Charlie Hunt

May 5, 2015

String Density Impact on SPECjbb2005 on SPARC

This is a report on the impact of the String Density changes on memory footprint, frequency of garbage collection and to a lesser extent the impact on the reported performance score of the SPECjbb2005 benchmark. The report is organized by first offering a summary of the finding and going into more details.

Experiment Goals

An experiment was executed where the SPECjbb2005 workload was run 10 times with a baseline JDK 9 (1.9.0-ea-b61) against a String Density JDK 9 (built from the String Density JDK 9 sandbox repository April 27, 2015). The experiment's goal was to observe the memory footprint reduction obtained from the String Density JDK, and also observe the reduction in the frequency at which GC executed. Secondary goals were to observe the duration of GC events to see how they might differ, and also observe the workload score(s) to quantify any impact there.

The experiment was executed in a non-memory pressure environment so as to reduce the "reduction of time spent in GC" effect on throughput performance. The actual system the tests were executed on is a single socket SPARC Enterprise M3000 SPARC640VII processor with 8 hardware threads, 16 GB of RAM, running Solaris 10.

Summary

In terms of memory footprint reduction, the amount of live data during heaviest load of the benchmark execution, the String Density JDK shows a 21% reduction in live data (retained bytes after a GC) in the Java heap versus a JDK 9 baseline, (363 MB for String Density, 458 MB for baseline JDK 9).

In terms of the frequency of GC, the String Density JDK allowed the application to execute 23% longer amount of time between in GC events during the heaviest load of benchmark execution, (29 seconds for String Density JDK, 23.5 seconds for JDK 9 baseline). Since the workload is run for a specific amount of wall clock time, the total number of GC events observed with the String Density is about 17% fewer, (15 for String Density, 18 for baseline JDK 9).

Secondarily, in terms of the amount of time spent in GC when under peak load, the String Density JDK observed about 12% lower GC time than that of the baseline JDK 9, (430 ms for String Density, 490 ms for baseline JDK 9). And, in terms of the throughput performance score at peak load, the String Density JDK realized about a 10% improvement in throughput performance score versus the baseline JDK 9, (37002 for String Density JDK, 33403 for baseline JDK 9).

Hence, in all metrics observed as goals for the experiment there is an improvement with the String Density JDK.

Command Line Options

The following JVM command line options were used on both JDKs. These are known JDK options that tend to offer the best SPECjbb2005 workload scores when running the HotSpot JVM.

```
-Xmx12g -Xms12g -Xmn8g
-XX:-UseAdaptiveSizePolicy
-XX:MaxTenuringThreshold=15-XX:InitialTenuringThreshold=15
-XX:+UseParallelOldGC
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps -XX:+PrintFlagsFinal
```

Baseline JDK 9 GC Logs

* The first three GC events are ignored since they are induced by the workload at the start of the workload and offer little or no value.

```
44.163: #4: [GC (Allocation Failure)
     [PSYoungGen: 6291456K->121136K(7340032K)] 6292473K->122161K(11534336K),
          0.3799041 secs] [Times: user=0.94 sys=0.91 real=0.38 secs]
66.631: #5: [GC (Allocation Failure)
     [PSYoungGen: 6412592K->176640K(7340032K)] 6413617K->177737K(11534336K),
          0.5489319 secs] [Times: user=1.22 sys=1.44 real=0.55 secs]
88.817: #6: [GC (Allocation Failure)
     [PSYoungGen: 6468096K->176704K(7340032K)] 6469193K->177801K(11534336K),
          0.3095359 secs] [Times: user=1.41 sys=0.46 real=0.31 secs]
110.263: #7: [GC (Allocation Failure)
     [PSYoungGen: 6468160K->232528K(7340032K)] 6469257K->233633K(11534336K),
          0.3683771 secs] [Times: user=1.98 sys=0.42 real=0.37 secs]
132.290: #8: [GC (Allocation Failure)
     [PSYoungGen: 6523984K->288592K(7340032K)] 6525089K->289705K(11534336K),
          0.5339771 secs] [Times: user=2.34 sys=0.84 real=0.53 secs]
154.744: #9: [GC (Allocation Failure)
     [PSYoungGen: 6580048K->288448K(7340032K)] 6581161K->289569K(11534336K),
          0.4253734 secs] [Times: user=2.36 sys=0.46 real=0.43 secs]
177.587: #10: [GC (Allocation Failure)
     [PSYoungGen: 6579904K->344624K(7340032K)] 6581025K->345753K(11534336K),
          0.4977306 secs] [Times: user=2.96 sys=0.45 real=0.50 secs]
200.613: #11: [GC (Allocation Failure)
     [PSYoungGen: 6636080K->400512K(7340032K)] 6637209K->401649K(11534336K),
          0.6702418 secs] [Times: user=3.34 sys=0.92 real=0.67 secs]
```

```
224.247: #12: [GC (Allocation Failure)
     [PSYoungGen: 6691968K->456352K(7340032K)] 6693105K->457497K(11534336K),
          0.7281422 secs] [Times: user=3.81 sys=0.91 real=0.73 secs]
247.986: #13: [GC (Allocation Failure)
     [PSYoungGen: 6747808K->456112K(7340032K)] 6748953K->457265K(11534336K),
          0.5926165 secs] [Times: user=3.78 sys=0.41 real=0.59 secs]
271.748: #14: [GC (Allocation Failure)
     [PSYoungGen: 6747568K->456160K(7340032K)] 6748721K->457313K(11534336K),
          0.4946066 secs] [Times: user=3.93 sys=0.01 real=0.49 secs]
295.222: #15: [GC (Allocation Failure)
     [PSYoungGen: 6747616K->456560K(7340032K)] 6748769K->457713K(11534336K)
          0.4946785 secs] [Times: user=3.89 sys=0.01 real=0.49 secs]
318.794: #16: [GC (Allocation Failure)
     [PSYoungGen: 6748016K->456240K(7340032K)] 6749169K->457393K(11534336K),
          0.4914548 secs] [Times: user=3.90 sys=0.01 real=0.49 secs]
342.323: #17: [GC (Allocation Failure)
     [PSYoungGen: 6747696K->455888K(7340032K)] 6748849K->457041K(11534336K),
          0.4973604 secs] [Times: user=3.92 sys=0.01 real=0.50 secs]
366.011: #18: [GC (Allocation Failure)
     [PSYoungGen: 6747344K->456112K(7340032K)] 6748497K->457265K(11534336K),
          0.4899623 secs] [Times: user=3.89 sys=0.01 real=0.49 secs]
389.408: #19: [GC (Allocation Failure)
     [PSYoungGen: 6747568K->349408K(7340032K)] 6748721K->458082K(11534336K),
          0.6062592 secs] [Times: user=3.70 sys=0.57 real=0.61 secs]
413.523: #20: [GC (Allocation Failure)
     [PSYoungGen: 6640864K->300640K(7340032K)] 6749538K->458258K(11534336K),
          0.4596131 secs] [Times: user=3.10 sys=0.33 real=0.46 secs]
438.343: #21: [GC (Allocation Failure)
     [PSYoungGen: 6592096K->300512K(7340032K)] 6749714K->458194K(11534336K),
          0.3610807 secs] [Times: user=2.87 sys=0.01 real=0.36 secs]
```

