

## Evacuation YGC 转移暂停

当没有更多的空闲region被提升到老一代或者复制到幸存空间时，并且由于堆已经达到最大值，堆不能扩展，从而发生Evacuation Failure GC，它是非常耗时的。

- 对于成功复制的对象，G1需要更新引用，并且该region被一直引用。
- 对于未成功复制的对象，G1将自动转发它们，并保留这些region。

解决方案：

- 不要过度加一些jvm参数。比如-Xmn,这个参数会限制G1的参数的自动扩展。可以仅使用-Xms, -Xmx和暂停时间目标-XX: MaxGCPauseMillis，删除任何额外的堆大小，例如-Xmn, -XX: NewSize, -XX: MaxNewSize, -XX: SurvivorRatio等。
- 如果问题仍然存在，则增加JVM堆大小（即-Xmx）。
- 如果您无法增加堆大小，并且您注意到marking cycle没有足够早地开始回收老一代，那么请减少-XX: InitiatingHeapOccupancy 默认值是45%。减小该值将提前开始marking cycle。另一方面，如果marking cycle 提前开始并且未收回，请将-XX: InitiatingHeapOccupancyPercent阈值增加到默认值以上。
- 如果并发marking cycle准时开始，但需要很长时间才能完成，那么使用属性'-XX: ConcGCThreads'增加并发标记线程数的数量Workers: 1，单线程执行。
- 如果有大量“空间耗尽（to-space exhausted）”或“空间溢出（to-space overflow）”GC事件，则增加-XX:G1ReservePercent。Java堆的10%。注意：G1 GC将此值限制在50%以内。

```
1 2021-03-25T16:23:12.986-0800: [GC pause (G1 Evacuation Pause) (young), 0.0037463 secs]
2   [Parallel Time: 3.8 ms, GC Workers: 4]
3     [GC Worker Start (ms): Min: 122.9, Avg: 123.0, Max: 123.1, Diff: 0.2]
4     [Ext Root Scanning (ms): Min: 0.0, Avg: 0.1, Max: 0.1, Diff: 0.1, Sum: 0.4]
5     [Update RS (ms): Min: 0.0, Avg: 0.0, Max: 0.1, Diff: 0.1, Sum: 0.2]
6     [Processed Buffers: Min: 0, Avg: 1.5, Max: 3, Diff: 3, Sum: 6]
7     [Scan RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
8     [Code Root Scanning (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
9     [Object Copy (ms): Min: 3.5, Avg: 3.5, Max: 3.5, Diff: 0.1, Sum: 14.0]
10    [Termination (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.1]
11    [Termination Attempts: Min: 1, Avg: 1.0, Max: 1, Diff: 0, Sum: 4]
12    [GC Worker Other (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
13    [GC Worker Total (ms): Min: 3.5, Avg: 3.7, Max: 3.8, Diff: 0.2, Sum: 14.8]
14    [GC Worker End (ms): Min: 126.6, Avg: 126.6, Max: 126.7, Diff: 0.0]
15  [Code Root Fixup: 0.0 ms]
16  [Code Root Purge: 0.0 ms]
17  [Clear CT: 0.0 ms]
18  [Other: 0.3 ms]
19    [Choose CSet: 0.0 ms]
20    [Ref Proc: 0.1 ms]
21    [Ref Enq: 0.0 ms]
22    [Redirty Cards: 0.0 ms]
23    [Humongous Register: 0.0 ms]
24    [Humongous Reclaim: 0.0 ms]
25    [Free CSet: 0.0 ms]
26  [Eden: 30.0M(30.0M)->0.0B(35.0M) Survivors: 4096.0K->5120.0K Heap: 53.5M(512.0M)->30.4M(512.0M)]
27  [Times: user=0.01 sys=0.00, real=0.00 secs]
28
```

## Humongous GC

它是由'G1 Humongous Allocation'造成的。大型对象（Humongous）是大于G1中region大小50%的对象。频繁大型对象分配会导致性  
果region里面包含大量的大型对象，则该region中最后一个具有巨型对象的区域与区域末端之间的空间将不会使用。如果有多个这样的大型对象

