

Coldfusion MX and the 64 bit JVM

Posted At : June 18, 2006 6:32 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Upgrading

CF Muse Reader, Eric Priest from Montana University, Asks:

We are getting ready to upgrade our web and database servers from Windows 2000, running SQL server 2000, 32-bit, to Windows 2003 server running SQL server 2005, 64-bit. We have a lot of legacy applications written in JavaScript and Coldfusion mx studio. I am wondering if these changes will break any of our existing applications and how can I determine the impacts?

First, you should know that this is not a widely used platform yet - at least not for web servers. Secondly, if you are running both SQL 2005 and Coldfusion on the same 64 bit server make sure and buy 6 Gigs or more of RAM. Now, assuming you have separate servers SQL will run splendidly in 64 bit mode (assuming you have the right version). Let's talk about Coldfusion.

I put this question to my guru list and they came up with some wonderful information for you.

- First, you must use a 64 bit JVM - that means moving to 1.05.05 right away.
- Secondly, JRUN will *not* tolerate this JVM on a 64 bit platform. That means you will need to move to the J2EE version of CF running on JBOSS or TOMCAT. I assume if you have the chops to license the SQL server and buy 64 bit hardware you won't mind the licensing cost.
- Thirdly there is the issue of heap size and memory detailed below.

Memory and Heap Sizes in CFMX

You might be aware of a limitation of 2Gb of contiguous memory for a JVM instance in Windows. This limitation has to do with 32 bit Windows running on x86 hardware. Since the Sun JVM needs *contiguous space* in memory (registers stacked end to end). For this reason I typically use a limit of 1.3 gigabytes as the maximum heap size value (1380m). On a 64 bit system however, the underlying architecture is completely different. So different in fact, that it may be assumed that you can throw this convention away. Folks are sometimes surprised at just how much memory a 64 bit system consumes compared with a 32 bit system. But when you think about it, it stands to reason. A 64 bit system is designed to manipulate much larger blocks of data. Everything on it is engineered to have more capacity to accommodate that new condition - including the memory allocations.

Unfortunately I do not yet have the experience to give you any specific recommendations on a starting point for the heap size. You will have to keep in mind that you cannot allocate more memory on the heap than the physical memory available on the server (that would be the *minimum* number plus the *permSize* number) otherwise you will get an error indicating "cannot allocate enough virtual memory" in the *-out.log or *-err.log file (I can't remember which).

Additional Resources

I do have the following links that may be helpful:

Nat Papovich - 64 bit CF

Ben Forta - Running 32 bit CF on a 64 bit system

Patrick Quinn on the Heap Size Limitations on 32 Bit System

Final Note

It should be noted here than current 64 bit systems usually support 32 bit programs. It is possible to run 32 bit CF on a 64 bit system and it will probably work as before. See Ben's article above for an example. My thanks to the Cfguru list, Patrick Quinn and Nat Papovich for help with this answer.

As always I welcome additional information. I will happily post a follow up as we gather real world experience on this issue.