

Relatório feito por Anderson Aparecido do Carmo Frasão para a matéria de programação paralela do curso de Ciência da computação, da Universidade Federal do Paraná, ministrada pelo professor Wagner Zola, no segundo período de 2022.

1) Implementação:

O algoritmo `prefixSumPth.c` tem como base o algoritmo `reduceSumPth.c`, disponibilizado pelo professor, para ajustar o propósito do algoritmo, a função `reducePartialSum` foi modificada, se tornando a `prefixPartialSum` e o algoritmo segue o fluxo de ideias as quais o professor explanou em aula:

Exemplo para vetor de 8 números e fazendo com 3 threads:

Exemplo de vetor In (de entrada):

vetor In: [3 1 7 . 0 4 1 . 6 3]

Cada thread faz a soma de prefixos na sua faixa de números (chunk)

Nesse caso obtemos o seguinte vetor Out (de saída)

Out: [3 4 11 . 0 4 5 . 6 9].

Declaramos um vetor global de máximos por thread

Nesse caso teremos um vetor de 3 posições

`maximosPorThread`: [11 5 9]

A thread 0 (somente) faz: a soma de prefixos nos `maximosPorThread` (global), ficando:

`maximosPorThread`: [11 16 25]

Cada thread soma em cada célula de sua faixa o máximo global da thread anterior

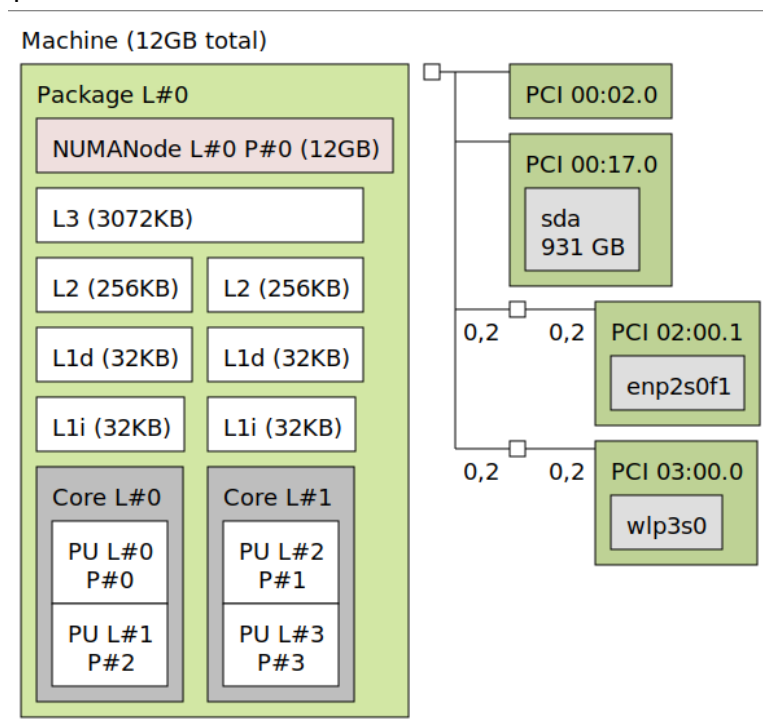
sendo que o máximo global da thread 0 é 0, assim:

vetor Out: [3 4 11 . 0 4 5 . 6 9].

cada thread soma: 0 0 0 . 11 11 11 . 16 16

vetor Out: [3 4 11 . 11 15 16 . 22 25]

2) descrição do processador:



3) Medições:

Todas as medições foram feitas seguindo as mesmas métricas do algoritmo reduceSumPht.c.

4) planilha:

Tempo e Throughput para 1, 2, 3, 4, 5, 6, 7, 8 Threads e sequencialmente (ao final).

Usando 1 thread(s) para prefixar 5000000 elementos	Usando 2 thread(s) para prefixar 5000000 elementos
Throughput (OP/s):	Throughput (OP/s):
370680396,470856	569706365,363403
336995076,906321	492882918,521426
325913159,786749	548171082,002009
314608169,682029	572795949,920221
319792365,213114	537765554,008225
368779156,602069	564573934,118287
368162188,108641	588383843,732782
372073253,18677	582421358,556061
365523181,004959	589066987,40186
322612281,97862	573521169,928129
Throughput médio: 346513922,894013	Throughput médio: 561928916,35524
Tempo total (s):	Tempo total (s):
0,013489	0,008776
0,014837	0,010144
0,015342	0,009121
0,015893	0,008729
0,015635	0,009298
0,013558	0,008856
0,013581	0,008498
0,013438	0,008585
0,013679	0,008488
0,015498	0,008718
TEMPO MÍNIMO (s): 0,013438	TEMPO MÍNIMO (s): 0,008488
TEMPO MÉDIO (s): 0,014495	TEMPO MÉDIO (s): 0,0089213
ACELERAÇÃO: 0,968	ACELERAÇÃO: 1,572719223

