

From: Tim Lindholm.

Sent: 8/5/2005 12:49 PM.

To: [-] Andy Rubin.

Cc: [-]

Bcc: [-]

Subject: Re: Fwd: Java VM for Android.

This is indeed entertaining, and I'm sure lots of offers to "help" will pop up as the Android Project becomes better known here.

FWIW I largely agree with Brian, and think that the guy pointing to the various open source efforts out there is largely clueless. OK, maybe there might be a few odds and ends that would be worth picking up, where we have no value to add and the license isn't onerous. But most of that stuff is complete crap. The first 30% of a Java runtime is not nearly as valuable or costly as the last 30%, or 5%, of a commercial-quality one.

I do want to second Robert G's assessment: he, Urs and Srdjan were three of what is surely one of the best small JVM implementation teams that has existed (there were five implementors total in the original team). They shared an aesthetic for clean design and coding that you probably won't see outside of academics. They had some flaws, but not many. One is that they are stubborn Northern Europeans, who will not necessarily appreciate some other way of doing things. But I don't think they want to be JVM engineers any more, and this separation makes for less reason for concern. Frank Yellin doesn't have the elegance of the HotSpot guys, but is extremely bright and very experienced with Sun's CLDC implementations. The other Sun guys he mentions were each very good in their ways, but not in ways that directly relate to what you need -- they were Big Java guys, e.g. worrying about scalability and management.

On this same line, yesterday (or the day before??) I had lunch with the guy at Sun who is the brain behind Sun's little-JVM-on-Linux effort, on the efficient use of multiple processes on Linux. He wants to get out of Sun and is extremely interested in Google, even while not knowing anything about Android. (That email to Vineet might be able to change this?). Unfortunately he has immigration things require him to be out of the US Sept-Oct, and then career things that might make him want to stay at Sun until January. Knowledge of Android might overcome the latter, but if we tried for him in the shorter term we'd need to accommodate the away time. If we can't get him before the away time then he very likely will stay until January. This guy could be a key hire. The only possible downfall is that there had historically been bad blood between the team he worked in and the HotSpot guys, but since then he has proven his value.

-- Tim

Andy Rubin wrote:

> Thought you'd get a kick out of this thread...

>

> Begin forwarded message:

>

>> *From: *Robert Griesemer <gri@google.com <mailto:gri@google.com>>

>> *Date: *August 5, 2005 11:35:38 AM PDT *To: *Brian Swetland

>> <swetland@google.com <mailto:swetland@google.com>> *Cc: *Sascha

>> Brawer <sascha@google.com <mailto:sascha@google.com>>,

>> arubin@google.com <mailto:arubin@google.com>, nicksears@google.com

>> <mailto:nicksears@google.com>, miner@google.com

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA

TRIAL EXHIBIT 5

CASE NO. 10-03561 WHA

DATE ENTERED _____

BY _____

DEPUTY CLERK

>> <mailto:miner@google.com>, cwhite@google.com
>> <mailto:cwhite@google.com>, fadden@google.com
>> <mailto:fadden@google.com>, tcole@google.com
>> <mailto:tcole@google.com>, ficus@google.com
>> <mailto:ficus@google.com>, Patrik Reali <patrik@google.com
>> <mailto:patrik@google.com>>, Urs Hoelzle <urs@google.com
>> <mailto:urs@google.com>> *Subject: **Re: Java VM for Android*
>>
>>
>> Brian;
>>
>> I can't really comment on your project but I'd like to give you
>> some more background info in case you are interested: There are
>> several people here at Google that have intimate VM knowledge: Urs
>> Hoelzle, Srdjan Mitrovic, and I all worked on Sun's HotSpot JVM; we
>> are in fact part of the original designers of the VM. Srdjan and I
>> later wrote the "client" compiler for the VM (this is the default
>> compiler shipped with Sun's VM, for ia32 and SPARC); and we also
>> wrote the compiler for one version of Sun's CLDC VM for embedded
>> devices (also referred to as the "Monty" VM, running on StrongARM).
>> Todd Turnidge, David Stoutamire, and Ben Gomes joined the HotSpot
>> VM effort a bit later but also have done significant work in that
>> space. Srdjan, Todd, and I later also worked on a successor of the
>> HotSpot VM (for MIPS) at a startup. And last but not least we now
>> also have Tim Lindholm and Frank Yellin, the original Java VM guys
>> here at Google. But you probably don't want that many cooks...
>>
>> Anyway, you may want to consider chatting with some of these
>> people, there is a considerable amount of knowledge that can be
>> tapped. - gri
>>
>> PS: I am not cc: these extra people to reduce the amount of spam
>> you're getting... :-)
>>
>>
>>
>> On 8/5/05, *Brian Swetland* <swetland@google.com
>> <mailto:swetland@google.com>> wrote:
>>
>> I am a somewhat familiar with the grungy work involved in embedded
>> JVM building -- I wrote the VM that the Danger Hiptop platform
>> uses.
>>
>> There are some useful reasons (in my mind) for going through the
>> effort of building our own embedded VM rather than just going with
>> off the shelf solutions:
>>
>> - We'd like things to be really well-integrated with the
>> environment, small, fast, and fast to launch. Both the "run the vm
>> in a little box just for midlets" model (used by most handsets
>> today) and "run the entire world inside one vm" model (used by
>> danger) have downsides. I'd like to take advantage of running Linux
>> on CPUs with a MMU (arm9 and better) and having multiple instances
>> of the VM run in their own process space. Being able to have hard
>> limits imposed by the kernel on memory use, etc, and tear down a
>> whole VM if an app misbehaves is something we wished for often at
>> Danger. To do this, we need to make sure we can start up things
>> quickly when apps launch -- if java is core to the system and not a
>> little novelty like in current handsets, users are not going to
>> want to wait 10-15 seconds for apps to launch.
>>
>> - License choice is important. One of the goals of this project is