未使用的空间可能导致堆碎片化。直到jdk1.8u40之前，这些巨型对象的回收只在full GC期间完成。在较新的JVM中，对这些对象的清理放在段。

解决方案：

- 可以加大region的大小，设置-XX:G1HeapRegionSize=n，但是这个参数需要设置为2的幂次方，最小值是1M，做大值是32M。
- 如果可以的话增加JVM堆大小（即-Xmx -Xms）。

```
1 2021-03-25T16:23:13.335-0800: [GC pause (G1 Humongous Allocation) (young)
2 (initial-mark), 0.0019508 secs]//初始化标记 stw
3 [Parallel Time: 1.1 ms, GC Workers: 4] // cpu核心数
4 [GC Worker Start (ms): Min: 443.2, Avg: 443.2, Max: 443.2, Diff: 0.0]
5 [Ext Root Scanning (ms): Min: 0.2, Avg: 0.2, Max: 0.2, Diff: 0.0, Sum: 0.8]//退出根扫描
6 [Update RS (ms): Min: 0.0, Avg: 0.1, Max: 0.2, Diff: 0.2, Sum: 0.4]//更新rset
7 [Processed Buffers: Min: 1, Avg: 1.5, Max: 3, Diff: 2, Sum: 6]
8 [Scan RS (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0] // 处理rs
9 [Code Root Scanning (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]//code root 扫描
10 [Object Copy (ms): Min: 0.1, Avg: 0.4, Max: 0.8, Diff: 0.7, Sum: 1.6] // 复制
11 [Termination (ms): Min: 0.0, Avg: 0.4, Max: 0.6, Diff: 0.6, Sum: 1.6]
12 [Termination Attempts: Min: 1, Avg: 1.0, Max: 1, Diff: 0, Sum: 4]
13 [GC Worker Other (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
14 [GC Worker Total (ms): Min: 1.1, Avg: 1.1, Max: 1.1, Diff: 0.1, Sum: 4.4]
15 [GC Worker End (ms): Min: 444.3, Avg: 444.3, Max: 444.3, Diff: 0.0]
16 [Code Root Fixup: 0.0 ms] // 修复code root
17 [Code Root Purge: 0.0 ms] // 清除code root
18 [Clear CT: 0.0 ms] // 清除card tables中的dirty card
19 [Other: 0.8 ms] // 其他线程工作消耗的时间
20 [Choose CSet: 0.0 ms] // 选择cset
21 [Ref Proc: 0.6 ms]
22 [Ref Enq: 0.0 ms]
23 [Redirty Cards: 0.0 ms]//从新标记卡牌
24 [Humongous Register: 0.0 ms]//大对象注册
25 [Humongous Reclaim: 0.0 ms]
26 [Free CSet: 0.0 ms]
27 [Eden: 5120.0K(116.0M)->0.0B(105.0M) //伊甸园清理后的内存
28 Survivors: 4096.0K->6144.0K // 幸存者区清理后的内存
29 Heap: 276.8M(512.0M)->274.3M(512.0M)] // 堆中清理后的内存
30 [Times: user=0.00 sys=0.00, real=0.00 secs]
31 2021-03-25T16:23:13.337-0800: [GC concurrent-root-region-scan-start]
32 2021-03-25T16:23:13.337-0800: [GC concurrent-root-region-scan-end, 0.0000855 secs]
33 2021-03-25T16:23:13.337-0800: [GC concurrent-mark-start]
34 2021-03-25T16:23:13.338-0800: [GC concurrent-mark-end, 0.0012343 secs]
35 2021-03-25T16:23:13.338-0800: [GC remark 2021-03-25T16:23:13.338-0800: [Finalize Marking, 0.0006002
36 [Times: user=0.01 sys=0.00, real=0.01 secs]
37 2021-03-25T16:23:13.341-0800: [GC cleanup 281M->281M(512M), 0.0005391 secs]
38 [Times: user=0.00 sys=0.00, real=0.00 secs]
```

