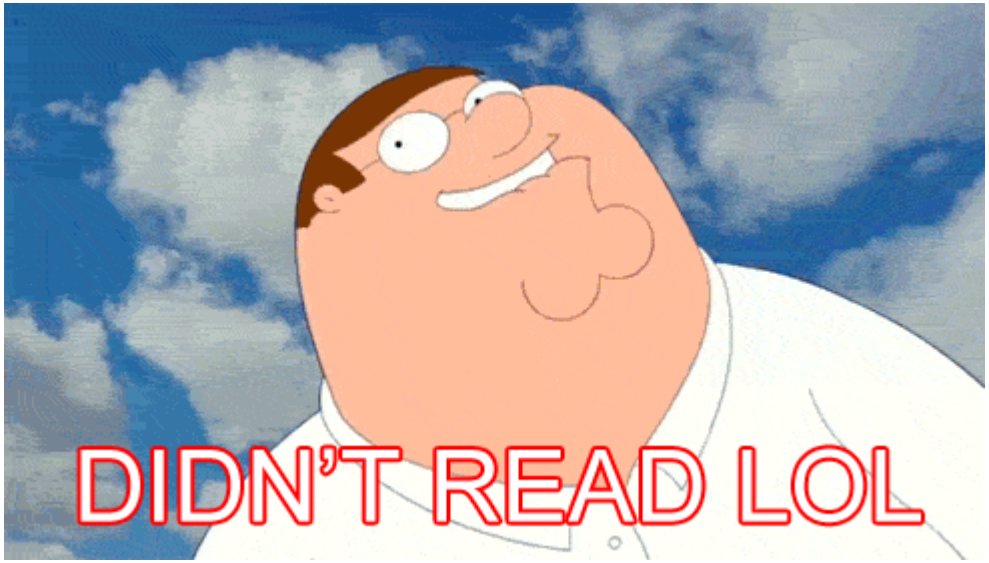
Podstawy JVM (dla tych, którym nie chce się czytać blogów)

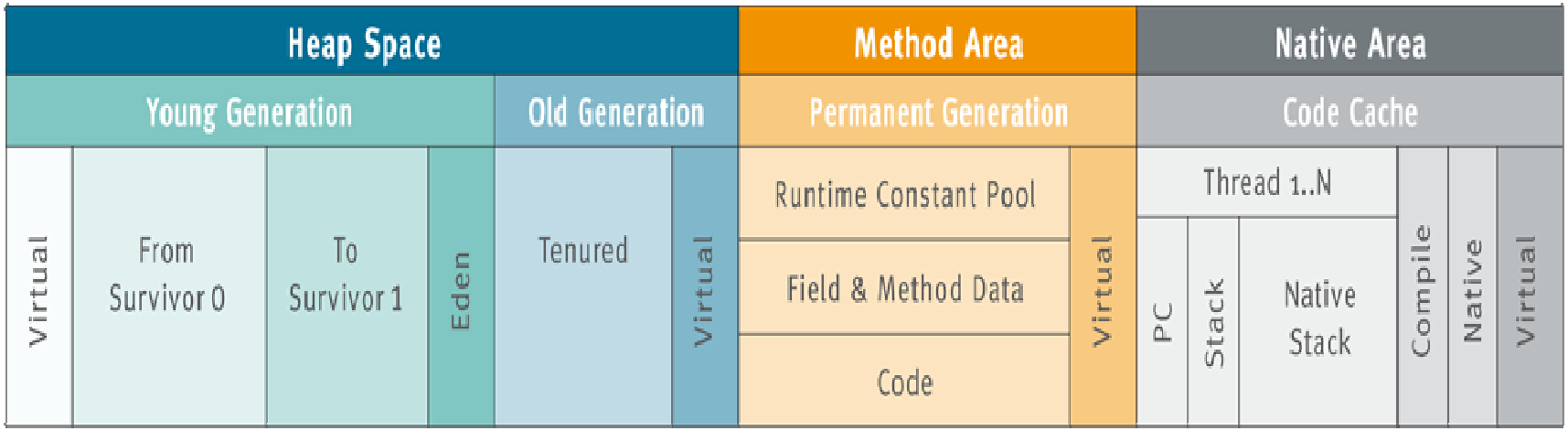


Bartłomiej Kaflowski

JVM Implementations

- Oracle Hotspot
- Oracle JRockit
- Azul Zing
- IBM JVM
- …

Hotspot JVM 7 Memory Model



http://www.pointsoftware.ch/wp-content/uploads/2012/10/JUtH_20121024_RuntimeDataAreas_1_MemoryModel.png

