suraj.v.shingade@gmail.com   Dashboard   Sign Out

ADF & Weblogic How To

Sharing experience about Oracle ADF, JDeveloper, Weblogic & more.

06 November 2011

Dealing with Weblogic Stuck Threads

Check below for:

- Definition or What is a Stuck Thread?
- The problem or Why are Stuck Threads evil?
- What you can do to avoid your application completely fail?
- How to identify the problem?
- How to workaround the problem?
- Test: How to create a Stuck Thread?

Definition or What is a Stuck Thread?
WebLogic Server diagnoses a **thread as stuck** if it is *continually working (not idle) for a set period of time*.
You can tune a server's thread detection behavior by changing the length of time before a thread is diagnosed as stuck (**Stuck Thread Max Time**), and by changing the frequency with which the server checks for stuck threads. Check here to see how to change the Stuck Thread Max Time.

The problem or Why are Stuck Threads evil?
WebLogic Server automatically detects when a thread in an execute queue becomes "stuck." Because a stuck thread cannot complete its current work or accept new work, the server logs a message each time it diagnoses a stuck thread. *If all threads in an execute queue become stuck, the server changes its health state to either "warning" or "critical" depending on the execute queue:*

- If all threads in the default queue become stuck, the server changes its health state to "**critical**." (You can set up the Node Manager application to automatically shut down and restart servers in the critical health state. For more information, see "Node Manager Capabilities" in Configuring and Managing WebLogic Server.)
- If all threads in weblogic.admin.HTTP, weblogic.admin.RMI, or a user-defined execute queue become stuck, the server changes its health state to "**warning**."

So practically, a couple of Stuck Threads might not crash your server preventing it from serving request, but it is a bad sign. Usually, the number of stuck threads will increase and your server will eventually crash.

What you can do to avoid your application completely fail?
WebLogic Server checks for stuck threads periodically (this is the **Stuck Thread Timer Interval** and you can adjust it here). If all application threads are stuck, a server instance marks itself failed, if configured to do so, exits. You can configure Node Manager or a third-party high-availability solution to restart the server instance for automatic failure recovery. You can configure these actions to occur when not all threads are stuck, but the number of stuck threads have exceeded a configured threshold:Shut down the Work Manager if it has stuck threads. A Work Manager that is shut down will refuse new work and reject existing work in the queue by sending a rejection message. In a cluster, clustered clients will fail over to another cluster member.

- Shut down the application if there are stuck threads in the application. The application is shutdown by bringing it into admin mode. All Work Managers belonging to the application are shut down, and behave as described above.
- Mark the server instance as failed and shut it down it down if there are stuck threads in the server. In a cluster, clustered clients that are connected or attempting to connect will fail over to another cluster member.

The ADF How To Search

[Search]

Popular ADF How To

Dealing with Weblogic Stuck Threads

OPSS config files: jps-config.xml, jazn-data.xml and cwallet.sso

Set session timeout in Weblogic

Adding a lib / jar dependency to my project

Possible ADF Interview questions

ADF Faces Memory Scopes

ojdeploy : Deploying ADF application from Command Line

TROUBLESHOOTING: JDeveloper will not start

Deployment Profiles

TROUBLESHOOTING: Changing the source location is not allowed for a previously attempted deployment. Try deploying without specifying the source. (WLS Error upon deployment)

Author
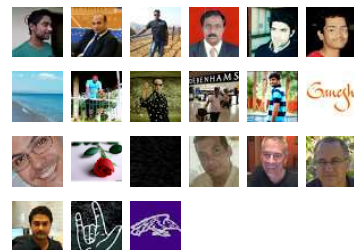
Spyros Doulgeridis

Software Engineer.
I am blogging on:
-Spyros' Log
-ADF How To

View my complete profile

Followers

Followers (50) Next

Follow

Subscribe To

Posts

How to identify the problem?

The most recommended way is to check the thread dumps. Check Sending Email Alert For Stuck Threads With Thread Dumps post of Middleware magic, to have Thread Dumps mailed to you automatically when they occur.

Tools to help you with analyzing the Thread Dumps can be:

- TDA - Thread Dump Analyzer
- Samurai

How to workaround the problem?

After you have identify the code that causes the Stuck Thread, that is the code which execution takes more than the Stack Thread Max Time, you can use Work Manager to execute your code. Work Managers have a Ignore Stuck Thread options that gives the ability to execute long running jobs. See below:

Below are some posts on how to create a Work Manager

- https://blogs.oracle.com/jamesbayer/entry/work_manager_leash_for_slow_js
- http://jdeveloperfaq.blogspot.com/2011/05/faq-34-using-weblogic-work-managers-to.html

Test: How to create a Stuck Thread?

