Runtime Infrastructure - Support #5354

usage of GraalVM for better performance

05/07/2021 10:43 AM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 05/07/2021 10:44 AM - Greg Shah

From Eugenie:

> Thinking about the ways to speed up the <large_gui_application> I've found the option previously declined. What do you think about using GraalVM EE(https://www.graalvm.org) as runtime environment?
>
> The advantage in 25-30% of performance gain comparing to OpenJDK(I have measured). The disadvantage: it is free only for evaluation and development. Is it a stopper?

#2 - 05/07/2021 10:48 AM - Greg Shah

> Is it a stopper?

For runtime usage, customers would need to obtain a license/subscription. Oracle has added GraalVM to its Java SE subscription. This means it would cost the customer $25/processor core each month. On a server with 8 cores, that is $200/month extra. That is a pretty substantial amount BUT some customers might choose it if it really makes a 25%-30% difference.

We would never be able to REQUIRE its usage. That means that OpenJDK will need to be fully functional for FWD. Offering it as an optional choice is OK.

#3 - 05/07/2021 10:49 AM - Greg Shah

Did you test AOT (ahead of time) compilation? Or was your result purely based on swappng out the runtime?

#4 - 05/07/2021 10:52 AM - Greg Shah

Eric: It is worth having you try this out. If a different JDK can provide 25% improvement, then we should know it and offer it to customers.

I think the AOT support is worth trying as well. I would expect:

- It would greatly speed up any scenarios where there is normally a warm up.
- I would hope we could force everything to be pre-compiled with maximum optimizations. This would potentially compile things that would not be compiled in normal runtime cases.
- The reduced Hotspot compilation will buys back some CPU cycles and reduce memory usage.

#5 - 05/07/2021 11:05 AM - Eugenie Lyzenko

Greg Shah wrote:

> Did you test AOT (ahead of time) compilation? Or was your result purely based on swappng out the runtime?

Just swapped to new JDK and back. I have not tested AOT. I'm not event sure if we can transform very large and complex Java project into native image executable. Getting several hundred MB exe file...

#6 - 05/07/2021 11:12 AM - Igor Skornyakov

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Did you test AOT (ahead of time) compilation? Or was your result purely based on swappng out the runtime?
>
> Just swapped to new JDK and back. I have not tested AOT. I'm not event sure if we can transform very large and complex Java project into native image executable. Getting several hundred MB exe file...

I cannot imagine how it is possible to convert to a native form an application that heavily uses AspectJ, Reflection API, and even class generation at runtime.

#7 - 05/07/2021 11:47 AM - Greg Shah

> I cannot imagine how it is possible to convert to a native form an application that heavily uses AspectJ, Reflection API, and even class generation at runtime.

AOT compilation is not making these into standalone native programs. There is still a virtual machine packaged with the compiled classes. The compiled classes are no different from JIT compiled classes. I don't see any issue with reflection or class generation (these are still generated and loaded at runtime by our code).

I am less sure if there are any issues with our load-time weaving in AspectJ. Any compile-time weaving should work fine.

#8 - 05/10/2021 01:01 PM - Eugenie Lyzenko

I've made some additional testing and benchmarking for Java versions. Especially comparing AdoptOpenJDK I used before with GraalVM different editions.

The result is interesting I think. Tested GraalVM Community Edition (absolutely free at this time for all purposes).

GraalVM CE shows 3000 ms vs 5400 ms in renaissance test case on my system. Then I have tested "large GUI application" with GraalVM CE on my local bare metal case. The opening time for "basic GUI application screen" is about 1.5 seconds in Chrome (vs 2.1+ seconds with AdoptOpenJDK). I think this is a good advantage for free we can(and I guess need) to use.

I would like to re-configure testing system to use GraalVM CE to see what speed up we can have here if you do not mind. I need to stop the server, install new Java, re-compile FWD and start server again. Please let me know if I can do it now or choose another time frame.

And a bit of side effects (not a stoppers I think).
1. From my imagination the start up time is now increased a bit with GraalVM. I guess this is from more aggressive JIT compilation at starting time.
2. Some system variables should be modified to keep compatibility for GraalVM with our current build environment to be able to recompile FWD:

```
export ANT_OPTS='-Dpolyglot.js.nashorn-compat=true'
export GRADLE_OPTS='-Dpolyglot.js.nashorn-compat=true'
```

#9 - 05/10/2021 01:23 PM - Greg Shah

Eugenie: Please do whatever setup is needed in the VM so that you can easily switch between OpenJDK and GraalVM. Once that is ready, Eric will test the system using OpenJDK, you will switch to GraalVM and Eric will test again. Does that make sense?

