often the most valuable. JBoss caching of this information can alleviate this bottleneck.

Understanding Caching (Continued)...

- Could an application benefit from caching?
- Application Scalability Increased
- I/O focused Applications
- CPU focused Applications
- How much will caching speed an application?

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Remember, the emphasis for caching is aimed at improving and, where possible, enhancing performance. However, it is also very important to recognize it strategically reduces load.

Benefits for caching can largely depend on how an application is bound (example: I/O vs. CPU). With applications that are I/O focused, the time consumed to finish a computation depends on the rate at which data can be obtained. With applications that are CPU focused, the time consumed to finish a computation depends on the speed of the CPU and main system memory.

Speeding up an application with cache is influenced by many factors, but the key ones are how often a set of data can be reused, and the proportion of the response time taken in communicating with cacheable data.

Caching with Seam

Making Cache Efficient

- Persistent ORM Caching
 - EJB3
 - Java Persistence API
 - Hibernate
- Web Page Caching
- Web Page Fragment Caching

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Most ORM solutions implement a second-level cache of data from the database. This is a very powerful capability, but is often misused. It makes the most sense for data which is **shared** between many users, and is updated **rarely**.

Note: In traditional stateless architectures, people often try to use the second-level cache for conversational state. This is always bad, and is especially wrong in Seam, since Seam manages conversations for the developer in a much simpler and more efficient manner.

The Seam conversation context is itself a cache of conversational state. Components you put into the conversation context can hold and cache state relating to the **current** user interaction (as opposed to information which would be shared among users).

The Seam-managed persistence context acts as a cache of data that has been read in the current conversation. This cache tends to have a pretty high hit-rate! Seam optimizes the replication of Seam-managed persistence contexts in a clustered environment, and there is no requirement for transactional consistency with the database (optimistic locking is sufficient) so you don't need to worry too much about the performance implications of this cache, unless you read thousands of objects into a single persistence context.

Seam also lets you cache rendered fragments of a JSF page. Unlike the ORM second-level cache, this cache is not automatically invalidated when data changes, so you need to write application code to perform explicit invalidation, or set appropriate expiration policies.

Seam's cacheProvider

Batteries Not Included

- cacheProvider component
- Supports
 - JBoss Cache 1.x
 - JBoss Cache 2.x
 - JBoss POJO Cache 1.x (built in)
 - EHCACHE
- BYO JARs

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



The application can cache transactional state using the Seam `cacheProvider` component (in earlier releases, this was called `pojoCache`), which integrates JBossCache, JBoss POJO Cache or EHCACHE (depending on the container - see below) into the Seam environment. This state will be visible to other nodes if your cache supports running in a clustered mode.

Seam's built-in `cacheProvider` component manages an instance of:

- JBoss Cache 1.x (suitable for use in JBoss 4.2.x or later and other containers)
- JBoss Cache 2.x (suitable for use in JBoss 5.x and other containers)
- JBoss POJO Cache 1.x (suitable for use in JBoss 4.2.x or later and other containers)
- EHCACHE (suitable for use in any container)

To use `cacheProvider`, the developer needs to include the jars of the cache implementation in the application, as well as a `treeCache.xml` file defining the cache. No changes to `components.xml` are required!

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

[#summitjbw](https://twitter.com/summitjbw)

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT

