解决 MacOS 下 getByInetAddress 导致 tomcat 启动项目很慢的问题

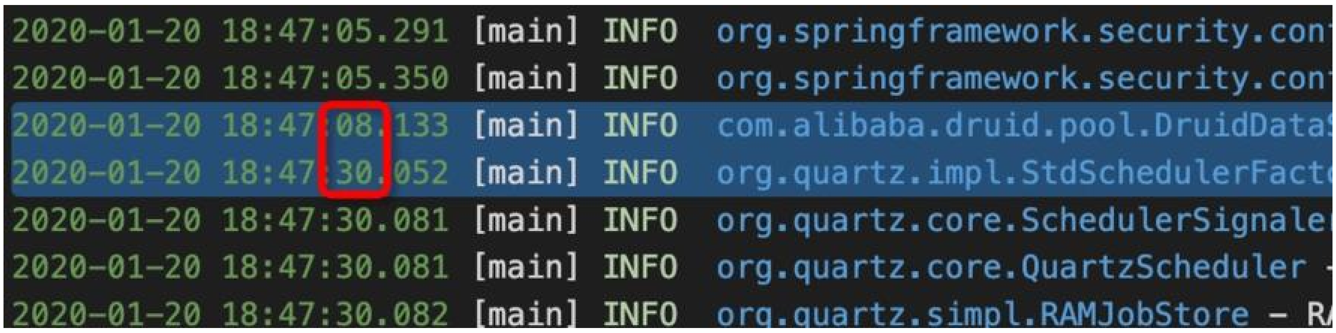
作者：cloudlang

原文链接：https://ld246.com/article/1579605454929

来源网站：链滴

表象：tomcat 启动项目很慢，比linux下启动至少慢20-30秒以上

开始的时候没有换系统测试，一直以为是项目配置或者jar文件版本冲突引起的问题。



```
2020-01-20 18:47:05.291 [main] INFO  org.springframework.security.con
2020-01-20 18:47:05.350 [main] INFO  org.springframework.security.con
2020-01-20 18:47:08.133 [main] INFO  com.alibaba.druid.pool.DruidDataS
2020-01-20 18:47:30.052 [main] INFO  org.quartz.impl.StdSchedulerFact
2020-01-20 18:47:30.081 [main] INFO  org.quartz.core.SchedulerSignale
2020-01-20 18:47:30.081 [main] INFO  org.quartz.core.QuartzScheduler -
2020-01-20 18:47:30.082 [main] INFO  org.quartz.simpl.RAMJobStore - R
```

通过不停的断点，debug，分析日志，跟踪了一两个小时，最后落在最关键的一行代码上

java.netNetworkInterface.getByInetAddress(InetAddress addr)

只要调用了这个方法，就会延迟5秒钟左右。
n次调用，那么就是5*n秒，可以想象每次启动项目的绝望。

baidu，google都发现有不少类似的情况，但是都没有给出更深入解析。

解决办法也大致相同：

1. 就是把本机的hostname 添加到 hosts 中，例如:

```
127.0.0.1 localhost
127.0.0.1 adeMacBook-Pro.local
255.255.255.255 broadcasthost
::1 localhost
::1 adeMacBook-Pro.local
```

2. 执行 scutil --set HostName "localhost"

参考：

1. DnsNameResolver hangs for 5 seconds in InetAddress.getAllByName0 on Mac OSX
2. Jvm takes a long time to resolve ip-address for localhost
3. mac系统InetAddress.getLocalHost().getHostAddress() 很慢
4. Mac上java应用（例如spring boot）启动慢的原因之一

终于在这篇2012年的博客里面找到引起此次问题的原因

博文： ipv6造成的死锁问题 https://www.iteye.com/blog/jiajianchao-gmail-com-1597253

Description
For sites using only IPv4, you may find better performance and simplicity in configuring the e
vironment to use only IPv4, wherever possible. This wiki page provides guidance on forcing a