#10 - 05/10/2021 01:28 PM - Eugenie Lyzenko

Greg Shah wrote:

> Eugenie: Please do whatever setup is needed in the VM so that you can easily switch between OpenJDK and GraalVM. Once that is ready, Eric will test the system using OpenJDK, you will switch to GraalVM and Eric will test again. Does that make sense?

Yes. This way we can easily see the difference.

#11 - 05/10/2021 01:43 PM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Eugenie: Please do whatever setup is needed in the VM so that you can easily switch between OpenJDK and GraalVM. Once that is ready, Eric will test the system using OpenJDK, you will switch to GraalVM and Eric will test again. Does that make sense?
>
> Yes. This way we can easily see the difference.

It is ready for fast switch (5-10 min including server restart).

#12 - 05/10/2021 02:09 PM - Eric Faulhaber

Eugenie Lyzenko wrote:

> It is ready for fast switch (5-10 min including server restart).

OK, let's do a blind test. Please set it to either OpenJDK or GraalVM, but don't tell me which it is.

Let me know when it is ready and I will test and post my results. Then you can switch it to the other and I will test again. Thanks.

#13 - 05/10/2021 02:19 PM - Eugenie Lyzenko

Eric Faulhaber wrote:

> Eugenie Lyzenko wrote:
>
>> It is ready for fast switch (5-10 min including server restart).
>
> OK, let's do a blind test. Please set it to either OpenJDK or GraalVM, but don't tell me which it is.
>
> Let me know when it is ready and I will test and post my results. Then you can switch it to the other and I will test again. Thanks.

OK. The case 1 is ready to test.

#14 - 05/10/2021 02:59 PM - Eric Faulhaber

Case 1 Results:

Using Chrome (same set up as before, no config changes), I logged on and opened every screen from the target screen's menu group, once each. Then I opened the target screen many times, until the timing was fairly consistent for 5 openings. I was just using the system clock and eyeballs to measure, so granularity of measurement was not exact. The screen consistently took around 3 seconds to open.

Please let me know when you have switched to the other JVM and I'll try again.

#15 - 05/10/2021 03:07 PM - Eugenie Lyzenko

Eric Faulhaber wrote:

> Case 1 Results:
>
> Using Chrome (same set up as before, no config changes), I logged on and opened every screen from the target screen's menu group, once each. Then I opened the target screen many times, until the timing was fairly consistent for 5 openings. I was just using the system clock and eyeballs to measure, so granularity of measurement was not exact. The screen consistently took around 3 seconds to open.
>
> Please let me know when you have switched to the other JVM and I'll try again.

The case 2 is ready to test.

#16 - 05/10/2021 03:24 PM - Eric Faulhaber

Case 2 Results:

Same browser, same test procedure. I didn't notice any dramatic difference with this VM. After warm-up, the target screen took around 3 seconds to open, on some occasions it seemed just slightly faster than 3 seconds.

#17 - 05/10/2021 03:31 PM - Eugenie Lyzenko

Well, a kind of surprise for me.

Case 1 - GraalVM CE 21.1.0
Case 2 - AdoptOpenJDK 1.8.0_292-b10

#18 - 05/10/2021 03:47 PM - Eric Faulhaber

That is a bit surprising. Which VM were we using when I first tested (May 5)?

#19 - 05/10/2021 03:48 PM - Eugenie Lyzenko

Eric Faulhaber wrote:

> That is a bit surprising. Which VM were we using when I first tested (May 5)?

AdoptOpenJDK 1.8.0_292-b10.

#20 - 05/10/2021 04:10 PM - Greg Shah

The enterprise edition is supposed to have some extra optimizations. We also have not yet tested the benefits of AOT compilation.

How easy is it to test the enterprise edition?

#21 - 05/10/2021 05:08 PM - Eugenie Lyzenko

Greg Shah wrote:

> The enterprise edition is supposed to have some extra optimizations. We also have not yet tested the benefits of AOT compilation.
>
> How easy is it to test the enterprise edition?

Not difficult. I need to:
1. Upload and unpack .tar.gz
2. Set up additional environment option:

```
_JAVA_OPTIONS="$_JAVA_OPTIONS -Dgraal.CompilerConfiguration=enterprise"
```

3. Restart the server.

Actually the same time as for GraalVM CE

#22 - 05/10/2021 05:29 PM - Eugenie Lyzenko

Greg Shah wrote:

> The enterprise edition is supposed to have some extra optimizations. We also have not yet tested the benefits of AOT compilation.
>
> How easy is it to test the enterprise edition?

