

Distribuição não cíclica

Tempo mínimo

$n_{\text{comm_sz}}$	1	2	4	8	16
1000000000	4.795452	2.807694	1.358758	0.700628	0.317507
	4.786036	2.799903	1.361447	0.634377	0.368112
	4.795945	2.884835	1.379414	0.641625	0.398468
	4.792478	2.830811	1.366540	0.658877	0.361363
	0.005584	0.046948	0.011230	0.036339	0.040900
2000000000	9.579308	5.592917	2.754716	1.336810	0.676934
	9.623440	5.570635	2.760277	1.328466	0.669340
	9.394254	5.590585	2.758311	1.329319	0.692249
	9.532334	5.584712	2.757768	1.331532	0.679507
	0.121600	0.012247	0.002820	0.004591	0.011869

Speedup

$n_{\text{comm_sz}}$	1	2	4	8	16
1000000000	----	1.692970	3.507017	7.273710	13.262244
2000000000	----	1.706862	3.456539	7.158924	14.028301

Eficiência

$n_{\text{comm_sz}}$	1	2	4	8	16
1000000000	----	0.846485	0.876754	0.909214	0.828890
2000000000	----	0.853431	0.864135	0.894865	0.876769

Informações de Hardware:

CPU AMD Ryzen 9 5900X - 12 core @ 3.7GHz

Tempo máximo

$n_{\text{comm_sz}}$	1	2	4	8	16
1000000000	4.795452	2.807701	1.564272	0.807801	0.618491
	4.786036	2.799909	1.563530	0.870643	0.627301
	4.795945	2.884843	1.570570	0.816833	0.640634
	4.792478	2.830818	1.566124	0.831759	0.628809
	0.005584	0.046949	0.003868	0.033976	0.011148
2000000000	9.579308	5.592924	3.120697	1.643041	1.179188
	9.623440	5.570640	3.166439	1.663674	1.189517
	9.394254	5.590593	3.147444	1.593172	1.133791
	9.532334	5.584719	3.144860	1.633296	1.167499
	0.121600	0.012248	0.022980	0.036247	0.029645

Speedup

$n_{\text{comm_sz}}$	1	2	4	8	16
1000000000	----	1.692966	3.060088	5.761859	7.621522
2000000000	----	1.706860	3.031084	5.836257	8.164749

Eficiência

$n_{\text{comm_sz}}$	1	2	4	8	16
1000000000	----	0.846483	0.765022	0.720232	0.476345
2000000000	----	0.853430	0.757771	0.729532	0.510297

A presença do -O3 contribuiu para diminuir o tempo de execução do programa (mesmo o serial). Observa-se uma piora no speedup (calculado com os tempos de máximo), já que, como temos processos em blocos (não cíclico), os processos finais estão com tempos maiores que os iniciais. Além disso, observa-se que, com o aumento do número de processos, temos também uma maior diferença entre os tempos de mínimo e máximo. Além disso, percebe-se que o código é fracamente escalável, uma vez que aumentamos o número de processos (n) e o tamanho do problema (comm_sz) na mesma proporção, temos uma eficiência bem próxima