plications to use IPv4 when possible.
Many OSs will enable IPv6 by default, even if the environment is only using IPv4. Configuring
he OS to disable IPv6 would help to prevent these sorts of problems.
Known Problems
There are some known problems with applications trying to use IPv6, when the environment i
 really only configured for IPv4.
1. IPv6 changes the way that Round-Robin A-records are used, and instead forces an ordered
priority weighting of A-records. This can cause high-availability techniques dependent on rou
d-robin A-records to fail or work incorrectly. If using IPv4/IPv6 dual-stack, then the JVM will u
e getaddrinfo() instead of gethostbyname(). getaddrinfo() will sort the DNS results in an orde
ed way, due to RFC 3484 [1]. Some OSs provide /etc/gai.conf configuration options in an att
mpt to configure record handling.
2. The JVM can be blocked by an infinite loop problem between the JVM and libc when using
he IPv6 getaddrinfo() libraries with IPv4 addresses. This issue is discussed in more detail here:

http://bugzilla.zimbra.com/show_bug.cgi?id=68432
http://old.nabble.com/-Bug-libc-12926--New%3A-getaddrinfo%28%29-make_request%28%2
-may-spin-forever-td31913044.html
http://sourceware.org/bugzilla/show_bug.cgi?id=12926
In the JVM, following is what the threads look like. You will see many threads in the thread d
mp file locked in a state similar to this:

```
"btpool0-41529" prio=10 tid=0x00002aaac45dd000 nid=0x7db9 in Object.wait() [0x0000000
47155000]
   java.lang.Thread.State: WAITING (on object monitor)
        at java.lang.Object.wait(Native Method)
        at java.lang.Object.wait(Object.java:485)
        at java.net.InetAddress.checkLookupTable(InetAddress.java:1267)
        - locked <0x00000006e385b000> (a java.util.HashMap)
        at java.net.InetAddress.getAddressFromNameService(InetAddress.java:1190)
```

These threads are waiting for the Lookup Table object to be released from another thread tha
 holds it. Here's the thread holding it:

```
"btpool0-41526" prio=10 tid=0x00002aaac5904800 nid=0x7db6 runnable [0x0000000044c02
00]
   java.lang.Thread.State: RUNNABLE
        at java.net.Inet6AddressImpl.lookupAllHostAddr(Native Method)
        at java.net.InetAddress$1.lookupAllHostAddr(InetAddress.java:850)
        at java.net.InetAddress.getAddressFromNameService(InetAddress.java:1201)
        at java.net.InetAddress.getAllByName0(InetAddress.java:1154)
        at java.net.InetAddress.getAllByName(InetAddress.java:1084)
        at java.net.InetAddress.getAllByName(InetAddress.java:1020)
        at java.net.InetAddress.getByName(InetAddress.java:970)
```

You'll see that's running Inet6AddressImpl.lookupAllHostAddr. Because of a bug between Jav
 and libc, this lookup can enter an infinite loop when a certain race condition occurs. This occ
rs infrequently, but can cause deadlocks where all threads of one type (such as LMTP threads)
or even all JVM threads can end up blocked.
With java.net.preferIPv4Stack set to true, Java will not execute this code and the problem shou
d be avoided.
Configuration
1. Java processes can be configured to prefer the IPv4 stack. The default is to prefer the IPv6 s
ack, so it requires a specified JVM argument to prefer IPv4:
-Djava.net.preferIPv4Stack=true
This would need to be added to your existing mailboxd_java_options. Your existing configurat
on may vary depending on your performance tuning [see http://wiki.zimbra.com/wiki/Perfor
ance_Tuning_Guidelines_for_Large_Deployments], so be careful to append this option to wha

ever is there currently:

```
# su - zimbra
$ zmlocalconfig mailboxd_java_options
$ zmlocalconfig -e mailboxd_java_options="-server -Djava.awt.headless=true -Dsun.net.ineta
dr.ttl=60 -XX:+UseConcMarkSweepGC -XX:NewRatio=2 -XX:PermSize=192m -XX:MaxPermSi
e=192m -XX:SoftRefLRUPolicyMSPerMB=1 -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCTim
Stamps -XX:+PrintGCApplicationStoppedTime -XX:+HeapDumpOnOutOfMemoryError -XX:H
apDumpPath=/opt/zimbra/log -Djava.net.preferIPv4Stack=true"
```

For more reference background on this, please review here:
http://bugzilla.zimbra.com/show_bug.cgi?id=13161#c55

2. Configuring the OS to disable IPv6

Each OS may have unique recommendations for disabling IPv6. This article does not currently nclude all OS-level recommendations, but please do a web search and determine methods for disabling the IPv6 interfaces, modules, and stack for your OS of choice.

原文链接：解决 MacOS 下 getByInetAddress 导致 tomcat 启动项目很慢的问题