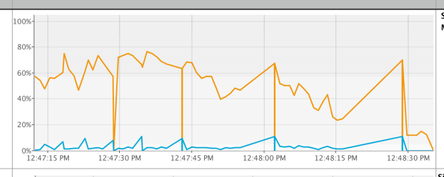
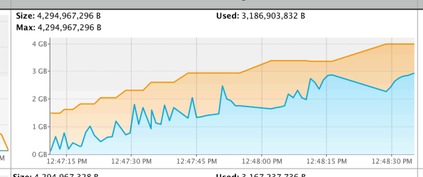
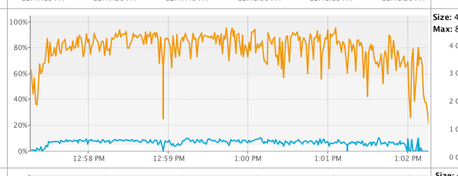
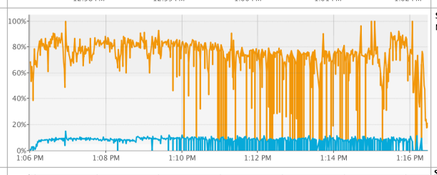
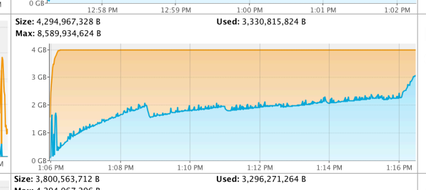
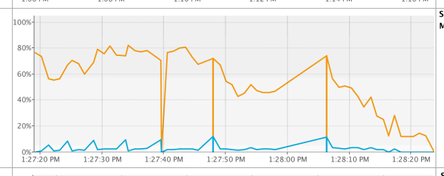
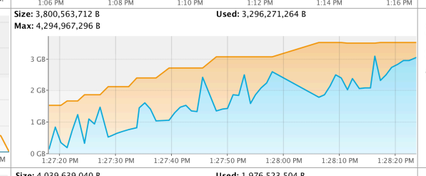
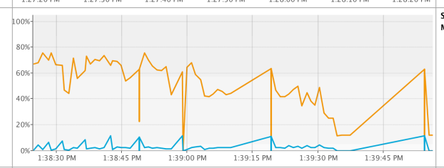
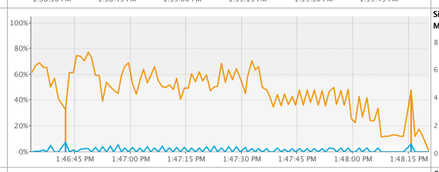
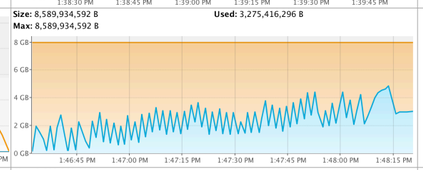
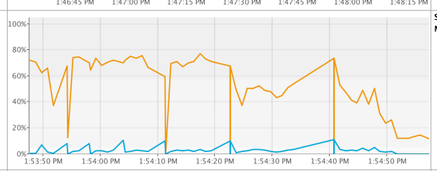
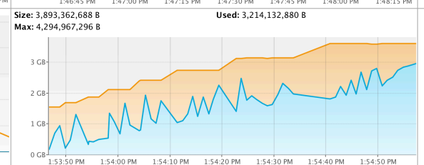
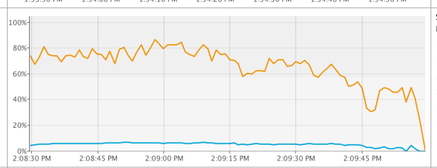
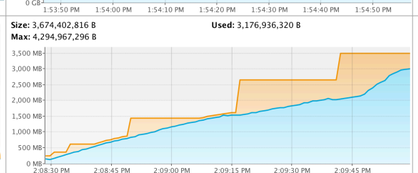
Table 1

Run	JVM arguments	Run time	CPU	Memory	Remarks			
1		1m 27s			Initial run to establish a base case			
2	-XX:+UseG1GC	5m 3s			Much worse in terms of, well, everything, but has interesting sidekick argument that worths a shot			
3	-XX:+UseG1GC -XX:+UseStringDeduplication	10m 33s			Even worse, G1GC option just doesn't work for this task			
4	-XX:+AggressiveOpts	1m 8s			Significantly better than the base case, definitely worth keeping in a final setup			
5	-XX:-BackgroundCompilation	1m 35s			Slightly slower than base case, but consumes less memory			
6	-XX:+AggressiveHeap	1m 51s			Slow memory hog			
7	-XX:+UseParallelGC	1m 14s			Faster than the base case, worth keeping			
8	-XX:+UseConcMarkSweepGC	1m 35s			A bit slow, but very nice memory footprint			
Legend			CPU usage GC activity	Heap size Used heap				
	All of memory and threading options combinations are impossible to test for human being, so I've delegated this task to a fellow Python script, you can see its output in the Appendix No more nice pictures, though							