Title: Multithreading during image build

Post Body:

I'm building a custom image that contains my java app. Basically I'm calling `mvn clean install` inside the `Dockerfile`. I'm using this `RUN mvn -T 1C clean install` in order to speed up the process.

This will use 1 thread per available CPU core. Will this be enough? How does mvn calculate the available cpu cores? Do I have to do something in order to tell it to check the host's CPUs and not the VM's? I'm using Docker for Mac (which internally uses xhyve as far as I know).

Update:

I'm building one module (fat jar, spring boot application with tomcat embedded), maven 3.3.9, about 30 mins of total duration without the thread boost. The huge overhead is that it downloads dependencies from my private repo in S3.

Accepted Answer:

According to my humble opinion I think that build application by maven inside Dockerfile get more issues than benefits.

When you run new build maven cache is empty so your build spend many time to download all dependencis.

Changing threds used by build do not give you any performance boost.

You can try prepare one docker image with maven and try run some maven command to populate maven cache in this image build. Next you use this image as base for your project.

Another solution maybe separate download dependency and build in two RUN, thanks docker cache your next build may be faster.

```
RUN mvn dependency:go-offline RUN mvn clean install
```

Highest Rated Answer: None