

How can I speed up my maven2 build?

Asked 16 years, 2 months ago Modified 5 years, 10 months ago

Viewed 32k times



35



I'm using a local artifactory to proxy the request, but the build and test phases are still a bit slow. It's not the actual compile and tests that are slow, it's the "warmup" of the maven2 framework. Any ideas?

maven-2

Share

Improve this question

Follow

edited Aug 5, 2014 at 22:21



Kara

6,206 ● 16 ● 53 ● 58

asked Oct 2, 2008 at 10:37



s3v1

2,933 ● 2 ● 33 ● 25

2 If you find a good answer out-of-band (f.ex. on other sites), it would be very appreciated if you posted them here as an answer. – [Einar](#) Oct 2, 2008 at 13:29

I found [this article](#) helpful. Limiting internet access works well for me. – [artronics](#) May 1, 2016 at 19:04 ✎

8 Answers

Sorted by:

Highest score (default)



24

There are some possibilities to optimize some of the build tasks. For example the 'clean' task can be optimized from minutes to just milliseconds using simple trick - rename 'target' folder instead of delete.



To get details how to do it refer to [Speed up Maven build](#).



Share Improve this answer

answered Feb 19, 2009 at 9:15

Follow



Roman



neat tricks, I'm often wasting undue time on the clean task too. – [s3v1](#) Dec 3, 2009 at 13:25

2 I tackled this issue for a while and took the time to write a blog post about it... hope it helps others!
urbancat.org/2012/05/how-to-speed-maven-up.html – Nadav Jun 2, 2012 at 15:21

1 @nadav the link is not working anymore. Could you correct it? – [Gábor Lipták](#) Aug 20, 2013 at 11:30



9

I don't know what version of Maven you are using, I assume 2, but I will give what I use for Maven 1.x to speed up and make things build a tiny bit quicker.



These will fork the junit tests into a new process (also helps when you use environment variables in tests etc and gives the tests a little more memory.





```
-Dmaven.junit.fork=true  
-Dmaven.junit.jvmargs=-Xmx512m
```

This forks the compilation which might speed things up for you

```
-Dmaven.compile.fork=true
```

I hope this can help a little, try it out.

Also refer to [get more speed with your maven2 build](#).

Share Improve this answer

Follow

edited Aug 5, 2014 at 22:18



Kara

6,206 ● 16 ● 53 ● 58

answered Oct 2, 2008 at 10:42



Henry B

8,047 ● 11 ● 44 ● 47

I'm using maven2 but it's a nice link and I'll definitely try that patched version for my downloads. However the issue was more build and test startup times... – [s3v1](#) Oct 2, 2008 at 10:56

Enabling fork made my project slightly faster – [Gábor Lipták](#) Aug 20, 2013 at 11:59



If you are using Maven3 (`$ mvn -version`), you can also follow this [guide](#). In my case, the results are:



```
$ mvn clean install
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 03:05 min
[INFO] Finished at: 2015-07-15T11:47:02+02:00
[INFO] Final Memory: 88M/384M
```

With Parallel Processing (4 threads):

```
$ mvn -T 4 clean install
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:22 min (Wall Clock)
[INFO] Finished at: 2015-07-15T11:50:57+02:00
[INFO] Final Memory: 80M/533M
```

Parallel Processing (2 threads per core)

```
$ mvn -T 2C clean install

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:12 min (Wall Clock)
[INFO] Finished at: 2015-07-15T12:00:29+02:00
[INFO] Final Memory: 87M/519M
[INFO] -----
```

As we can see, the difference it's almost a minute, near 20-30% of speed improvement.

Follow

answered Jul 15, 2015 at 9:56



facundofarias

3,043 ● 32 ● 27



4



1. Adjust memory configurations to optimum for eg: add this line to mvn.bat set MAVEN_OPTS=-Xmx512m -XX:MaxPermSize=256m
2. Clean phase of mvn normally deletes target folder. Instead if we are renaming target folder the cleaning phase will be much faster.<[quickClean](#)>
3. -Dmaven.test.skip=true will skip the test execution.
4. Add -Denforcer.skip=true to mvn command line argument (This is enforcing versions of maven, jdk etc ,we can skip it after initial runs)
5. Disable non-critical operations during build phase: Analysis, javadoc generation, source packaging. This will save huge time.
6. Spawnig new process also helps in time improvement -Dmaven.junit.fork=true (fork the junit tests into a new process) -Dmaven.compile.fork=true (forks the compilation)

Hope it helps.

Share Improve this answer

answered Sep 20, 2013 at 8:10

Follow



Sinto

930 ● 5 ● 10



You can use `-DskipTests=true` to skip unit tests. which would speed up builds

3

Share Improve this answer

answered Dec 28, 2011 at 20:41



Follow



user1059176



1 So would not doing any tests in the first place..? – [Rob](#) Sep 10, 2012 at 20:51

The CI robot will do all these when building, so they need to be there. – [Thorbjørn Ravn Andersen](#) Feb 7, 2013 at 10:30

-o indicating offline mode helps as it does not look at remote updates for snapshots – [Abs](#) May 1, 2015 at 7:20



2

I've found that parsing reactor projects is significantly slower than single-pom projects. If your build is reactor (multi-module) and your developers are not working on all modules at the same time, you can remove the parent POM and build them separately, resolving the dependencies using the local repo. The disadvantage is that you need to install or deploy a module in order for its dependents to see the changes.



Also, you might want to look at the new Maven 2.1 M1 which contains some significant speed improvements.

If none of these helps, post more details about your project configuration (modules structure and plugins), command line parameters and hardware config (memory and disk). Running Maven with -X might also show where is it taking its time.

Share Improve this answer

edited Oct 25, 2008 at 12:34

Follow

community wiki
[ddimitrov](#)

it is indeed a reactor project that I'm using. three small modules and a webproject. I tried that 2.1 build and it did seem a little faster. – [s3v1](#) Oct 3, 2008 at 10:23



I'd use locally installed Nexus.

1

Share Improve this answer

answered Feb 19, 2009 at 19:11

Follow



[Sergey Aldoukhov](#)

22.7k ● 18 ● 74 ● 104



Nexus is good too. I've used both nexus and artifactory, and they're both pretty fast. Nexus is a bit easier to configure IMO

– [s3v1](#) Dec 3, 2009 at 13:28

What does Nexus do that would help this process? – [snowe](#)
Jan 10, 2014 at 20:11



0

Initially, you should get a finer analysis on your build times using something like [this](#) and identify the candidates that are taking the most time.



Are tests spinning up a H2 database per test? Is the download of external jar files taking the time? This will guide where to focus your investigation. Just applying go-fast flags don't usually work as they would have already been included by default, and you don't want to be sacrificing your tests with skip flags.



Share Improve this answer

Follow

answered May 9, 2016 at 8:49



[Tom Chamberlain](#)

3,065 ● 23 ● 24