Adaptive Heap Size

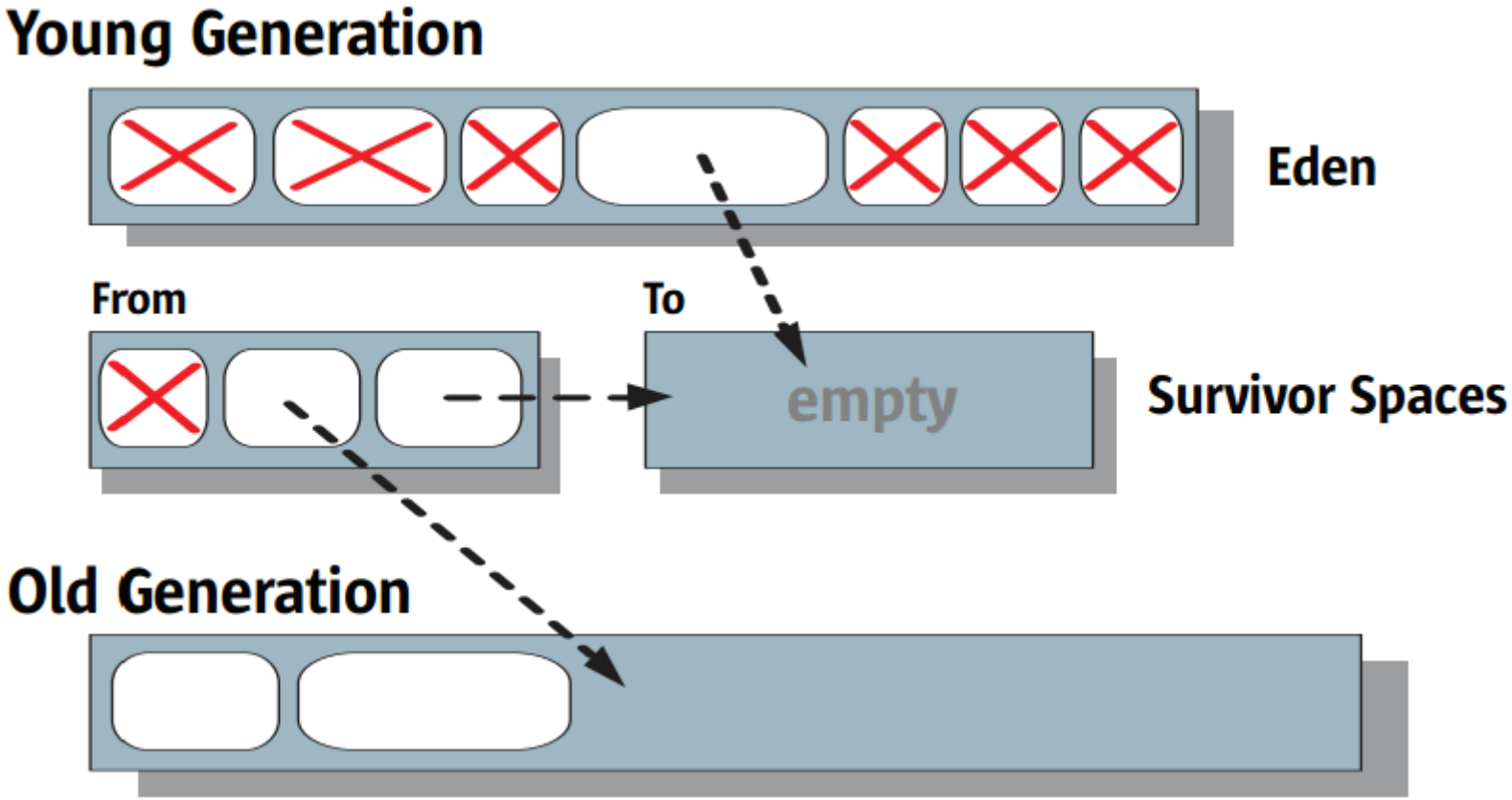
- -XX:+UseAdaptiveSizePolicy
- -XX:+PrintAdaptiveSizePolicy
- Xms <-> Xmx
- -XX:NewRatio=N
- -XX:NewSize <-> -XX:MaxNewSize

Young Generation GC

- SerialGC
- ParallelGC
- ParNewGC
- G1

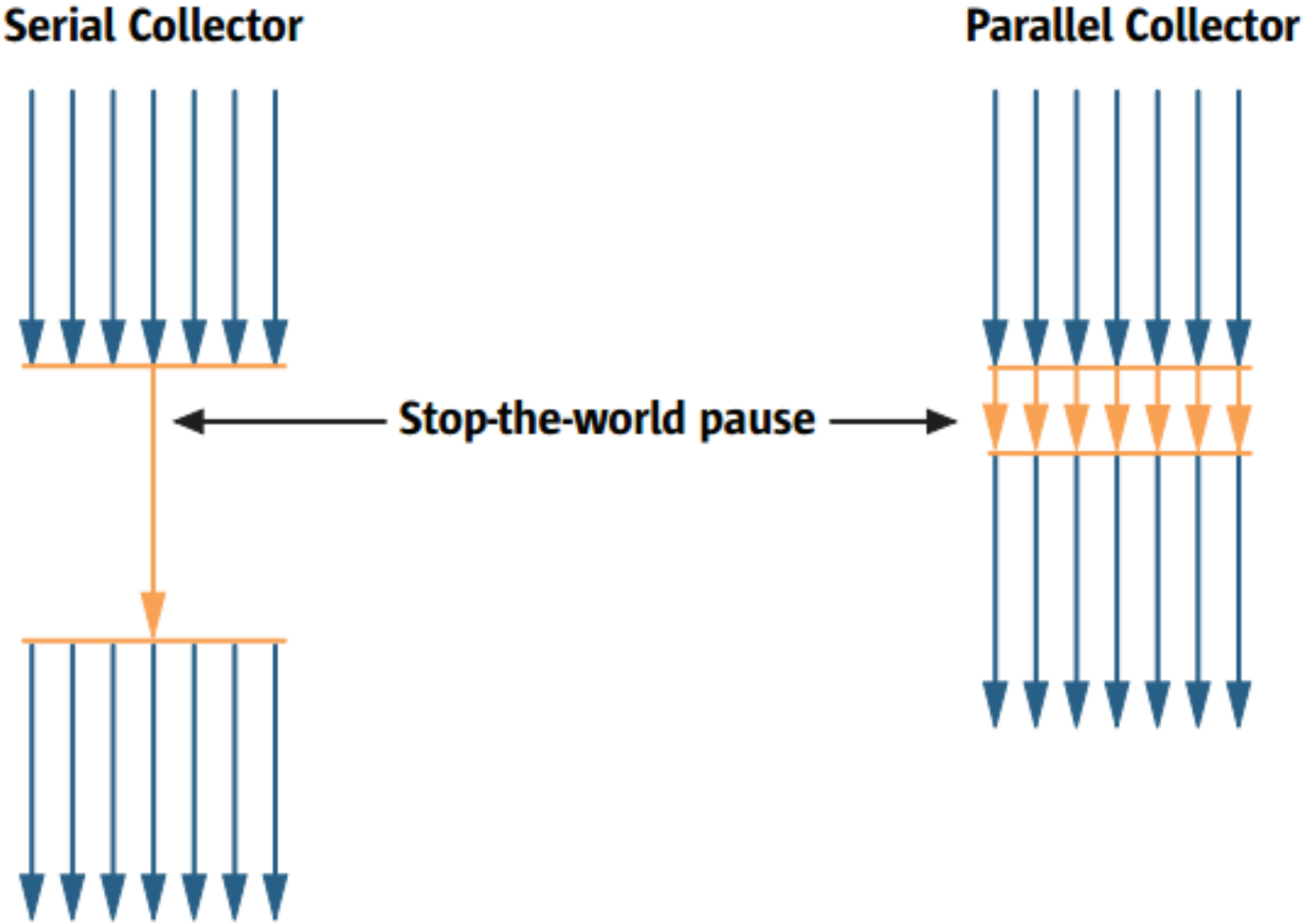
http://2.bp.blogspot.com/-Le4QjeHJH70/TiSk5-9MUYI/AAAAAAAAEd0/q9a9LtJ_w2E/s1600/09_basurero.gif

SerialGC



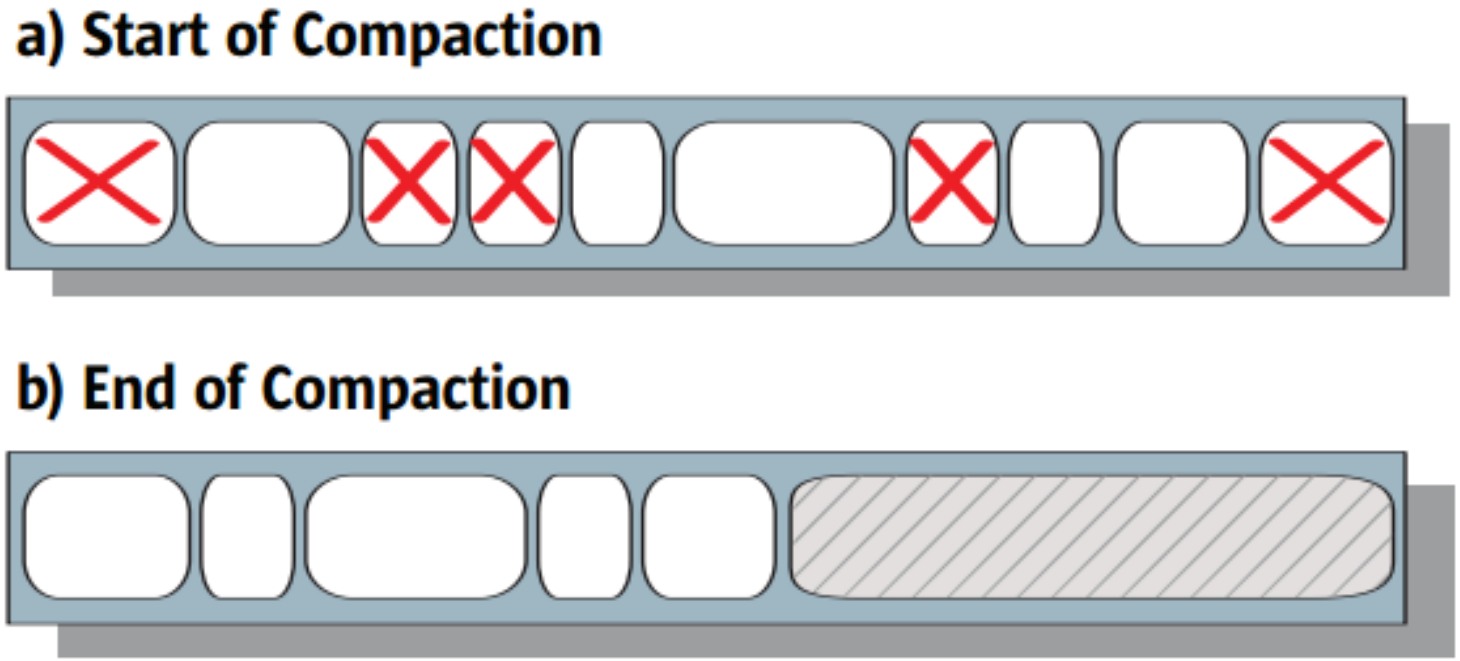
http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf

Parallel Collector



http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf

OldGen Compaction



http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf

Adaptive Sizing ThroughputGC

1. -XX:MaxGCPauseMillis=N
2. -XX:GCTimeRatio=N

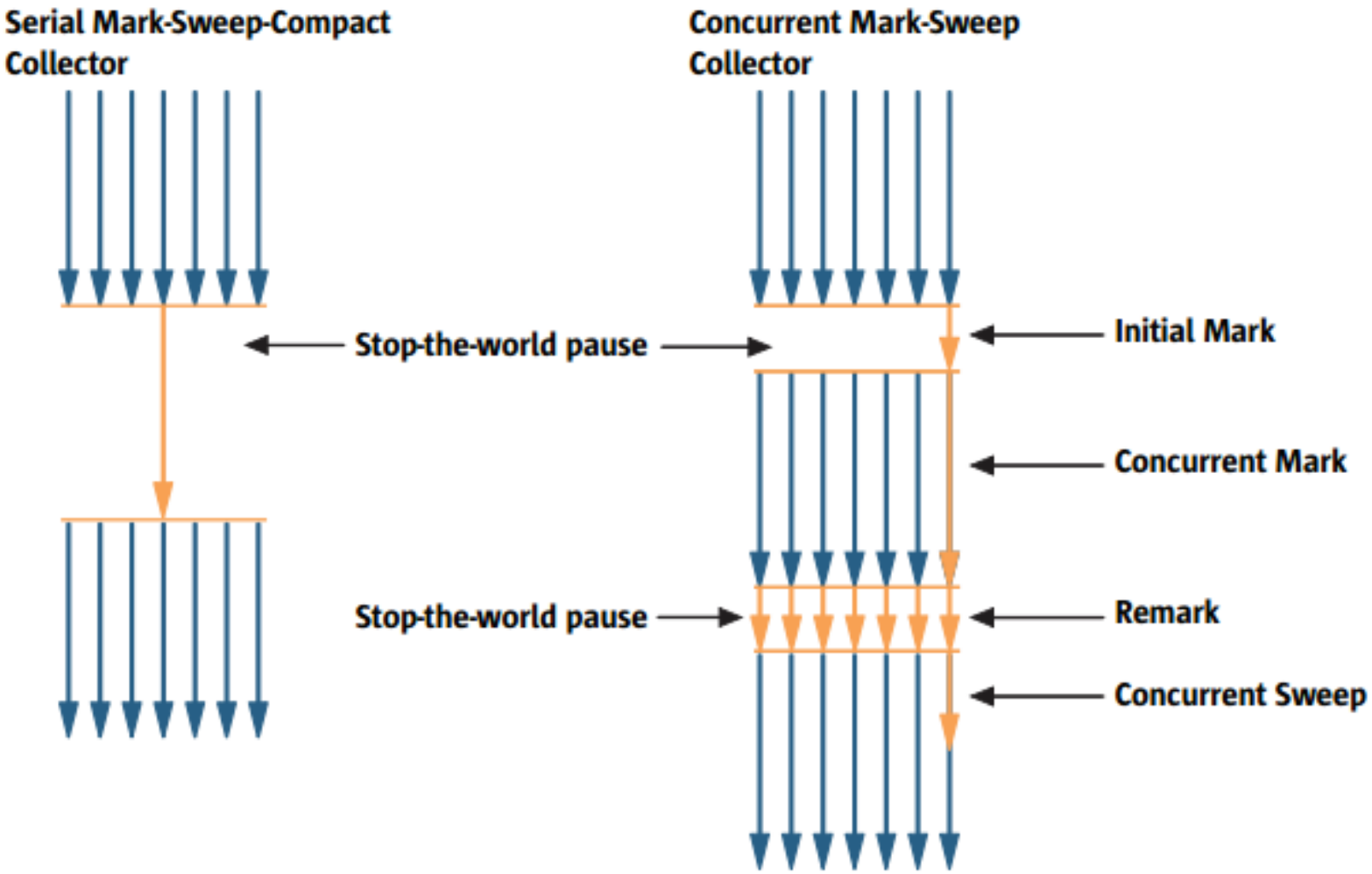
ThroughputGoal = 1 – 1 / (1 + GCTimeRatio)

