# Java & der Mythos OutOfMemory

## Typische Probleme mit Java OutOfMemory - die keine sind

Mario Nolte
Managing Consultant - Opitz Consulting

# Inhalt

1. Das Java Memory Model
2. Ausgewählte OutOfMemory Errors
3. Was bringt uns der Java GC - G1 ?

# Das Java Memory Model


32Bit => Java-Heap (Xmx) + C-Heap = 4GB
C-Heap laufen native Operation:
wie z.B. Thread create, JNDI, ..

# Ausgewählte OutOfMemory Errors

- java.lang.OutOfMemoryError: unable to create new native thread
- java.lang.OutOfMemoryError: Perm Space
- java.lang.OutOfMemoryError:  Stack Size
- java.lang.OutOfMemoryError: ChunkPool::XX

# OutOfMemory: unable to create new native thread

- Exception in thread "Thread-0" java.lang.OutOfMemoryError: unable to create new native thread
  at java.lang.Thread.start0(Native Method)
  at java.lang.Thread.start(Thread.java:657)

  - SuMario@gandalf:~$ ulimit –a
    open files (-n) 1024
    stack size (kbytes, -s) 8192
    max user processes (-u) 193027
    virtual memory (kbytes, -v) unlimited

# OutOfMemory: PermSpace

```
Heap
 PSYoungGen      total 1599360K, used 308765K
   eden space 1451264K, 17% used
   from space 148096K, 39% used
   to   space 147264K, 0% used
 PSOldGen        total 3495296K, used 3326686K
   object space 3495296K, 95% used
 PSPermGen       total 1079104K, used 1079078K
   object space 1079104K, 99% used

[Full GC [PSYoungGen: 1441069K->0K(1568896K)] [PSOldGen:
3495288K->1106834K(3495296K)] 4936358K->1106834K(5064192K)
[PSPermGen: 1130420K->1130420K(1250688K)], 5.4386910 secs]
[Times: user=15.35 sys=18.02, real=35.43 secs]
```

PermSpace ist nun Teil des OldSpace

mit Java 1.8 obsolete

(Der Java Programm Fehler – meist Code Issue - bleibt)

# OutOfMemory: pthread_create (stack size X bytes) failed

- Der OutOfMemory error mit “Stack Size x bytes“ kann tritt meist im Zusammenhang mit “Thread create“ oder mit Reflection auf.

- SuMario@gandalf:~$ ulimit –a
  open files (-n) 1024
  stack size (kbytes, -s) 8192

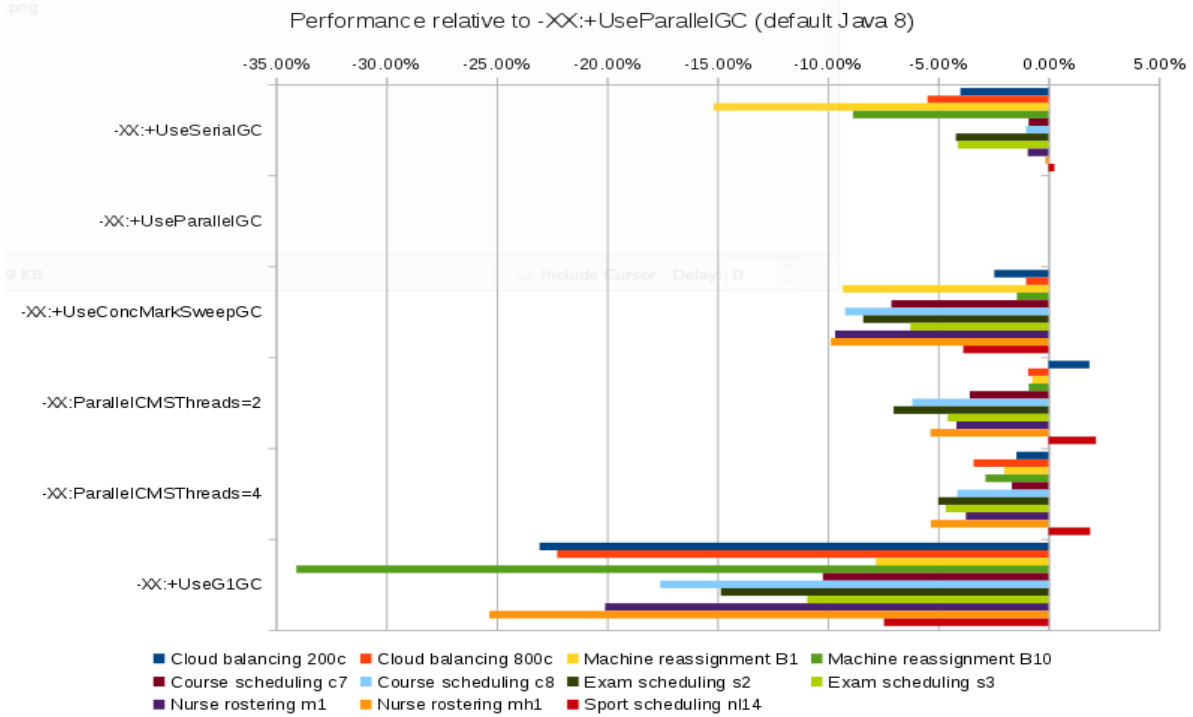
  Java Option –Xss // Anzahl der Prozesse pro User

