Bugs - Bug #3102

P2J client consumes all CPU in pseudoTerminalWait across multiple threads

05/06/2016 07:26 PM - Eric Faulhaber

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Eugenie Lyzenko	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stablization for Server Features		
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

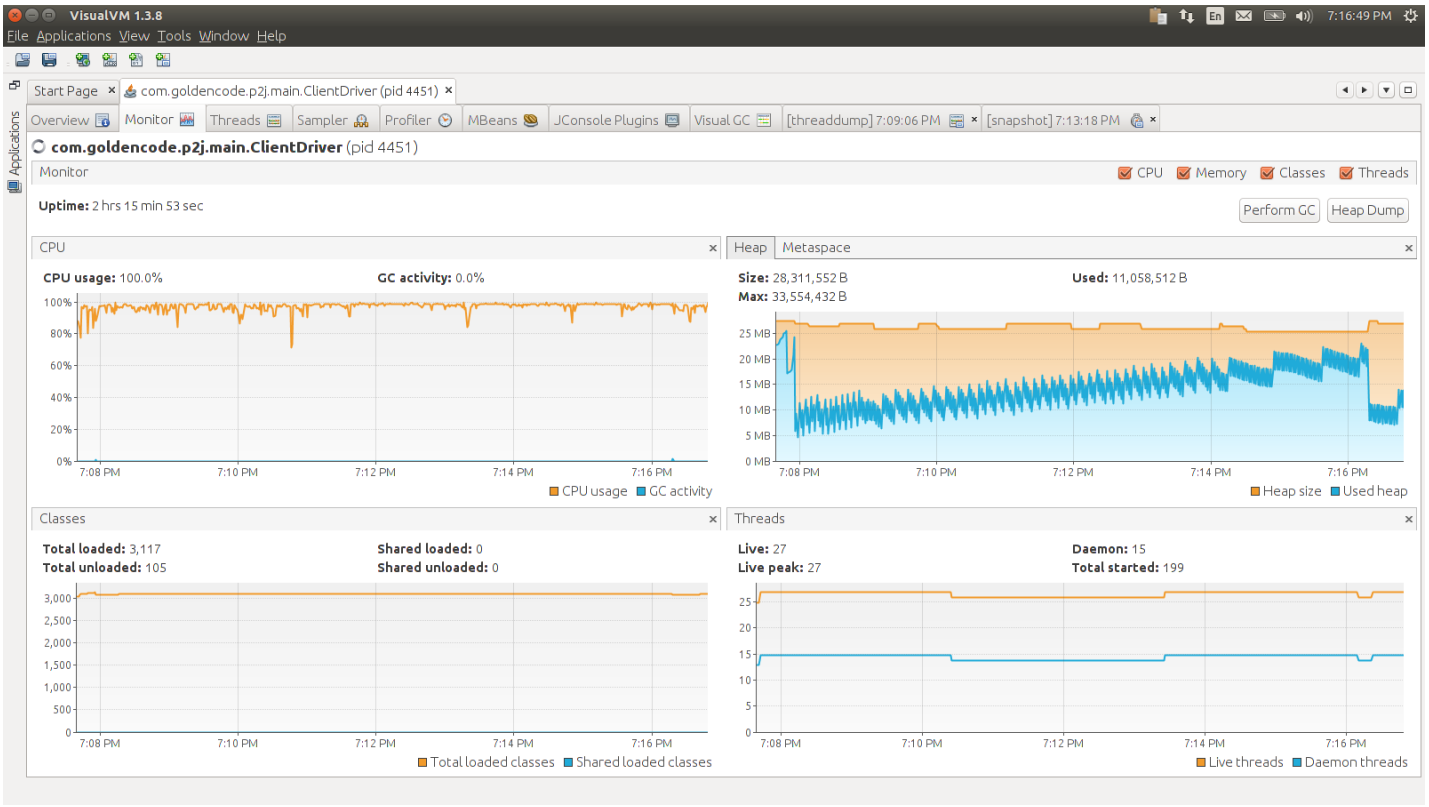
#1 - 05/06/2016 07:38 PM - Eric Faulhaber

- File threaddump-1462576146528.tdump added

- File client_activity.png added

- File pseudoTerminalWait.png added

I just ran through a batch test group in appserver mode. After the test group was finished, the server was completely idle, but one of the clients was still running flat out, using nearly 100% of my machine's CPU across all cores.