Do we need this setup to be done?

#23 - 05/11/2021 07:38 AM - Greg Shah

Yes, please do.

#24 - 05/11/2021 07:56 AM - Eugenie Lyzenko

Greg Shah wrote:

> Yes, please do.

OK. I'm stopping the server for set it up.

#25 - 05/11/2021 08:26 AM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Yes, please do.
>
> OK. I'm stopping the server for set it up.

Done. My timing is 2.1 sec for the best case with warmed JVM.

The time measurement approach I use (this can slightly affect the result may be):
1. Using https://sekundomer.net on separate system.
2. The time started **just after** click mouse on screen initiation.
3. I'm trying to **match** time stop with the moment I see the screen. When done - read the elapsed time.

#26 - 05/11/2021 08:44 AM - Greg Shah

> 2. The time started just after click mouse on screen initiation.

Why "just after"? This will introduce a bias toward reporting the screen load is faster than it really is.

#27 - 05/11/2021 08:57 AM - Eugenie Lyzenko

Greg Shah wrote:

>> 2. The time started just after click mouse on screen initiation.
>
> Why "just after"? This will introduce a bias toward reporting the screen load is faster than it really is.

I'm trying to catch the moment when the mouse UP event occurs (actually next click event but the difference can be ignored I guess). Because DOWN event itself does not start the screen show (try to press and hold mouse in DOWN state to see). And I'm adding 0.1 second to final time to compensate distance between mouse UP and timer start (my estimation for how fast I can press the button starting timer).

#28 - 05/11/2021 09:00 AM - Greg Shah

The customer is probably testing based on the DOWN event since the user's perception is that at that moment they have told it to execute. Try to switch to that approach and see what the measurement is as a result. The bottom line here is that their perception is the important thing here, even if you are "technically correct".

#29 - 05/11/2021 10:01 AM - Eugenie Lyzenko

Greg Shah wrote:

> The customer is probably testing based on the DOWN event since the user's perception is that at that moment they have told it to execute. Try to switch to that approach and see what the measurement is as a result. The bottom line here is that their perception is the important thing here, even if you are "technically correct".

Mouse DOWN approach (latest Chrome used):
1.6 sec - my bare metal system
2.4-2.5 sec - remote system

#30 - 05/11/2021 10:08 AM - Constantin Asofiei

Eugenie: in your 'bare metal system' case, do you run the web client there, too?

Greg: for the remote system, as this is accessed over the internet, I wonder how much impact the network transfer has.

#31 - 05/11/2021 10:12 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

> Eugenie: in your 'bare metal system' case, do you run the web client there, too?

Yes, the Swing client timing is less almost in a twice factor (< 1.0 sec).

#32 - 05/11/2021 10:27 AM - Greg Shah

> Greg: for the remote system, as this is accessed over the internet, I wonder how much impact the network transfer has.

Yes, I wonder if this VM is in a European AWS data center. I was suspecting that Eric and myself probably have "round trip crossings of the Atlantic" for every packet. So his 3 second result may have some amount of unavoidable latency.

#33 - 05/11/2021 11:18 AM - Eugenie Lyzenko

Constantin,

What timing do you have for sample screen opening with remote system?

I still think the result is significantly depending on client environment. For example accessing from pure Windows is much slower comparing to Ubuntu even for Chrome browser (more than 3 sec for my case).

#34 - 05/11/2021 02:04 PM - Constantin Asofiei

The screen is opening in ~3.2 seconds on Chrome Version 90.0.4430.85 (Official Build) unknown (64-bit).

I haven't done anything special to my Chrome.

#35 - 05/11/2021 02:14 PM - Eugenie Lyzenko

Constantin Asofiei wrote:

> The screen is opening in ~3.2 seconds on Chrome Version 90.0.4430.85 (Official Build) unknown (64-bit).

> I haven't done anything special to my Chrome.

Do you use hardware acceleration option in Chrome browser?

#36 - 05/11/2021 02:59 PM - Eugenie Lyzenko

Constantin,

Do you use discrete GPU or integrated in CPU?

Can you please publish chrome://gpu content? Here or in separate mail.

#37 - 05/11/2021 03:43 PM - Constantin Asofiei

- File gpu.html added

See attached.

#38 - 05/19/2021 10:10 AM - Greg Shah

Eric: Did you ever test the Enterprise version of GraalVM?

Files

gpu.html	107 KB	05/11/2021	Constantin Asofiei