How to create a Stuck Thread in order to test your weblogic settings? Put a breakpoint in a backing bean or model method that is called with you request. If you wait in the breakpoint for Stuck Max Thread Time, you notice a Stuck Thread trace will be shown in servers log:

```
<16 =Yí 2011 12:28:22 ΉΉ EET> <Error> <WebLogicServer> <BEA-000337> <[STUCK] ExecuteThread: '2' for
queue:
    'weblogic.kernel.Default (self-tuning)' has been busy for "134" seconds working on the
 request "weblogic.servlet.internal.ServletRequestImpl@6e6f4718[
GET /---/---/----/---/days.xhtml HTTP/1.1
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.2 (KHTML, like Gecko)
Chrome/15.0.874.120 Safari/535.2
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Cookie: JSESSIONID=DYG5TDTZSnKLTFw5CMMdLCD9sPsZS4Jqlmxj9wdGNyt1BnPcfNrR!-1520792836


]", which is more than the configured time (StuckThreadMaxTime) of "60" seconds. Stack trace:
        -------------------------------------------(--------------------.java:83)
        javax.faces.component.UIComponentBase.encodeBegin(UIComponentBase.java:823)

com.sun.faces.renderkit.html_basic.HtmlBasicRenderer.encodeRecursive(HtmlBasicRenderer.java:285)
        com.sun.faces.renderkit.html_basic.GridRenderer.renderRow(GridRenderer.java:185)
        com.sun.faces.renderkit.html_basic.GridRenderer.encodeChildren(GridRenderer.java:129)
        javax.faces.component.UIComponentBase.encodeChildren(UIComponentBase.java:848)
        org.primefaces.renderkit.CoreRenderer.renderChild(CoreRenderer.java:55)
        org.primefaces.renderkit.CoreRenderer.renderChildren(CoreRenderer.java:43)
        org.primefaces.component.fieldset.FieldsetRenderer.encodeContent(FieldsetRenderer.java:95)
        org.primefaces.component.fieldset.FieldsetRenderer.encodeMarkup(FieldsetRenderer.java:76)
```

```
org.primefaces.component.fieldset.FieldsetRenderer.encodeEnd(FieldsetRenderer.java:53)
javax.faces.component.UIComponentBase.encodeEnd(UIComponentBase.java:878)
javax.faces.component.UIComponent.encodeAll(UIComponent.java:1620)
javax.faces.render.Renderer.encodeChildren(Renderer.java:168)
javax.faces.component.UIComponentBase.encodeChildren(UIComponentBase.java:848)
org.primefaces.renderkit.CoreRenderer.renderChild(CoreRenderer.java:55)
org.primefaces.renderkit.CoreRenderer.renderChildren(CoreRenderer.java:43)
org.primefaces.component.panel.PanelRenderer.encodeContent(PanelRenderer.java:229)
org.primefaces.component.panel.PanelRenderer.encodeMarkup(PanelRenderer.java:152)
```

More digging:

- Excellent post by Frank Munz, WebLogic Stuck Threads: Creating, Understanding and Dealing with them. Updated for Weblogic 12c. Includes sample app for creating Stuck Thread too.
  - http://stackoverflow.com/questions/2709410/weblogic-stuck-thread-protection

src:

- Maxence Button excellent post: http://m-button.blogspot.com/2008/07/using-wlst-to-perform-regular.html
  - http://download.oracle.com/docs/cd/E13222_01/wls/docs81/perform/WLSTuning.html#1125714
  - http://download.oracle.com/docs/cd/E21764_01/web.1111/e13701/overload.htm
  - http://java.sys-con.com/node/358060?page=0,0

You might also like:

Enable Diagnostic Messages on you embedded or standalone ...

Run embedded weblogic in 64 bits (run it with +2gigs)

Cost of Ownership Analysis: Oracle WebLogic Server vs. JBoss

Record in a WLST script the actions performed in ...

Linkwithin

Reactions: Like (0)   nuetral (0)   wasted my time (0)

G+

Labels: weblogic

2 comments:

Anonymous February 6, 2012 at 4:38 PM

Setting the breakpoint as you say does not work.

Reply

Replies

Spyros Doulgeridis   February 6, 2012 at 7:27 PM

Hi,

It does work.

My test was performed with Eclipse. A "Remote Java Application" Debug configuration was created with the proper IP and port of my WLS. The startWeblogic.bat script were modified with the following lines:

Custom Search   Search

Pageviews last month

4,846

Translate

Select Language ▼

Powered by Google Translate

Archive

► 2012 (2)

▼ 2011 (95)

  ▼ November (2)

    Dealing with Weblogic Stuck Threads

    TROUBLESHOOTING: weblogic.descriptor.BeanAlreadyEx...

  ► October (1)

  ► September (1)

  ► August (2)

  ► July (11)

  ► June (10)

  ► May (20)

  ► April (9)

  ► March (16)

  ► February (12)

  ► January (11)

► 2010 (53)

::Mods for remote debugging in Eclipse
set PRODUCTION_MODE=false
set debugFlag=true
set DEBUG_PORT=8453

before actually start the WLS.

What is you case?

Spyros

Reply

Enter your comment...

Comment as: Unknown (Goc ▼   Sign out

Publish   Preview   ☐ Notify me

Links to this post

Create a Link

Newer Post   Home   Older Post

Subscribe to: Post Comments (Atom)

Tweet

You might also like:

There was an error in this gadget

Simple theme. Powered by Blogger.