## Mixed GC

```
1 2021-03-25T16:23:13.286-0800: [GC pause (G1 Evacuation Pause) (mixed), 0.0049398 secs]
2 [Parallel Time: 4.6 ms, GC Workers: 4]
3 [GC Worker Start (ms): Min: 395.0, Avg: 395.1, Max: 395.2, Diff: 0.2]
4 [Ext Root Scanning (ms): Min: 0.0, Avg: 0.1, Max: 0.1, Diff: 0.1, Sum: 0.4]
```

```

5      [Update RS (ms): Min: 0.2, Avg: 0.5, Max: 1.1, Diff: 0.9, Sum: 1.9]
6      [Processed Buffers: Min: 1, Avg: 4.2, Max: 6, Diff: 5, Sum: 17]
7      [Scan RS (ms): Min: 0.0, Avg: 0.0, Max: 0.1, Diff: 0.0, Sum: 0.1]
8      [Code Root Scanning (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
9      [Object Copy (ms): Min: 3.4, Avg: 3.9, Max: 4.1, Diff: 0.8, Sum: 15.7]
10     [Termination (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
11     [Termination Attempts: Min: 1, Avg: 1.0, Max: 1, Diff: 0, Sum: 4]
12     [GC Worker Other (ms): Min: 0.0, Avg: 0.0, Max: 0.0, Diff: 0.0, Sum: 0.0]
13     [GC Worker Total (ms): Min: 4.4, Avg: 4.5, Max: 4.6, Diff: 0.2, Sum: 18.2]
14     [GC Worker End (ms): Min: 399.6, Avg: 399.6, Max: 399.6, Diff: 0.0]
15     [Code Root Fixup: 0.0 ms]
16     [Code Root Purge: 0.0 ms]
17     [Clear CT: 0.1 ms]
18     [Other: 0.3 ms]
19     [Choose CSet: 0.0 ms]
20     [Ref Proc: 0.1 ms]
21     [Ref Enq: 0.0 ms]
22     [Redirty Cards: 0.0 ms]
23     [Humongous Register: 0.0 ms]
24     [Humongous Reclaim: 0.0 ms]
25     [Free CSet: 0.0 ms]
26     [Eden: 5120.0K(5120.0K)->0.0B(94.0M) Survivors: 22.0M->3072.0K Heap: 322.4M(512.0M)->299.8M(512.
27     [Times: user=0.01 sys=0.00, real=0.01 secs]

```

## 简化输出G1GC日志

```

1 java -XX:+PrintGC -XX:+PrintGCDateStamps -Xmx512m -Xms512m -XX:+UseG1GC week_01_jvm.jmm.GCLogAnalyse
2 正在执行...
3 2021-03-25T16:38:39.422-0800: [GC pause (G1 Evacuation Pause) (young) 32M->9830K(512M), 0.0025886 se
4 2021-03-25T16:38:39.431-0800: [GC pause (G1 Evacuation Pause) (young) 34M->16M(512M), 0.0055173 secs
5 2021-03-25T16:38:39.458-0800: [GC pause (G1 Evacuation Pause) (young) 55M->30M(512M), 0.0115159 secs
6 2021-03-25T16:38:39.489-0800: [GC pause (G1 Evacuation Pause) (young) 74M->45M(512M), 0.0265381 secs
7 2021-03-25T16:38:39.541-0800: [GC pause (G1 Evacuation Pause) (young) 98M->69M(512M), 0.0307247 secs
8 2021-03-25T16:38:39.608-0800: [GC pause (G1 Evacuation Pause) (young) 132M->91M(512M), 0.0455985 sec
9 2021-03-25T16:38:39.746-0800: [GC pause (G1 Evacuation Pause) (young) 220M->127M(512M), 0.0612337 se
10 2021-03-25T16:38:39.840-0800: [GC pause (G1 Evacuation Pause) (young) 193M->150M(512M), 0.0240125 s
11 2021-03-25T16:38:39.921-0800: [GC pause (G1 Evacuation Pause) (young) 292M->201M(512M), 0.0442064 s
12 2021-03-25T16:38:39.977-0800: [GC pause (G1 Humongous Allocation) (young) (initial-mark) 241M->215M
13 2021-03-25T16:38:39.998-0800: [GC concurrent-root-region-scan-start]
14 2021-03-25T16:38:39.999-0800: [GC concurrent-root-region-scan-end, 0.0011378 secs]
15 2021-03-25T16:38:40.000-0800: [GC concurrent-mark-start]
16 2021-03-25T16:38:40.005-0800: [GC concurrent-mark-end, 0.0044920 secs]
17 2021-03-25T16:38:40.005-0800: [GC remark, 0.0069794 secs]
18 2021-03-25T16:38:40.012-0800: [GC cleanup 229M->229M(512M), 0.0034068 secs]
19 2021-03-25T16:38:40.136-0800: [GC pause (G1 Evacuation Pause) (young)-- 406M->290M(512M), 0.0338086
20 2021-03-25T16:38:40.171-0800: [GC pause (G1 Evacuation Pause) (mixed) 298M->276M(512M), 0.0203709 s
21 2021-03-25T16:38:40.192-0800: [GC pause (G1 Humongous Allocation) (young) (initial-mark) 277M->276M
22 2021-03-25T16:38:40.194-0800: [GC concurrent-root-region-scan-start]
23 2021-03-25T16:38:40.194-0800: [GC concurrent-root-region-scan-end, 0.0001599 secs]
24 2021-03-25T16:38:40.194-0800: [GC concurrent-mark-start]
25 2021-03-25T16:38:40.195-0800: [GC concurrent-mark-end, 0.0013783 secs]
26 2021-03-25T16:38:40.195-0800: [GC remark, 0.0015245 secs]
27 2021-03-25T16:38:40.197-0800: [GC cleanup 285M->285M(512M), 0.0006853 secs]