String Density JDK 9 GC Logs

* The first three GC events are ignored since they are induced by the workload at the start of the workload and offer little or no value.

```
51.173: #4: [GC (Allocation Failure)
     [PSYoungGen: 6291456K->96192K(7340032K)] 6292367K->97175K(11534336K),
          0.2990217 secs] [Times: user=0.86 sys=0.65 real=0.30 secs]
78.821: #5: [GC (Allocation Failure)
     [PSYoungGen: 6387648K->140192K(7340032K)] 6388631K->141184K(11534336K),
          0.4374951 secs] [Times: user=1.12 sys=1.05 real=0.44 secs]
105.885: #6: [GC (Allocation Failure)
     [PSYoungGen: 6431648K->184448K(7340032K)] 6432640K->185448K(11534336K),
          0.3918349 secs] [Times: user=1.64 sys=0.65 real=0.39 secs]
132.922: #7: [GC (Allocation Failure)
     [PSYoungGen: 6475904K->228816K(7340032K)] 6476904K->229816K(11534336K),
          0.4457606 secs] [Times: user=2.09 sys=0.62 real=0.45 secs]
160.505: #8: [GC (Allocation Failure)
     [PSYoungGen: 6520272K->272912K(7340032K)] 6521272K->273920K(11534336K),
          0.5076889 secs] [Times: user=2.52 sys=0.67 real=0.51 secs]
188.875: #9: [GC (Allocation Failure)
     [PSYoungGen: 6564368K->316976K(7340032K)] 6565376K->317992K(11534336K),
          0.5494661 secs] [Times: user=2.96 sys=0.64 real=0.55 secs]
217.166: #10: [GC (Allocation Failure)
     [PSYoungGen: 6608432K->360624K(7340032K)] 6609448K->361640K(11534336K),
         0.6034299 secs] [Times: user=3.37 sys=0.64 real=0.60 secs]
246.273: #11: [GC (Allocation Failure)
     [PSYoungGen: 6652080K->360992K(7340032K)] 6653096K->362016K(11534336K),
         0.4950298 secs] [Times: user=3.30 sys=0.27 real=0.50 secs]
275.067: #12: [GC (Allocation Failure)
     [PSYoungGen: 6652448K->361344K(7340032K)] 6653472K->362368K(11534336K),
         0.4398912 secs] [Times: user=3.47 sys=0.01 real=0.44 secs]
303.766: #13: [GC (Allocation Failure)
     [PSYoungGen: 6652800K->361248K(7340032K)] 6653824K->362272K(11534336K),
         0.4333022 secs] [Times: user=3.43 sys=0.01 real=0.43 secs]
332.562: #14: [GC (Allocation Failure)
     [PSYoungGen: 6652704K->361440K(7340032K)] 6653728K->362464K(11534336K),
         0.4351580 secs] [Times: user=3.45 sys=0.00 real=0.44 secs]
361.378: #15: [GC (Allocation Failure)
     [PSYoungGen: 6652896K->361088K(7340032K)] 6653920K->362112K(11534336K),
         0.4250735 secs] [Times: user=3.37 sys=0.00 real=0.43 secs]
390.136: #16: [GC (Allocation Failure)
     [PSYoungGen: 6652544K->361440K(7340032K)] 6653568K->362464K(11534336K),
         0.4309616 secs] [Times: user=3.41 sys=0.01 real=0.43 secs]
418.871: #17: [GC (Allocation Failure)
     [PSYoungGen: 6652896K->361424K(7340032K)] 6653920K->362448K(11534336K),
         0.4327096 secs] [Times: user=3.43 sys=0.01 real=0.43 secs]
447.728: #18: [GC (Allocation Failure)
     [PSYoungGen: 6652880K->361440K(7340032K)] 6653904K->362464K(11534336K),
         0.4373009 secs] [Times: user=3.47 sys=0.00 real=0.44 secs]
```