Old Generation GC

- MarkSweepCompact
- ParallelOldGC
- CMS
- G1

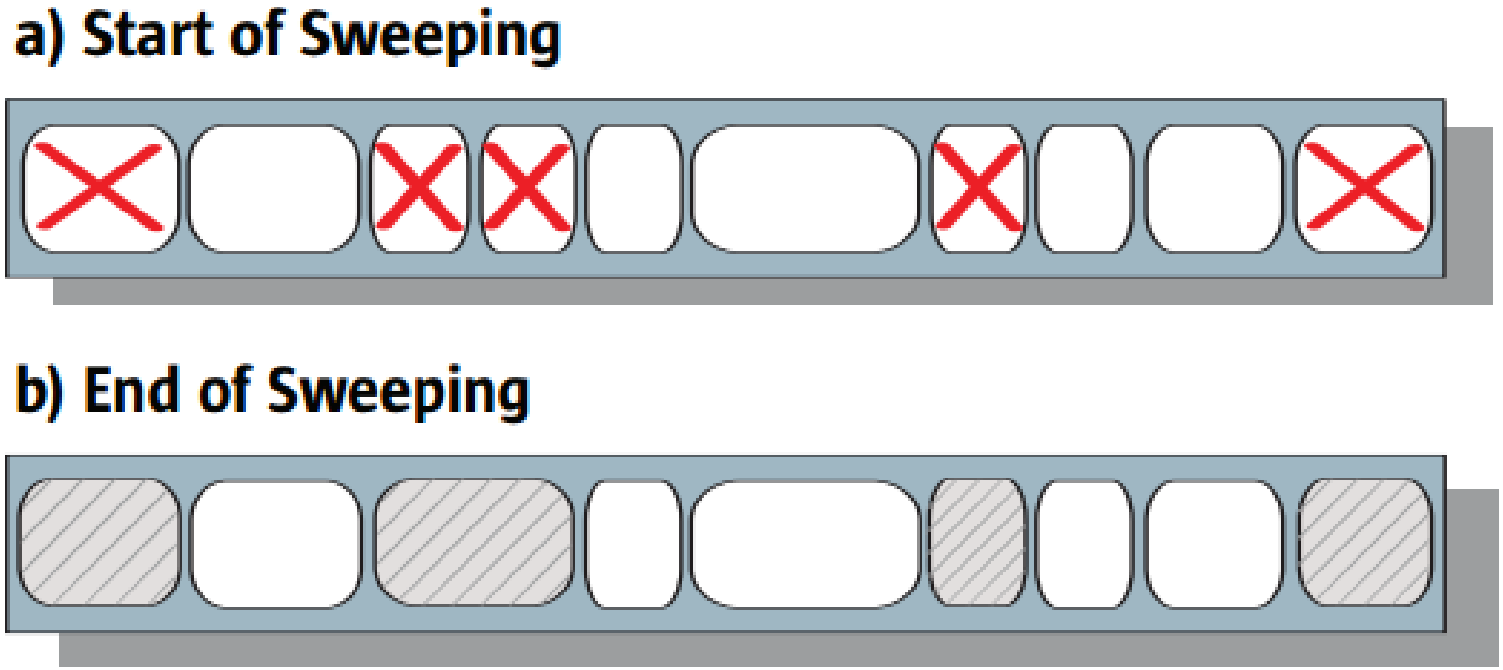
http://androidarts.com/misc/garbage.jpg

CMS



http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf

Lack of OldGen compaction



http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf

Adaptive Sizing CMS

- Concurrent Mode Failures
- -XX:MaxGCPauseMillis=N
- -XX:GCTimeRation=N
- -XX:CMSInitiatingOccupancyFraction=N
- -XX:+UseCMSInitiatingOccupancyOnly

Monitoring GC

- -XX:+PrintGC
- -XX:+PrintGCDetails
- -XX:+PrintGCTimeStamps
- -XX:+PrintGCDateStamps
- -Xloggc:<file>
- -XX:+UseGCLogFileRotation
- -XX:GCLogFileSize=<number>

Throughput GC log

2014-09-08T03:43:38.014+0200: 74.955: [GC [**PSYoungGen**: 209761K->25781K(198144K)] 269365K->95842K(285696K), 0.0392011 secs] [Times: **user=0.14 sys=0.00, real=0.03 secs**]

2014-09-08T03:43:38.062+0200: 75.011: [**Full GC** [**PSYoungGen**: 25781K->0K(198144K)] [**ParOldGen**: 70060K->62464K(137728K)] 95842K->62464K(335872K) [**PSPermGen**: 108119K->107883K(216064K)], 0.4670577 secs] [Times: **user=1.40 sys=0.06, real=0.47 secs**]

CMS GC log

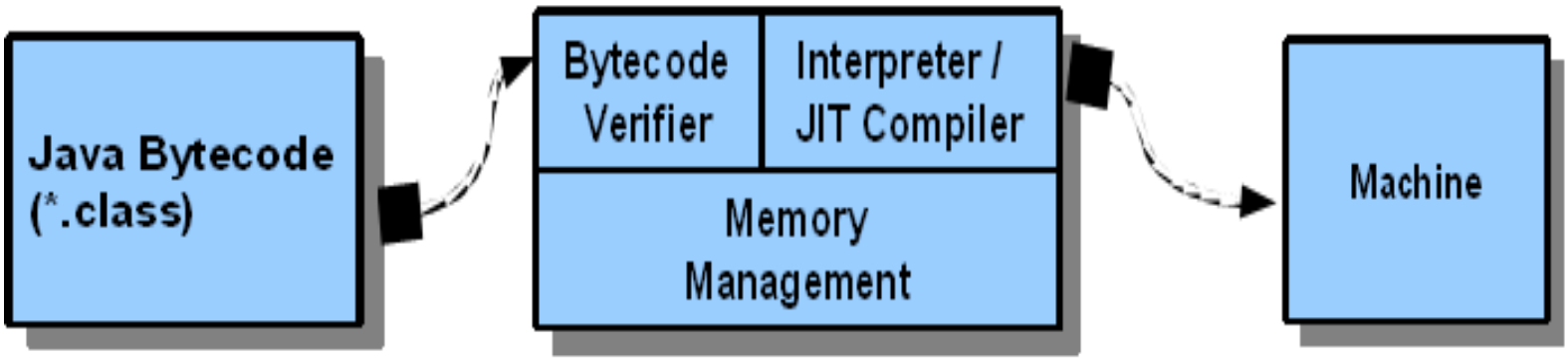
- 2014-09-08T03:33:02.741+0200: 48.413: [GC2014-09-09T03:33:02.741+0200: 48.413: [**ParNew**: 39296K->4352K(39296K), 0.0194475 secs] 81560K->52751K(126720K), 0.0195754 secs] [Times: **user=0.06 sys=0.00, real=0.02 secs**]

- 2014-09-08T03:33:02.756+0200: 48.433: [GC [**1 CMS-initial-mark**: 48399K(87424K)] 52754K(126720K), 0.0034731 secs] [Times: user=0.00 sys=0.02, real=0.02 secs]
- 2014-09-08T03:33:02.772+0200: 48.437: [**CMS-concurrent-mark-start**]
- 2014-09-08T03:33:02.865+0200: 48.534: [**CMS-concurrent-mark**: 0.030/0.097 secs] [Times: user=0.00 sys=0.00, real=0.09 secs]
- 2014-09-08T03:33:02.865+0200: 48.534: [**CMS-concurrent-preclean-start**]
- 2014-09-08T03:33:02.865+0200: 48.535: [**CMS-concurrent-preclean**: 0.001/0.001 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
- 2014-09-08T03:33:02.865+0200: 48.535: [**GC[YG occupancy**: 4935 K (39296 K)]
- 2014-09-08T03:33:02.865+0200: 48.535: [**Rescan (parallel)** , 0.0012456 secs]
- 2014-09-08T03:33:02.865+0200: 48.536: [weak refs processing, 0.0001246 secs]
- 2014-09-08T03:33:02.865+0200: 48.536: [scrub string table, 0.0004180 secs] [1 CMS-remark: 48399K(87424K)] 53334K(126720K), 0.0018758 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
- 2014-09-08T03:33:02.865+0200: 48.537: [**CMS-concurrent-sweep-start**]
- 2014-09-08T03:33:02.881+0200: 48.555: [**CMS-concurrent-sweep**: 0.017/0.018 secs] [Times: user=0.02 sys=0.00, real=0.02 secs]
- 2014-09-08T03:33:02.881+0200: 48.555: [**CMS-concurrent-reset-start**]
- 2014-09-08T03:33:02.897+0200: 48.561: [**CMS-concurrent-reset**: 0.006/0.006 secs] [Times: user=0.00 sys=0.01, real=0.02 secs]

GCViewer to the rescue!

http://funny-pics.co/wp-content/uploads/funny-dog-on-rescue-board-445x299.jpg

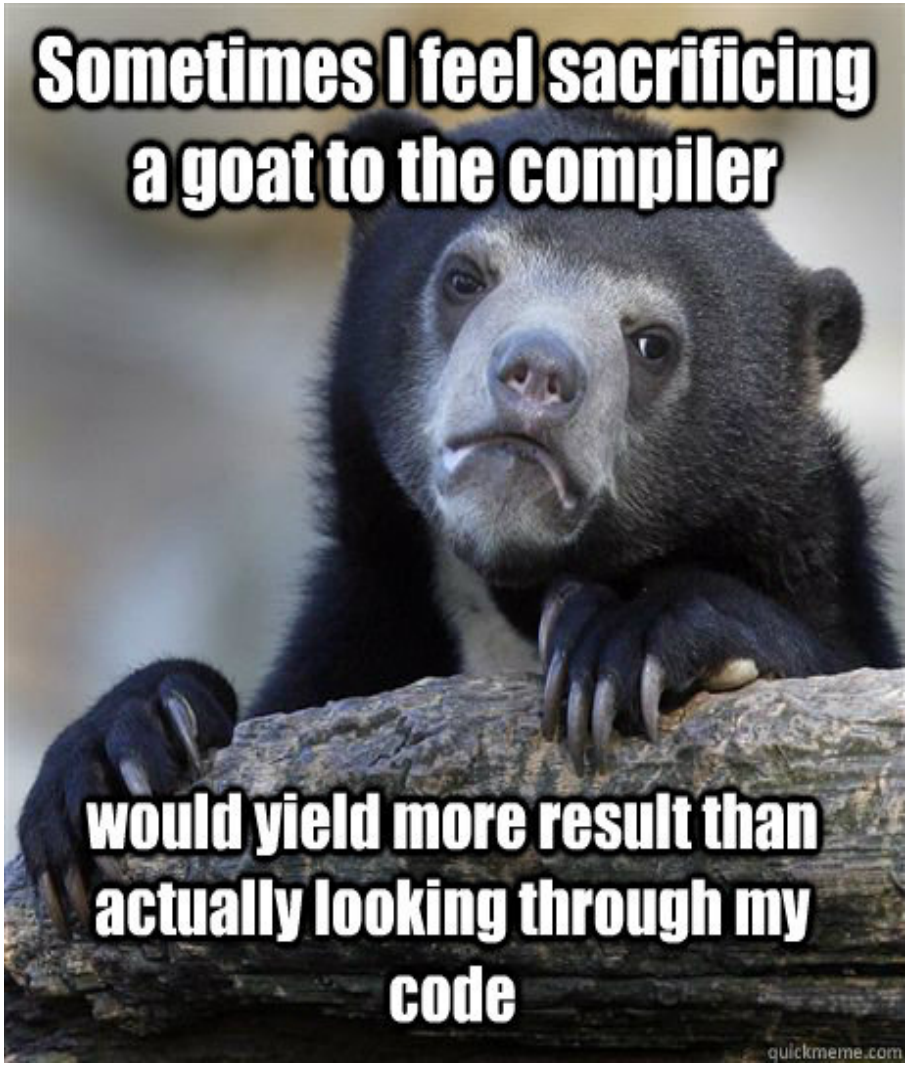
Compilation Process



JVM Internal Architecture

http://j4school.files.wordpress.com/2013/01/jit.gif

JIT



- client (C1)
- server (C2)
- tiered compilation

http://www.quickmeme.com/img/78/78dcb375218fcb9ab3cb8be64e006a009a7288ba163a381b463c24e20b6c5270.jpg

JIT „HOTNESS”

- Compilation Thresholds
  - client (1,500)
  - server (10,000)
- OSR (on-stack replacement)

http://3.bp.blogspot.com/-ZG7Gbk5tStA/UDYUdEStxyI/AAAAAAAAABQ/RxpoeNcs2NA/s1600/funny-owl-suprised-eyes.jpg

Code Cache

JVM type	Default size
32-bit server	32MB
64-bit server	48MB
64-bit tiered compilation	96MB
64-bit tiered compilation, Hotspot 8	240MB

-XX:ReservedCodeCacheSize=N
-XX:InitialCodeCacheSize=N
-XX:+UseCodeCashFlushing

JIT Code Optimizations Flags

- -XX:+DoEscapeAnalysis
- -XX:-Inline
- -XX:MaxFreqInlineSize=N (325 bytes)
- -XX:MaxInlineSize=N (35 bytes)

JIT Monitoring

- jstat
  - compiler <pid>
  - printcompilation <pid> <interval>
- -XX:+PrintCompilation
  - not entrant
  - zombie

Time for some examples!

http://blogs.salesforce.com/.a/6a00e54ee3905b883301543865717e970c-pi

ThreadDump

- Tools to make:
  - jstack
  - kill -3
  - jcmd
- Tools to analyze:
  - „by hand”
  - ThreadLogic

HeapDump

- Tools to make:
  - jmap
  - jcmd
  - -XX:OnOutOfMemoryError
- Tools to analyze:
  - jhat
  - MAT
  - „by hand”

OutOfMemoryErrors

- PermGen
- Heap
- Finalizer

More advanced tools

- JConsole
- JVisualVM
- Java Mission Control (7u40)
- FlightRecorder

  (-XX:+UnlockCommercialFeatures)

Blogs

- mechanical-sympathy.blogpost.com
- blog.ragozin.info
- plumbr.eu/blog
- www.javaperformancetuning.com
- psy-lob-saw.blogspot.com
- www.cubrid.org/blog/tags/JVM
- …

Books & Whitepapers

- „Java Performance: The Definitive Guide”
  - Scott Oaks
- „Java Performance”
  - Charlie Hunt, Binu John
- „Oracle JRockit: The Definitive Gude”
  - Marcus Hirt
- „Memory Management in the Java HotSpot™ Virtual Machine”
  - http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf

Q & A

In case of any further questions, please contact me at:
bartlomiej.kaflowski at gmail.com