```

```
28 2021-03-25T16:38:40.233-0800: [GC pause (G1 Evacuation Pause) (young) 403M->314M(512M), 0.0117194 s
29 2021-03-25T16:38:40.248-0800: [GC pause (G1 Evacuation Pause) (mixed) 329M->282M(512M), 0.0142372 s
30 2021-03-25T16:38:40.273-0800: [GC pause (G1 Evacuation Pause) (mixed) 307M->290M(512M), 0.0083614 s
31 2021-03-25T16:38:40.281-0800: [GC pause (G1 Humongous Allocation) (young) (initial-mark) 291M->290M
32 2021-03-25T16:38:40.285-0800: [GC concurrent-root-region-scan-start]
33 2021-03-25T16:38:40.285-0800: [GC concurrent-root-region-scan-end, 0.0001336 secs]
34 2021-03-25T16:38:40.285-0800: [GC concurrent-mark-start]
35 2021-03-25T16:38:40.287-0800: [GC concurrent-mark-end, 0.0021998 secs]
36 2021-03-25T16:38:40.288-0800: [GC remark, 0.0035731 secs]
37 2021-03-25T16:38:40.291-0800: [GC cleanup 294M->294M(512M), 0.0005668 secs]
38 2021-03-25T16:38:40.329-0800: [GC pause (G1 Evacuation Pause) (young) 403M->316M(512M), 0.0068002 s
39 2021-03-25T16:38:40.339-0800: [GC pause (G1 Evacuation Pause) (mixed) 331M->297M(512M), 0.0148475 s
40 2021-03-25T16:38:40.355-0800: [GC pause (G1 Humongous Allocation) (young) (initial-mark) 299M->297M
41 2021-03-25T16:38:40.373-0800: [GC concurrent-root-region-scan-start]
42 2021-03-25T16:38:40.373-0800: [GC concurrent-root-region-scan-end, 0.0001872 secs]
43 2021-03-25T16:38:40.373-0800: [GC concurrent-mark-start]
44 2021-03-25T16:38:40.377-0800: [GC concurrent-mark-end, 0.0039452 secs]
45 2021-03-25T16:38:40.379-0800: [GC remark, 0.0060275 secs]
46 2021-03-25T16:38:40.385-0800: [GC cleanup 300M->300M(512M), 0.0007202 secs]
47 执行结束!共生成对象次数:4096
```