Usando 3 thread(s) para prefixar 5000000 elementos	Usando 4 thread(s) para prefixar 5000000 elementos
Throughput (OP/s):	Throughput (OP/s):
571414400,351443	430617340,770841
566075823,139346	471760332,164563
567107857,676344	420205320,723812
552410315,356686	398587787,525286
567118471,048602	427552656,102468
570449744,860647	454357639,619332
552642210,038304	459334053,814108
500739141,046098	423486672,493279
394468230,002118	411353217,160603
418225147,827952	441480169,991057
Throughput médio: 526065134,134754	Throughput médio: 433873519,036535
Tempo total (s):	Tempo total (s):
0,00875	0,011611
0,008833	0,010599
0,008817	0,011899
0,009051	0,012544
0,008816	0,011694
0,008765	0,011005
0,009047	0,010885
0,009985	0,011807
0,012675	0,012155
0,011955	0,011326
TEMPO MÍNIMO (s): 0,008750	TEMPO MÍNIMO (s): 0,010599
TEMPO MÉDIO (s): 0,0096694	TEMPO MÉDIO (s): 0,0115525
ACELERAÇÃO: 1,45104143	ACELERAÇÃO: 1,214516338
Usando 5 thread(s) para prefixar 5000000 elementos	Usando 6 thread(s) para prefixar 5000000 elementos
Throughput (OP/s):	Throughput (OP/s):
590757062,264377	451169069,29235
492102783,736869	474192769,793162
711932066,872633	462281696,714361
777775651,857663	975270076,665981
560012947,499346	634740328,366406
561381258,37651	1141417000,72138
418623942,911751	938947211,073267
442824807,789678	927001934,838438
461982874,849209	1106564361,10294
398150511,245164	1118647063,49553
Throughput médio: 541554390,74032	Throughput médio: 823023151,20638
Tempo total (s):	Tempo total (s):
0,008464	0,011082
0,01016	0,010544
0,007023	0,010816
0,006429	0,005127
0,008928	0,007877
0,008907	0,004381
0,011944	0,005325
0,011291	0,005394
0,010823	0,004518
0,012558	0,00447
TEMPO MÍNIMO (s): 0,006429	TEMPO MÍNIMO (s): 0,004381
TEMPO MÉDIO (s): 0,0096527	TEMPO MÉDIO (s): 0,0069534
ACELERAÇÃO: 1,453551856	ACELERAÇÃO: 2,017818621

Usando 7 thread(s) para prefixar 5000000 elementos		Usando 8 thread(s) para prefixar 5000000 elementos
Throughput (OP/s):		Throughput (OP/s):
402618307,376531		407480162,235782
489512392,789443		387127184,907119
412003643,760225		437521990,94907
443116080,904134		454393687,053979
714066801,806189		431938462,936613
645344235,004426		734276820,30528
435393295,117413		813273138,108743
954040255,537774		670636207,085486
1022213726,49036		814424964,666173
975975961,321682		693208703,6512
Throughput médio: 649428470,010818		Throughput médio: 584428132,189945
Tempo total (s):		Tempo total (s):
0,012419		0,012271
0,010214		0,012916
0,012136		0,011428
0,011284		0,011004
0,007002		0,011576
0,007748		0,006809
0,011484		0,006148
0,005241		0,007456
0,004891		0,006139
0,005123		0,007213
TEMPO MÍNIMO (s): 0,004891		TEMPO MÍNIMO (s): 0,006139
TEMPO MÉDIO (s): 0,0087542		TEMPO MÉDIO (s): 0,009296
ACELERAÇÃO: 1,602739257		ACELERAÇÃO: 1,509326592

Prefixando 5000000 elementos Sequencialmente

Throughput (OP/s):

332155465,764338

362106903,082884

385860765,075464

416353534,112886

359524501,522874

374114032,501082

320962847,459133

324866700,695371

327409148,545007

385584833,868846

Throughput médio: 358893873,262788

Tempo total (s):

0,015053

0,013808

0,012958

0,012009

0,013907

0,013365

0,015578

0,015391

0,015271

0,012967

TEMPO MÍNIMO (s): 0,012009

TEMPO MÉDIO (s): 0,0140307

5) Gráfico:

