

Lösung:

ACHTUNG: die unten stehenden Lösungen sind bezogen auf meine eigenen Messungen. Falls Ihre Messungen anders ausgefallen sind (und dies ist sehr gut möglich), dann kann es sein, dass Sie auch zu völlig anderen Schlussfolgerungen gekommen sind.

Durchsatzoptimierung, Messreihe:

Tuning Parameters	Generation Sizes (MB)				Throughput	Pause (sec)			Number of GCs			Full GC y/n
	Eden	Survivor	Old	Total		Sum	Min	Max	total	young	old	
Optimize for Throughput												
ohne	68.312	8.500	178.688	27.727	65.89%	40.96	0.00020	0.03710	6677	5161	1516	y
32MB	8.5	1.062	21.375	30.938	83.13%	20.26	0.00021	0.06391	4153	3484	669	y
32MB NewRatio = 3	6.5	0.768	24	31.25	78.11%	26.28	0.00022	0.02482	5364	4564	800	y
32MB NewRatio = 1	12.87	1.562	16	30.438	85.16%	17.82	0.00021	0.02492	3042	2293	749	y
32MB NR = 1 SR = 10	13.375	1.312	16	30.688	84.93%	18.09	0.00021	0.05328	3033	2212	821	y
32MB NR = 1 SR = 5	115.5	2.25	16	29.75	86.47%	16.25	0.0002	0.02587	3072	2522	550	y
32MB NR = 1 SR = 2	8	4	16	28	84.72%	18.35	0.00022	0.05065	3833	3409	424	y
64MB NR = 3 SR = 10	13.375	1.312	48	62.688	90.26%	11.69	0.00024	0.02401	2360	2210	150	y
64MB NR = 3 SR = 5	11.5	2.25	48	61.75	89.09%	13.1	0.00023	0.02524	2684	2557	127	y
64MB NR = 3 SR = 2	8	4	48	60	85.74%	17.12	0.00023	0.03098	3730	3640	90	y
64MB NR = 2 SR = 10	17.812	1.75	42.688	62.25	92.51%	8.99	0.00024	0.02529	1772	1660	112	y
64MB NR = 2 SR = 5	15.312	3	42.688	61	92.08%	9.5	0.00024	0.02591	1994	1911	83	y
64MB NR = 2 SR = 2	10.688	5.312	42.688	58.688	89.34%	12.8	0.00026	0.01905	2769	2737	32	y
64MB NR = 1 SR = 12	27.5	2.25	32	61.75	94.96%	6.05	0.00023	0.0257	1155	1066	89	y
64MB NR = 1 SR = 11	27.125	2.438	32	61.562	94.95%	6.07	0.00027	0.02396	1162	1080	82	y
64MB NR = 1 SR = 10	26.75	2.625	32	61.375	94.99%	6.01	0.00028	0.02432	1169	1094	75	y
64MB NR = 1 SR = 8	25.625	3.188	32	60.812	94.99%	6.01	0.0034	0.02382	1197	1133	64	y
64MB NR = 1 SR = 6	24	4	32	60	94.83%	6.2	0.00027	0.02459	1244	1195	49	y
64MB NR = 1 SR = 5	22.875	4.562	32	59.438	94.57%	6.52	0.00022	0.02388	1296	1258	38	y
64MB NR = 1 SR = 2	16	8	32	56	93.11%	8.27	0.00023	0.01903	1843	1843	0	n
1024MB NR = 1	409.625	51.188	512	972.812	99.69%	0.38	0.00088	0.01468	72	72	0	n

Die gelb markierten Zahlen sind Momentaufnahmen eines dynamisch verwalteten Heaps. Daher machen diese Zahlen zusammen keinen Sinn. Alle anderen Angaben in MB bedeuten, dass die JVM exakt auf diesen Wert eingestellt wurde.

Fragen aus dem Tutorialtext:

Which tunings are important and helpful?

Eine Grösse für den Heap vorzugeben war in diesem Fall sehr wichtig, da es dann zu keiner dynamischen Anpassung des Speichers kommt.

Which did only provide minor improvements?

Die Survivor-Spaces zu manipulieren brachte um das Optimum herum kaum mehr etwas.

If the test program were your product, which setting would you recommend for your customers?

Genügend Speicher kaufen und mit sehr viel Young-Generation arbeiten.

Pausenzeitoptimierung, Messreihe:

Tuning Parameters	Generation Sizes (MB)				Throughput	Pause (sec)			Number of GCs			Full GC y/n
	Eden	Survivor	Old	Total		Sum	Min	Max	total	young	old	
64MB NR = 1 SR = 2	16	8	32	56	93.11%	8.27	0.00023	0.01903	1843	1843	0	n
64MB NR = 1	25.625	3.188	32	60.812	94.92%	6.1	0.00027	0.03328	1193	1129	64	y
64MB NR = 1 CMS	25.625	3.188	32	60.812	95.55%	5.35	0.00012	0.03141	0			n
64MB NR = 2 CMS	17.062	2.125	42.688	61.875	93.37%	7.96	0.00012	0.01459	0			n
64MB NR = 3 CMS				62.438	90.88%	10.96	0.00011	0.02144	0			n

Fragen aus dem Tutorialtext:

Which tunings are important, and which have only minor effect?

Wiederum sind die Grössen der Survivor-Spaces weniger wichtig als die Grösse der Young-Generation.

What about the default configurations? How do they compare to your tuning efforts?

Schlecht, lieber nicht verwenden.

How do the tunings for throughput and minimum pause differ? How do the corresponding results differ?

Die Resultate waren schneller eindeutig als beim Durchsatzoptimieren.

How was the throughput, when you tuned for minimum pause times and the minimum pause times when you tuned for throughput? What is a good compromise?

Auf minimale Pausenzeiten optimieren hat in diesem Falle am meisten gebracht, da dabei oft auch sehr gute Durchsätze herausgekommen sind. Dies ist offenbar den optimierten und parallelen GCs zu verdanken.