

[articles](#) [Q&A](#) [forums](#) [lounge](#)

Search for articles, questions, tips



10 Things to Know about Memory Mapped File in Java



Javin Paul, 17 Nov 2013

CPOL

Rate this:

★★★★★ 4.71 (4 votes)

Memory Mapped IO is one of the fastest IO options available for Java program. If you are writing latency sensitive application, where IO can skew your numbers, then using `MemoryMappedByteBuffer` is a good option.

What is Memory Mapped File and IO in Java

Memory mapped files are special, which allows Java program to access contents directly from memory, this is achieved by *mapping whole file or portion of file into memory* and operating system takes care of loading page requested and writing into file while application only deals with memory which results in very fast IO operations. Memory used to load Memory mapped file is outside of [Java heap Space](#). Java programming language supports memory mapped file with the `java.nio` package and has `MappedByteBuffer` to read and write from memory.

Advantage and Disadvantage of Memory Mapped File

Possibly the *main advantage of Memory Mapped IO is the performance* which is important to build high-frequency electronic trading system. Memory Mapped Files are way faster than standard file access via normal IO. Another big advantage of memory mapped IO is that it allows you to load a potentially larger file which is not otherwise accessible. Experiments show that memory mapped IO performs better with [large files](#). Though it has a disadvantage in terms of increasing number of page faults. Since operating system only loads a portion of the file into memory if a page requested is not present in memory, then it would result in the page fault. It can also be used to share data between two processes.

Memory Mapped IO Support in Operating System

Most of the major operating systems like Windows platform, UNIX, Solaris and other [UNIX](#) like operating system supports memory mapped IO and with **64 bit architecture** you can map almost any file into memory and access it directly using Java programming language.

Important Points of Memory Mapped IO in Java

Here are some of the important facts to know about Memory Mapped File in Java:

1. Java supports Memory mapped IO with `java.nio` package.
2. Memory mapped files is used in **performance sensitive application**, e.g. high-frequency electronic trading

platforms.

3. By using memory mapped IO, you can **load portion of large files in memory**.
4. Memory mapped file can result in **page fault** if requested page is not in memory.
5. Ability to map a region of the file in memory depends on the addressable size of memory. In a 32 bit machine, you cannot access beyond 4GB or 2^{32} .
6. Memory mapped IO is much faster than Stream IO in Java.
7. Memory used to load File is outside of **Java heap** and resides on shared memory which allows two different processes to access File.
8. Reading and writing on memory mapped file is done by the operating system, so even if your Java program crashes after putting content into memory, it will make to File until OS is fine.
9. Prefer Direct Byte buffer over NonDirect Buffer for faster performance.
10. Don't call **MappedByteBuffer.force()** method too often, this method is meant to force operating system to write content of memory into disk, So if you call **force()** method each time you write into memory mapped file, you will not see true benefit of using mapped byte buffer, instead it will be similar to disk IO.
11. In case of power failure or host failure, there is a slim chance that content of memory mapped file is not written into the disk, which means you could lose critical data.

That's all folks. Memory mapped IO is an important concept for advanced Java developers, especially for writing high-performance applications in Java. If you like this tip, then you may like to visit my blog <http://javarevisited.blogspot.com> for more such tips.

Thank you!

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

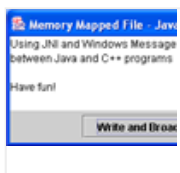
Share

EMAIL

TWITTER

About the Author

You may also be interested in...



Using Memory Mapped Files and JNI to communicate between Java and C++ programs



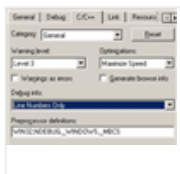
On-Demand Scalability for Mobile Game Apps



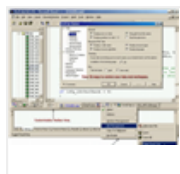
Using Memory Mapped Files in ASP.NET using VB.NET



SAPrefs - Netscape-like Preferences Dialog



Finding crash information using the MAP file



Window Tabs (WndTabs) Add-In for DevStudio

Comments and Discussions

You must [Sign In](#) to use this message board.

Search Comments

Go

-- There are no messages in this forum --

[Permalink](#) | [Advertise](#) | [Privacy](#) | [Terms of Use](#) | [Mobile](#)
Web02 | 2.8.160311.1 | Last Updated 17 Nov 2013

Select Language ▼

Layout: [fixed](#) | [fluid](#)

Article Copyright 2013 by Javin Paul
Everything else Copyright © [CodeProject](#), 1999-2016