Software Development

Tuesday, August 21, 2012

Increase your JSF application performance (Part 1 - Environment & Configuration)

- Always use the newest MyFaces version (As you can read here: Blog Entry)
- If you just need a plain servlet container, use Tomcat instead of Jetty
- Use JUEL as EL implementation
    - Add the newest JUEL API + implementation as depedency
    - Configure MyFaces to use JUEL:

```
<context-param>
        <param-name>org.apache.myfaces.EXPRESSION_FACTORY</pa
        <param-value>de.odysseus.el.ExpressionFactoryImpl</pa
```

- Increase expression cache in JUEL
    - Create *src/main/resources/el.properties*
    - Add property *javax.el.cacheSize* with a custom size. The default size is *1000*. In my application i use a size of *3000*.
- If you use CDI, consider to use OpenWebBeans as implementation and configure this in your web.xml:

```
<context-param>
    <param-name>org.apache.myfaces.EL_RESOLVER_COMPARATOR</param-name>
    <param-value>org.apache.myfaces.el.unified.OpenWebBeansELResolverCo
```

- Enable MyFaces EL caching as described in MyFaces Wiki
- Disable JSP support in MyFaces:

```
<context-param>
    <param-name>org.apache.myfaces.SUPPORT_JSP_AND_FACES_EL</param-name
    <param-value>false</param-value>
```

- Other params to increase performance:

```
<context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Production</param-value>
</context-param>
<context-param>
    <param-name>javax.faces.FACELETS_REFRESH_PERIOD</param-name>
    <param-value>-1</param-value>
</context-param>
<context-param>
    <param-name>org.apache.myfaces.CHECK_ID_PRODUCTION_MODE</param-nam
    <param-value>false</param-value>
</context-param>
<context-param>
    <param-name>org.apache.myfaces.VIEW_UNIQUE_IDS_CACHE_ENABLED</para
    <param-value>true</param-value>
</context-param>
<context-param>
    <param-name>org.apache.myfaces.SAVE_STATE_WITH_VISIT_TREE_ON_PSS</
    <param-value>false</param-value>
```

- Configure state management as described in MyFaces Wiki
- Use a custom ServletFilter to set the correct expires/cache headers of your resources (images, stylesheets, javascripts)
- Compress and optimize your Javascripts in your build process. If you use maven, try primefaces-extensions' closure compiler maven plugin

Posted by Thomas Andraschko at 2:03 PM

+1 Recommend this on Google

Labels: CDI, JSF, JUEL, MyFaces, OpenWebBeans, OWB, Performance, Tomcat

Search

Search

Blog Archive

- ► 2013 (3)
- ▼ 2012 (7)
    - ► December (1)
    - ► October (2)
    - ► September (1)
    - ▼ August (3)
        - Get bean instance in CDI
        - Increase your JSF application performance (Part 1 ...
        - Enable failover with OpenWebBeans (> 1.1.3)

Followers

Join this site
with Google Friend Connect

Members (7)

Already a member?Sign in

Follow by Email

Submit

Labels

CDI (5)

Cluster (3)

Clustering (3)

FailOver (3)

Java (3)

Java EE (2)

Jetty (1)

JSF (5)

JUEL (1)

8 comments:

smithh032772 November 6, 2012 at 8:43 AM

Thomas, thanks for this blog. I recognized it while reading one of your posts on PrimeFaces Core forum. I have taken most of your advice when it comes to JSF/Performance, I've already migrated to MyFaces Core, JUEL, I have added myfaces context-parameters in web.xml for performance reasons, I have read up on OpenWebBeans in last 12 hours, stayed up all night last night 'trying' to migrate from Glassfish 3.1.2.2 to Apache Tomcat 7.0.32, since I wanted to use PrimeFaces Push (atmosphere/websocket), but decided to postpone (for the 4th or 5th time after failure) using PrimeFaces Push with Glassfish 3.1.2.2, AND I decided to start migrating my app to CDI, initially the RI (weld), but OpenWebBeans (CDI) is in my sights. Reading about it now in Java EE 6 tutorial, and I think Ronald (in PrimeFaces Core forum) recommended someone else's blogs as well for using/coding CDI.

Reply

Replies

smithh032772 January 4, 2013 at 9:55 AM

UPDATE: As you may already know, Thomas, some time within the last month, I completed migration from Glassfish 3.1.2.2 and JSF-managed-beans to TomEE1.5.1+ and CDI-managed-beans, and the web app is running well in production, on 6-to-10-year old hardware and software (Windows Server 2003 32-bit 4GB RAM). So, I'm using OpenWebBeans, and myself and the endusers are loving the web app.

So, thank you thank you thank you Thomas for any/all advice that you have offered!

Reply

smithh032772 February 17, 2013 at 5:17 AM

Thomas,

Of course, I am revisiting this, because the only outstanding thing I left undone was the following:

Increase expression cache in JUEL
Create src/main/resources/el.properties
Add property javax.el.cacheSize with a custom size. The default size is 1000. In my application i use a size of 3000.

1. Can you update this blog and include code-tagged example of el.properties file
2. Also, I had to re-read JUEL documentation and learned that I can place el.properties anywhere on my classpath; i didn't understand what you meant by src/main/resources/..., as that folder path does not exist in my project (using netbeans).

Reply

Thomas Andraschko February 17, 2013 at 6:02 AM

Whats the problem? just add the property file with javax.el.cacheSize as key and the amount as value.

src/main/resources will be moved to WEB-INF/classes in the WAR file.

Reply

Replies

smithh032772 February 17, 2013 at 7:03 AM

Okay. I had to refer to a properties file I saw in glassfish

Memcached (1)

MyFaces (2)

OpenWebBeans (3)

OWB (4)

Performance (4)

PrimeFaces (2)

PrimeFaces Extensions (3)

Replication (1)

Session (3)

Tomcat (4)

About Me

Thomas Andraschko

Follow 28

View my complete profile

install folder that still remains on my dev server, and then I created the el.properties file and deployed it to production and tested it, but I'm not sure how to verify (via JMX) that javax.el.cacheSize is set. I'm multi-tasking a bit right now (improving performance of a SQL query).

Reply

smartsuite March 14, 2013 at 9:29 PM

Thanks for such social platform which give us variety of idea to explore ourself technically .This exposure give benefits to everyone to fit or to survive in global market which is very essential in the global era.
Time Attendance Software In Pune

Reply

François Néron April 8, 2013 at 5:29 AM

I have clashing between javax.el version : cannot merge imports of package 'javax.el' from sources 'Import-Library 'org.apache.myfaces' version '2.1.7'(null), Import-Library 'org.springframework.spring' version '3.1.3.RELEASE'(null), Import-Bundle 'javax.el' version '2.2.0.v201108011116', Import-Bundle 'de.odysseus.juel-api' version '2.2.6'' because of conflicting values 'de.odysseus.juel-api', 'javax.el' of attribute 'bundle-symbolic-name'. Tried to use juel-spi but how to add the bundle in the manifest when you don't have a symbolic name for juel-spi?

Reply

PsyStark June 21, 2013 at 7:00 AM

Hi Thomas,
I found this interesting article while searching for a way to improve my web application performances and I was trying to follow your advices.
I'm struggling to make this implementation work, and also with a new project I'm finding some difficulties.
I'll be very grateful if you could help me or post an empty test project with the settings above that I can use as a base for some experimentation.
Thank you in advance :)

Reply

Comment as: Select profile...

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)