

PLAINTIFF'S
EXHIBIT
4262
Comes v. Microsoft
tabbles

From: David Sauntry
Sent: Wednesday, April 02, 1997 3:00 PM
To: Rodney Vieira; Alessandro Muti; Edward Ye
Cc: Guy Shefner (RhoTech); Sreeram Nivarthi; Colin Miller; Michael Toutonghi
Subject: RE: Future Plans for Java on Win3.1

the ce vm is tied pretty closely to windows ce, which is much like win32. WCE is hosted of 32 bit cpus, and we take advantage of threads, synchronization, virtual memory, etc. You should review our src code to better estimate if it will be of value for you.

That said, here's the info you requested:

- what is the footprint of your vm

~100k for the vm (including native code for io/lang/net/util). add around 200k for the base classes (io, lang, net, util). add another 1M or so for awt and applet.

- what level of functionality do you support...JDK 1.1 compatible?

these numbers are for 1.02. We're moving (fast) toward jdk 1.1 with wce v2 (alder release).

- what is your schedule

4/30 1.1 preview at WCE devcon
6/13 1.1 beta
8/29 1.1 alder rtm

- would are your thoughts on whether the vm can be easily ported to win16.
we rejected the win32 vm in part because its tightly tied to win32.

without really thinking about this, I think that we're just as tied to win32 (or at least the wince subset). java threads are win32 threads, etc. You are welcome to look thru our src to see if you can use it.

For v3 (birch/1Q98), we're going to look into doing a jitter -- something that will generate native code for the stack machine -- we think we can improve our execution speed by 3-4X by just getting rid of the overhead of the interpreter loop... No schedule or spec for this yet.

daves

From: Rodney Vieira
Sent: Wednesday, March 26, 1997 4:00 PM
To: Ben Slivka; Charles Fitzgerald; David Cole; John Ludwig; Michael Toutonghi; Russ Arun; Thomas Reardon
Cc: Alessandro Muti; Colin Miller; Edward Ye; Michael Guo; Vamshidhar Reddy; Rodney Vieira
Subject: Future Plans for Java on Win3.1

As you know we shipped the Java VM (JDK 1.02) for Win 3.1 in mid March. We are now thinking about what our strategy should be moving forward. Do we stop investing in Java and focus on core HTML features? Do we simply make incremental improvements on the 1.02 code or should we move to the JDK 1.1 codebase? even better..do we adopt the MS VM?

Below I have summarized some of the alternatives we are considering....

1. *Stop investing in Java for Win 3.1*

Several people have suggested that we should give this more serious consideration. The thinking is that corps will not target Win3.1 for Java development because of the performance limitations and that

MS-PCA 2615137

CONFIDENTIAL

making huge gains in perf will be extremely difficult (Note: Peterku believes a JIT would only give us between 2-5X performance gain over the existing VM because of probs associated with 16 bit code generation) Consequently, we should think of Java on Win3.1 as a marketing "check box"...one which we have already satisfied.

My thoughts are that there are several minor reasons which together justified our continued investment in Java for Win 3.1. They are....

- As far as we know NS will have a 1.1 compatible VM for Win3.1
- Even if perf is limited, NS will have an advantage in selling into corps if they can claim 1.1 support.
- Compromising on Win3.1 development will just fuel the rhetoric about how we are not really committed to cross platform solutions and abandoning win 3.1.
- Performance is acceptable for a certain class of applets/applications. We are fine for the "eye candy" stuff and we aren't that bad on apps which aren't math intensive/graphical.
- The Win 3.1 VM is what we currently have for NT 3.51. How do we explain the lack of 1.1 support to these users.

Privilege Material Redacted

2. Make incremental improvements to the existing 1.02 codebase.

If we took this approach the intent would be to focus on performance and sacrifice additional major functionality like Javabeans, JAR file support and other JDK 1.1 features. Once we start trying to add major functionality to this codebase it makes more sense to just use the 1.1 codebase.

Using the existing code also allows us to release improved revs of our VM more frequently. If we use 1.1 or the MS VM we will probably only do qfes on the existing VM until we're ready to go public in several months with the new VM.

My thoughts are that this option is less desirable than (3) or (4) because the press expects 1.1 level functionality (NS will probably be 1.1 compatibility on Win3.1). Our corp customers probably also expect 1.1 features, although it could be argued that they are not depending on Java for Win16.

Again there may also be legal complications with option (2). (I suspect we could figure out how to work around this if we felt this option were the way to proceed)

3. Adopt the JDK 1.1

The two major positive aspects to this plan are that it involves using a codebase that we are reasonably familiar with (less risk/time than porting the MS VM) and we get 1.1 compatibility....which is probably good enough for our corp customers since they seem to be writing to the lowest common denominator (this will probably continue to be the case for as long as Win 3.1 is interesting). The principle down side is that (1) we remain dependent on Sun, (2) can not leverage our own code (and associated developers), (3) it sends the wrong message externally...although this hasn't been an issue so far and (4) we may not choose to implement Java to COM because it will be a significant investment in resources if we start with the Sun JDK. Given that we are not investing heavily in ActiveX controls for Win16 this may not be a big deal. Java apps can use Javabeans.

Clearly, option (4) is a better long term solution, however given the finite existence of Win 3.1 it may be appropriate to go with 1.1 based on the lower risk and shorter development time. Unfortunately, its difficult to predict how long it will take to port the MS VM without spending alot of time going thru the code. Nonetheless alessandro & co are making an effort to gather more data. However, given our past experience porting Win32 code

MS-PCA 2615138

CONFIDENTIAL

its safe to say that there will be many unknowns/surprises. We are also concerned about the fact the the MS Java code we would pick up is still in development.

Our conservative estimate for porting JDK 1.1 is 8 months (Final release) with the existing 4 dev resources we have on the team.

4. Adopt the Win 32 Java VM

This approach is attractive for obvious reasons....leverage MS code, leverage MS developers, more functionality and consistent with overall strategy. The main concern is that the investment needed to pull this off may exceed the benefit we derive over the remaining life of Win31. Another consideration is whether our customers can live with 1.02 while we prepare the new release of the MS Java VM or would we take an unacceptable beating because of our lack of a quick 1.1 solution (Win16).

My sense is that we should pursue (3) or (4).

Thoughts/Comments?? Other approaches?

Rodney

MS-PCA 2615139

CONFIDENTIAL