Subject: Ubuntu 12.04 templates and java based applications
Posted by adamjseed on Wed, 20 Jun 2012 09:22:30 GMT

Hi all,

I have recently changed to the new ubuntu 12.04 template but im running into some issues.

Let me take one example: Serviio is a upnp server based on java. serviio.org

I have download, extracted and ran serviio.sh on my ubuntu 12.04 container but it doesn't start. I have tried both apt-get default-jre and doing a full oracle java install. from what I can tell there is are java error preventing it from starting.

It works fine on my Debian 6 container and on a full ubuntu 12.04 vm. Am I missing something?

Thanks
Adam

Subject: Re: Ubuntu 12.04 templates and java based applications
Posted by jac2703 on Thu, 28 Jun 2012 16:14:03 GMT

Hi,

I am also having issues with the 12.04 container.

I am trying to run a java application after installing apt-get install default-jre however I get an error which I have posted below. It looks to be anything Java related fails as I have tried a few other java applications.

Container spec:
1 core i7
1 GB RAM
1 GG SWAP
5 GB HD - mostly empty.

Has anyone else had any trouble with this template?

```
#
# A fatal error has been detected by the Java Runtime Environment:
#
#  SIGSEGV (0xb) at pc=0x00007f62e995dcc0, pid=630, tid=140062169667328
#
# JRE version: 6.0_24-b24
```

Stack: [0x00007f62c3ce9000,0x00007f62c3dea000], sp=0x00007f62c3de7768, free space=1017k

Native frames: (J=compiled Java code, j=interpreted, Vv=VM code, C=native code)
V [libjvm.so+0x3afcc0] constMethodOopDesc::object_size(int, int, int, int)+0x60
V [libjvm.so+0x34d02a] ClassFileParser::parse_methods(constantPoolHandle, bool, AccessFlags*, bool*, objArrayOopDesc**, objArrayOopDesc**, objArrayOopDesc**, Thread*)+0x3ea
V [libjvm.so+0x350898] ClassFileParser::parseClassFile(symbolHandle, Handle, Handle, KlassHandle, GrowableArray<Handle>*, symbolHandle&, bool, Thread*)+0x8c8
V [libjvm.so+0x355773] ClassLoader::load_classfile(symbolHandle, Thread*)+0x1f3
V [libjvm.so+0x71b854] SystemDictionary::load_instance_class(symbolHandle, Handle, Thread*)+0x354
V [libjvm.so+0x719f60] SystemDictionary::resolve_instance_class_or_null(symbolHandle, Handle, Handle, Thread*)+0x7c0
V [libjvm.so+0x71bee3] SystemDictionary::resolve_or_fail(symbolHandle, Handle, Handle, bool, Thread*)+0x33
V [libjvm.so+0x3b4118] constantPoolOopDesc::klass_at_impl(constantPoolHandle, int, Thread*)+0x268
V [libjvm.so+0x4a3456] InterpreterRuntime::_new(JavaThread*, constantPoolOopDesc*, int)+0x66
j java.net.ServerSocket.accept()Ljava/net/Socket;+34
j org.serviio.ApplicationInstanceManager$1.run()V+23
j java.lang.Thread.run()V+11

90,1

    20%
j org.serviio.ApplicationInstanceManager$1.run()V+23
j java.lang.Thread.run()V+11
v ~StubRoutines::call_stub
V [libjvm.so+0x4afabf] JavaCalls::call_helper(JavaValue*, methodHandle*, JavaCallArguments*, Thread*)+0x29f
V [libjvm.so+0x4aeef5] JavaCalls::call(JavaValue*, methodHandle, JavaCallArguments*, Thread*)+0x25
V [libjvm.so+0x4af3f3] JavaCalls::call_virtual(JavaValue*, KlassHandle, symbolHandle, symbolHandle, JavaCallArguments*, Thread*)+0x1a3
V [libjvm.so+0x4af51f] JavaCalls::call_virtual(JavaValue*, Handle, KlassHandle, symbolHandle, symbolHandle, Thread*)+0x4f
V [libjvm.so+0x4e4fd6] thread_entry(JavaThread*, Thread*)+0x86
V [libjvm.so+0x74c098] JavaThread::thread_main_inner()+0x58
V [libjvm.so+0x64bbb2] java_start(Thread*)+0x112

Java frames: (J=compiled Java code, j=interpreted, Vv=VM code)
j java.net.ServerSocket.accept()Ljava/net/Socket;+34
j org.serviio.ApplicationInstanceManager$1.run()V+23
j java.lang.Thread.run()V+11

89,1

    23%
j org.serviio.ApplicationInstanceManager$1.run()V+23
j java.lang.Thread.run()V+11

v ~StubRoutines::call_stub

Subject: Re: Ubuntu 12.04 templates and java based applications
Posted by adamjseed on Mon, 02 Jul 2012 09:35:44 GMT
View Forum Message <> Reply to Message

I think I have fixed the issue. I upped the CPU to 2 and all of the java apps I'm trying are working.
One thing I have noticed is the CPU Usage is high, like 99% constantly. The same app is using
10% on a full vm.

So this leads me onto a new problem of the Java process using way more cpu that it should be
doing. grr