>> to provide an open source system that's appealing to handset OEMs.
>> The Linux kernel is GPL'd, but all the pieces above the kernel
>> that we're using or building so far are under much friendlier
>> licenses (BSD or MIT style typically). Bringing in third party
>> commercial solutions is tricky for this reason too, unless we plan
>> on buying them outright or otherwise convincing them to release
>> their software under an open source license.
>>
>> - After some amazingly negative experiences at Be, dealing with
>> Cygnus C++ compiler support, I would have a lot of concerns about
>> throwing money (away) at Redhat or Cygnus or the like for language
>> or compiler support. Of course there's also the concern of
>> shopping core parts of the system out to possibly disinterested
>> third parties.
>>
>> - The JVM is going to be a central piece of the system we're
>> building, not some little add-on on the side -- so we can provide
>> some really good java application development and user experiences.
>> I'd like to take recycle bits where possible to support javascript
>> and other language bindings, which will require doing things a
>> little differently than an off the shelf JVM.
>>
>> - Classpath is interesting for their "build it all in java"
>> approach, but from a performance perspective (which matters a lot
>> on small devices), pushing chunks of the core library to native
>> code is a huge win. Also, it is GPL with some special riders
>> (which I thought the GPL disallowed...).
>>
>> Anyway, those are just some points off the top of my head,
>>
>> Brian
>>
>> On 8/5/05, Sascha Brawer <sascha@google.com
>> <mailto:sascha@google.com>> wrote:
>>> Hi androids,
>>>
>>> I happened to stumble upon your wiki page [1]. Are you really
>>> sure you
>>> want to write your own JVM, as [2] seems to indicate? You
>>> certainly have your reasons, but it sounds like repeating lots of
>>> grungy work.
>>>
>>> So, if you don't mind, let me emit some random personal notes
>>> about the free Java scene.
>>>
>>> Everyone and their dog (not really, but way too many people) has
>>> been
>>> writing a JVM around GNU Classpath [3]. Most of them don't target
>>> embedded systems, many are crap, much has gone to oblivion, but
>>> there's also some stuff that might possibly be useful to you
>>> guys.
>>>
>>> I'd really recommend having a look at JamVM [4]: it's fast for a
>>> pure
>>> interpreter, with a decent and small codebase. JamVM is what most
>>> Classpath hackers use nowadays for development. The author seems
>>> a nice guy, he was working on optimizing Sun's and IBM's JVMs,
>>> and is now an independent contractor.
>>>
>>> I know of two companies using Classpath for JVMs that target
>>> embedded systems:
>>>

>>> /k/ [5] is a cover-up for one guy having his fun. My personal
>>> impression from the Classpath meetings is that the author is
>>> really into free software; I'm pretty sure he would be keen on a
>> contract for
>>> an open-source embedded JVM. In the meetings list, he seemed to
>>> know what he's talking about, but I haven't chatted that much
>>> with him.
>>>
>>> Aicas [6] is a real company whose embedded JVM is based on
>> Classpath.
>>> Since they haven't given anything back to the project, I'd be
>>> surprised if they would be interested in a contract for an
>> open-source
>>> embedded JVM.
>>>
>>> There's also a Bytecode-to-C compiler [7], but I've no idea
>>> whether it's any good. I don't think it gets much used. But it
>>> might be interesting in case you want to use an ahead-of-time
>>> compiler for selected hot spots, and use a pure interpreter like
>>> JamVM for the rest.
>>>
>>> Kaffe [8] has been used for embedded systems, but the licensing
>> is in
>>> a limbo (GPL, copyright held by a dead company).
>>>
>>> You could pay Redhat for tweaking gcj/gcc, but their focus
>> really is
>>> on desktop systems. But since you mention C++ linkage on your
>>> wiki: gcc uses the same vtables for Java and C++, they call this
>>> "Cygnus Native Interface (CNI)". There has been lots of talk
>>> about
>> giving gcc
>>> a better jitter for dynamically loaded bytecode; they currently
>> have a
>>> very inefficient interpreter as part of the Java runtime
>> library. But
>>> last I've heard, Redhat's plan now is to use the gcc backend as
>> a JIT
>>> -- hairy stuff, and certainly totally unusable for an embedded
>> sytem.
>>>
>>> If you need more info around the free Java projects, or if want
>>> to establish a contact, please feel to talk to either me or
>>> Patrik
>> Reali.
>>> We've both been somewhat active in this scene before joining
>>> Google, so we know most people from meetings.
>>>
>>> Oh, you surely know that quite a few people at Google have a JVM
>>> background? For instance Robert Griesemer or Urs Hoelzle. I hope
>>>
>> you
>>> don't mind that I'm taking the liberty to cc them on this post,
>>> in case they want to comment/shoot me down.
>>>
>>> Best wishes, and have fun with Android,
>>>
>>> -- Sascha
>>>
>>> [1] <http://wiki.corp.google.com/twiki/bin/view/Main/Android> [2]
>> <http://wiki.corp.google.com/twiki/bin/view/Main/AndroidJavaVM>
>> <<http://wiki.corp.google.com/twiki/bin/view/Main/AndroidJavaVM>>

>>> [3] <http://www.gnu.org/software/classpath/> [4]
>>> <http://jamvm.sourceforge.net/> [5]
>>> <http://www.kiffer.be/k/profile.html> [6] <http://www.aicas.com/> [7]
>>> <http://jcvms.sourceforge.net/> [8] <http://www.kaffe.org/>
>>>
>>
>>
>