[JDK-6679764] enable parallel compaction by default Created: 2008-03-25 Updated: 2017-05-16 Resolved: 2012-02-21	
Status:	Closed
Project:	JDK
Component/s:	hotspot
Affects Version/s:	hs13
Fix Version/s:	hs23

Type:	Enhancement	Priority:	P3
Reporter:	Paul Hohensee	Assignee:	John Coomes (Inactive)
Resolution:	Fixed	Votes:	0
Labels:	release-note=done		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Issue Links:	Backport			
	backported by	~~JDK-2220965~~	enable parallel compaction by default	Resolved
	backported by	~~JDK-2220860~~	enable parallel compaction by default	Closed
	CSR			
	Relates			
	relates to	~~JDK-6330863~~	vm/gc/InfiniteList.java fails intermi...	Closed
	relates to	~~JDK-6578152~~	fill_region_with_object has usability...	Resolved
	relates to	~~JDK-6765745~~	par compact - allow young gen spaces ...	Resolved
Subcomponent:	gc			
Resolved In Build:	b13			
CPU:	generic			
OS:	generic			

Imported:	17/Sep/12 3:10 AM
Indexed:	19/Jul/12 7:25 PM
Verification:	Verified

Description

Server-class machine ergonomics was introduced in jdk5. If the machine upon which the jvm is running is powerful enough (currently, at least 2 physical cores plus at least 2gb of memory), the server jvm is invoked using the parallel scavenger rather than the serial scavenger. Currently the old gen collector used is serial mark-sweep-compact. Now that the parallel old gen collector is mature, we should change to using it instead.

Comments

Comment by Devika Gollapudi (Inactive) [2012-04-25]

BT2:EVALUATION

The information has been added to the Java SE 7u4 Release Notes as a JVM feature.

Comment by JPRT Bug Updates (Inactive) [2012-03-22]

BT2:EVALUATION

http://hg.openjdk.java.net/lambda/lambda/hotspot/rev/24cae3e4cbaa

Comment by John Coomes (Inactive) [2012-02-03]

BT2:EVALUATION

After review and some discussion, went with the solution in the first entry: if UseParallelGC is enabled, then enable UseParallelOldGC (unless it has been otherwise disabled via the command line or rc file). Relatedly, if UseParallelOldGC is enabled, then enable UseParallelGC (this is done unconditionally, since UseParallelOldGC requires UseParallelGC).

Comment by JPRT Bug Updates (Inactive) [2012-02-03]

BT2:EVALUATION

http://hg.openjdk.java.net/hsx/hotspot-gc/hotspot/rev/24cae3e4cbaa

Comment by Paul Hohensee [2008-04-14]

BT2:EVALUATION

What I actually had in mind was for server ergo to set UseParallelOldGC, which would set UseParallelGC automatically. I think that if only UseParallelGC is

specified on the command line, the user should still get what he/she gets now, which is serial old + parallel young. If both UseParallelOldGC and UseParallelGC are specified in any order, the same thing happens as now, which is parallel old + parallel young.

Inother words, I think that everything should work exactly as it does now, except that server ergo should set UseParallelOldGC rather than UseParallelGC.

Comment by John Coomes (Inactive) [2008-04-14]

BT2:EVALUATION

The argument-processing semantics are simple, but not completely trivial. The current behavior is that if -XX:+UseParallelOldGC is specified on the command-line, it automatically enables -XX:+UseParallelGC (parallel young gc). This should be preserved. This RFE adds the behavior that when -XX:+UseParallelGC is enabled, either by ergonomics or explicitly on the command line, that -XX:+UseParallelOldGC is also enabled.

The only question is what the arg processing code should do with the command line:

    java -XX:-UseParallelOldGC -XX:+UseParallelGC ...

The relevant point is that -XX:-UseParallelOldGC occurs before -XX:+UseParallelGC. The convention for hotspot argument processing is "last one wins" and one could reasonably interpret -XX:+UseParallelGC as implying -XX:+UseParallelOldGC. Doing so would cause the above command line to enable UseParallelOldGC. While it's reasonable, it's not particularly intuitive or user friendly. I think the best choice for the above command line would be to disable UseParallelOldGC.