FREE eBook

LEARNING

jvm

Free unaffiliated eBook created from
Stack Overflow contributors.

#jvm

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: jvm

It is an unofficial and free jvm ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jvm.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with jvm

Remarks

This section provides an overview of what jvm is, and why a developer might want to use it.

It should also mention any large subjects within jvm, and link out to the related topics. Since the Documentation for jvm is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting jvm set up or installed.

Enabling Parallel GC

Parallel GC is Stop-The-World (STW) collector which stop all the application threads when running the garbage collector.

When Parallel GC was introduced it was only enabled the parallel GC in young generation collector and OldGeneration Collector was single thread stop-the-world collector, but later introduce separate command line option to enable the Old Parallel.

Enable Parallel GC on Java 6 : -XX:+UseParallelOldGC

Enable Parallel GC on Java 7u4 onward: -XX:+UseParallelGC OR -XX:+UseParallelOldGC

Parallel GC was made default on Java7 update 4 onward , however specifying the one of above option makes it explicit.

Read Getting started with jvm online: https://riptutorial.com/jvm/topic/3432/getting-started-with-jvm

Chapter 2: JVM Heap

Examples

Setting the maximum heap size.

Most JVMs have an option to set the maximum heap size e.g.

```
-Xmx64m
-Xmx8g
```

In Java 1.0 to 1.2 you could use

```
-mx64m
```

and this is still available on some JVMs for backward compatibility (E.g. Oracle JVM).

There are a few common misconceptions about this setting.

- It doesn't set the heap size, only the maximum. `-Xms` sets the initial heap size.
- It doesn't set the amount of memory the JVM will use. While the heap is an important area of memory, there are many other regions for code Perm Gen/Metaspace, thread stacks, GUI components, direct memory etc.

The amount of memory used at run time can change dynamically.

Specify heap region size

The latest JVMs supports Garbage First GC (G1 GC) and consists of set of regions which accumulate to make young and old generation.

The JVM will have approximately 2048 reagions and set heap region size accordingly from 1 MB to 32 MB and power of 2 bounds. This is important parameter which decide what size of object that can be store in a region.

Heap region size = Heap size/2048

you can overwrite the adaptive selection of the region size by comand line JVM paramter -XX:G1HeapRegionSize=n

Read JVM Heap online: https://riptutorial.com/jvm/topic/4058/jvm-heap

Chapter 3: JVM Heap Dump

Examples

Generating heap dump upon OutOfMemoryError

Note: This example is based on the Oracle JVM implementation.

Built-in tools like `jmap`, `jconsole`, and `jvisualvm` are available in a JDK and can be used to generate and analyze heap memory dumps taken from a running JVM application. However, one option to generate a heap dump without using JDK tools is to add the VM argument `-XX:+HeapDumpOnOutOfMemoryError` which tells the JVM to automatically generate a heap dump when an OutOfMemoryError occurs, and the argument `-XX:HeapDumpPath` to specify the path for the heap dump.

Also see: Java HotSpot VM Options, specifically:

> **-XX:HeapDumpPath=./java_pid.hprof** Path to directory or filename for heap dump. Manageable. (Introduced in 1.4.2 update 12, 5.0 update 7.)
>
> **-XX:-HeapDumpOnOutOfMemoryError** Dump heap to file when java.lang.OutOfMemoryError is thrown. Manageable. (Introduced in 1.4.2 update 12, 5.0 update 7.)

If a concurrent collector such as CMS or G1 are used then a FullGC can be considered a failure mode and using `HeapDumpBeforeFullGC` or `HeapDumpAfterFullGC` can be useful to diagnose them.

Read JVM Heap Dump online: https://riptutorial.com/jvm/topic/3901/jvm-heap-dump

Credits

S. No	Chapters	Contributors
1	Getting started with jvm	Community, jayalalk
2	JVM Heap	jayalalk, manouti, Peter Lawrey
3	JVM Heap Dump	manouti, the8472