# OutOfMemory: ChunkPool::XX. Out of swap space?

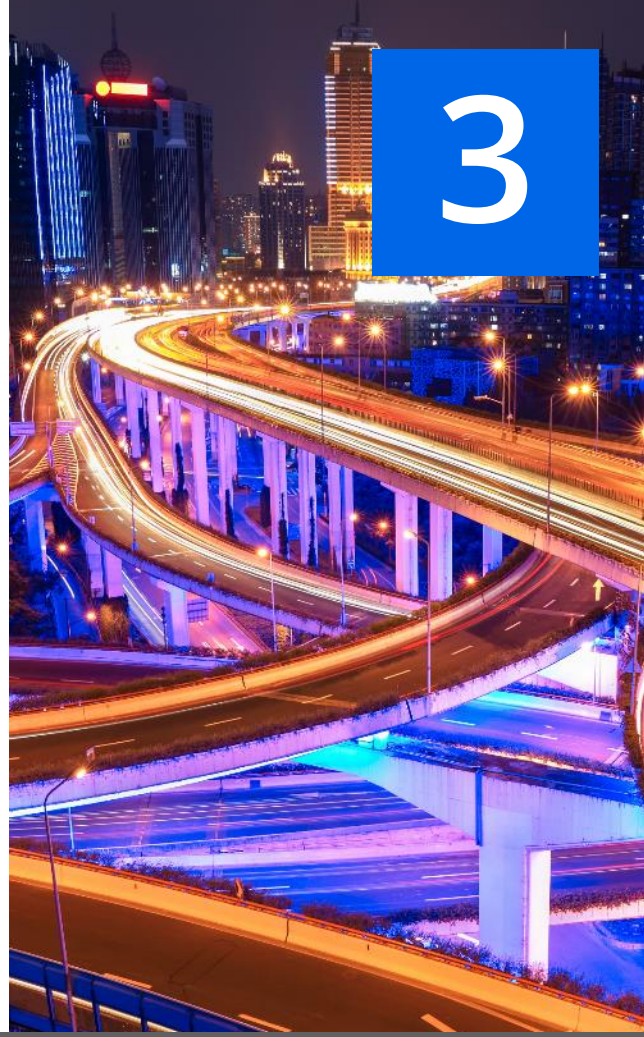
- OutOfMemoryError: requested XXX bytes for Chunk::new. Out of swap space?
- OutOfMemoryError: requested XXX bytes for Chunk::allocate. Out of swap space?

- Hier ist ein Missverhalten zwischen Java-Java-Heap und Java-C-Heap entstanden und tritt momentan typischerweise mit 32Bit Java VM‘s auf.
- Java Option –Xmx im Verhältnis zu groß für den Java C-Heap

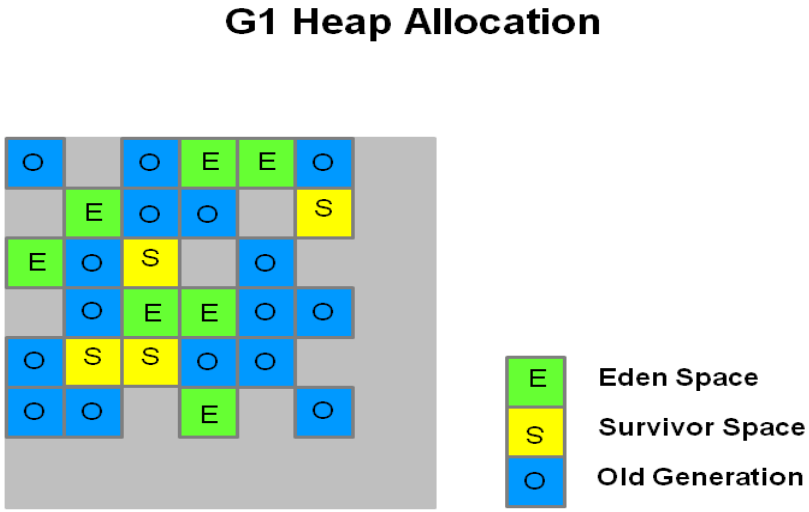
# Was bringt uns der Java GC G1 ?




- Quelle: https://www.optaplanner.org/blog/2015/07/31/WhatIsTheFastestGarbageCollectorInJava8.html



# Java GC1 Heap – Sektionen – damit auch größer 8GB




- Quelle: http://www.oracle.com/technetwork/tutorials/tutorials-1876574.html

# Fragen ?

Mario Nolte

Managing Consultant | Bereich Middleware

Weltenburger Straße 4, 81677 München

Mario.Nolte@Opitz-Consulting.com

WWW.OPITZ-CONSULTING.COM

@OC_WIRE

OPITZCONSULTING

opitzconsulting

opitz-consulting-bcb8-1009116