The P2J client had no child processes. I let it run for a while. It showed no signs of slowing after 10 minutes. I took a thread dump (attached) and did some CPU sampling to see where it was spending its time. It was in the native LaunchManager.pseudoTerminalWait method:

Hot Spots - Method	Self Time [%]	Self Time	Self Time (CPU)	Total Time	Total Time (CPU)
com.goldencode.p2j.util.LaunchManager.**pseudoTerminalWait[native]** ()		3,570,374 ms (84.6%)	3,570,374 ms	3,570,374 ms	3,570,374 ms
com.goldencode.p2j.ui.client.SigintWaiter.**waitForSigint[native]** ()		324,579 ms (7.7%)	324,579 ms	324,579 ms	324,579 ms
java.io.ObjectInputStream.**readObject** ()		324,579 ms (7.7%)	324,579 ms	324,579 ms	324,579 ms
java.lang.Thread.**run** ()		0.000 ms (0%)	0.000 ms	4,868,692 ms	3,894,953 ms
com.goldencode.p2j.util.LaunchManager.**pseudoTerminalWait** ()		0.000 ms (0%)	0.000 ms	3,570,374 ms	3,570,374 ms
com.goldencode.p2j.util.ProcessStream$Waiter.**run** ()		0.000 ms (0%)	0.000 ms	3,570,374 ms	3,570,374 ms
java.lang.Object.**wait** ()		1,298,317 ms (0%)	0.000 ms	1,298,317 ms	0.000 ms
com.goldencode.p2j.util.ThreadSafeQueue.**dequeue** ()		0.000 ms (0%)	0.000 ms	973,738 ms	0.000 ms
com.goldencode.p2j.net.Dispatcher.**dispatch** ()		0.000 ms (0%)	0.000 ms	649,158 ms	0.000 ms
com.goldencode.p2j.net.Dispatcher$DispatcherStub.**run** ()		0.000 ms (0%)	0.000 ms	649,158 ms	0.000 ms
com.goldencode.p2j.main.ClientCore.**start** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.net.Queue.**dequeueInbound** ()		0.000 ms (0%)	0.000 ms	649,158 ms	0.000 ms
com.goldencode.p2j.net.Conversation.**block** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.ui.client.SigintWaiter.**run** ()		0.000 ms (0%)	0.000 ms	324,579 ms	324,579 ms
com.goldencode.p2j.net.Protocol$Reader.**run** ()		0.000 ms (0%)	0.000 ms	324,579 ms	324,579 ms
com.goldencode.p2j.net.Protocol$Writer.**run** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.net.Queue.**dequeueOutbound** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.main.ClientDriver.**main** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.main.ClientDriver.**process** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.main.CommonDriver.**process** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.main.ClientDriver.**start** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.sun.proxy.$Proxy5.**standardEntry** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.net.InvocationStub.**invoke** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.net.RemoteObject$RemoteAccess.**invokeCore** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms
com.goldencode.p2j.net.HighLevelObject.**transact** ()		0.000 ms (0%)	0.000 ms	324,579 ms	0.000 ms

The client stopped when I killed the server, but was running continually until then.

#2 - 05/10/2016 06:55 PM - Eugenie Lyzenko

Eric,

1. Can you tell what command/program is launching inside child session?
2. Can the issue be reproduced permanently or only from time to time?

#3 - 05/11/2016 10:52 AM - Greg Shah

The problem is most likely with the newly added looping around the waitpid() call. I suspect we are missing an exit condition.

Eric: it is very important to know the recreate details ASAP.
Eugenie: this is your top priority as soon as you know how to recreate it.

#4 - 05/11/2016 11:10 AM - Eric Faulhaber

Eugenie Lyzenko wrote:

1. Can you tell what command/program is launching inside child session?

No, the Java client had no child processes. Based on the evidence above, I think it was looping infinitely in the native implementation of

pseudoTerminalWait.

> 2. Can the issue be reproduced permanently or only from time to time?

I've only run the batch test group which triggered this once since the #2738 fix went in. It consists of 149 batch tests (some of them long-running) and took ~2 hours to complete. I haven't run it since, so I don't know if it's repeatable or not. Unfortunately, I don't know of any other way to recreate this.

I essentially did this in #2951-18, but with the equivalent (i.e., same code, but before I merged to trunk) of P2J trunk rev. 11024.

#5 - 05/11/2016 11:18 AM - Greg Shah

> No, the Java client had no child processes.

It had one at one time otherwise we wouldn't be looping in the pseudoTerminalWait().

What is the tdump binary format? We need to see the thread dumps and know which child process was launched. Whatever it was, triggers some unexpected result where the child process exited but we don't detect it.

#6 - 05/11/2016 11:47 AM - Eric Faulhaber

Greg Shah wrote:

> > No, the Java client had no child processes.
>
> It had one at one time otherwise we wouldn't be looping in the pseudoTerminalWait().

Of course, but I meant it no longer had any child processes after the batch tests had run, at the time I noticed the P2J client consuming the CPU.

> What is the tdump binary format?

It's not binary, just a text file. tdump is just the default extension VisualVM assigned when I saved it.

#7 - 05/11/2016 12:06 PM - Eugenie Lyzenko

I think we getting into loop when res == -1(no child process exists) because we suppressed the stop/continue signals by

```
...
   signal(SIGCHLD, SIG_IGN);
...
```

in pseudoTerminalLaunch(). And in this can we could not have the proper status variable set, so loop exit conditions are not matching. I would suggest to change the loop this way adding result variable to check such a cases:

```
...
JNIEXPORT void JNICALL pseudoTerminalWait(JNIEnv* env, jclass cls, int pid, jboolean silent)
{
   pid_t child = (pid_t) pid;
   int   status;
   int rcWaitpid;

   do
   {
      rcWaitpid = waitpid(child, &status, WUNTRACED | WCONTINUED);
   }
   while (!WIFEXITED(status) && !WIFSIGNALED(status) && rcWaitpid != -1);
...
```

This will provide exit if no more child session.

#8 - 05/11/2016 12:39 PM - Greg Shah

It's not binary, just a text file. tdump is just the default extension VisualVM assigned when I saved it.

In the future, if you would please post it in Redmine as .txt, then we can view it in the browser itself.

The thread dump is weird. There are 8 threads in pseudoTerminalWait() and none that are in any clinet-side code waiting for the isAlive() notification.

```
"Thread-182" #197 prio=5 os_prio=0 tid=0x00007f5fe8891800 nid=0x257a runnable [0x00007f5fa4a40000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
```

```
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-181" #196 prio=5 os_prio=0 tid=0x00007f5fe87fe800 nid=0x2564 runnable [0x00007f5fa4b41000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-180" #195 prio=5 os_prio=0 tid=0x00007f5fe891a800 nid=0x2550 runnable [0x00007f5fa4c42000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-179" #194 prio=5 os_prio=0 tid=0x00007f5fe8834000 nid=0x250c runnable [0x00007f5fa4d43000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-178" #193 prio=5 os_prio=0 tid=0x00007f5fe80d0800 nid=0x250a runnable [0x00007f5fa4e44000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-177" #192 prio=5 os_prio=0 tid=0x00007f5fe80c6000 nid=0x2508 runnable [0x00007f5fa4f45000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-176" #191 prio=5 os_prio=0 tid=0x00007f5fe89f1800 nid=0x2505 runnable [0x00007f5fa5046000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-175" #190 prio=5 os_prio=0 tid=0x00007f5fe8802800 nid=0x2503 runnable [0x00007f5fa5147000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None
```

```
"Thread-174" #189 prio=5 os_prio=0 tid=0x00007f5fe8149800 nid=0x2501 runnable [0x00007f5fa5248000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-173" #188 prio=5 os_prio=0 tid=0x00007f5fe8b37800 nid=0x24fe runnable [0x00007f5fa5349000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None

"Thread-172" #187 prio=5 os_prio=0 tid=0x00007f5fe89ec000 nid=0x24fc runnable [0x00007f5fa585c000]
   java.lang.Thread.State: RUNNABLE
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(Native Method)
        at com.goldencode.p2j.util.LaunchManager.pseudoTerminalWait(LaunchManager.java:157)
        at com.goldencode.p2j.util.ProcessStream$Waiter.run(ProcessStream.java:1319)
        at java.lang.Thread.run(Thread.java:745)

   Locked ownable synchronizers:
        - None
```

A common idiom is to launch a child process that outputs a single line (e.g. pwd). I guess what may be happening is that the server-side does a read of a single line (e.g. one import or set) and then moves on to other code, never trying to read again. The old approach would have exited the waiting thread and no one would ever reference the isAlive(), but that is OK. In the new failure mode, the waiting thread doesn't exit (in some cases) and then these threads accumulate.

> And in this can we could not have the proper status variable set, so loop exit conditions are not matching. I would suggest to change the loop this way adding result variable to check such a cases

Eugenie: this seems reasonable, please go forward with this change.

#9 - 05/11/2016 01:42 PM - Eugenie Lyzenko

I know this is not good idea to mix several task into one. But due to urgency of this failure I can add the change into my current 3051b which has several steps completed and commit into the branch. So it can easily be tested if it works or not, then we can test 3051b and commit and then resume working on 3051 task with new 3051c branch.

Or I have to create separate 3102a branch for fixing this issue.

Let me know what do you think is better.

#10 - 05/11/2016 02:04 PM - Greg Shah

Please put it in 3102a.

#11 - 05/11/2016 02:18 PM - Eugenie Lyzenko

Created task branch 3102a from trunk revision 11025.

#12 - 05/11/2016 02:40 PM - Eugenie Lyzenko

Task branch 3102a for review updated to revision 11026.

The change is from note #7. Eric, is it possible to test it against the issue?

#13 - 05/11/2016 02:57 PM - Eric Faulhaber

Eugenie Lyzenko wrote:

> The change is from note #7. Eric, is it possible to test it against the issue?

I don't know whether I can get to this today, but I will test it. Thanks.

#14 - 05/11/2016 04:05 PM - Greg Shah

Code Review Task Branch 3102a Revision 11026

I'm fine with the changes.

While Eric is testing this on the ETF, please go ahead and test this in MAJIC runtime regression testing.

#15 - 05/11/2016 04:07 PM - Eugenie Lyzenko

Greg Shah wrote:

> While Eric is testing this on the ETF, please go ahead and test this in MAJIC runtime regression testing.

OK.

#16 - 05/12/2016 11:46 AM - Eugenie Lyzenko

The main part completed without regressions. Running the CTRL-C part.

#17 - 05/12/2016 02:50 PM - Eugenie Lyzenko

Testing completed, no regressions. The results - 3102a_11026_6c4b2ca_20160512_evl.zip.

#18 - 05/12/2016 04:41 PM - Greg Shah

Eric reports that the ETF batch tests (where this problem occurred) have successfully executed and the problem is gone.

Please merge 3102a to trunk.

#19 - 05/12/2016 04:53 PM - Eugenie Lyzenko

Branch 3102a was merged to trunk as revno 11026 then it was archived.

#20 - 05/12/2016 04:55 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Closed

- Assignee set to Eugenie Lyzenko

#21 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features

Files

client_activity.png	133 KB	05/06/2016	Eric Faulhaber
pseudoTerminalWait.png	110 KB	05/06/2016	Eric Faulhaber
threaddump-1462576146528.tdump	20.9 KB	05/06/2016	Eric Faulhaber