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.

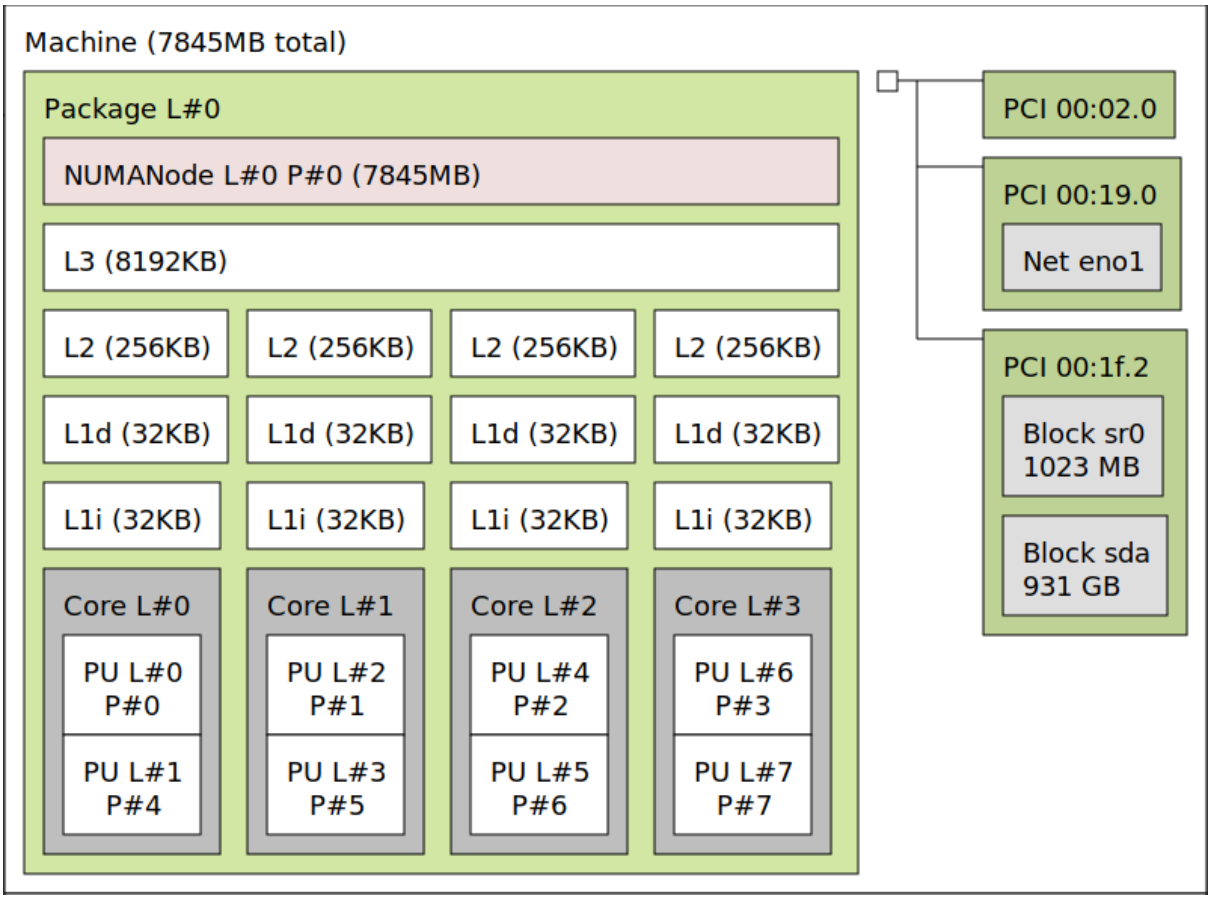
1) Implementação:

O algoritmo prefixSumPth-v2.c tem como base o algoritmo prefixSumPth.c, feito no trabalho 1, para ajustar o propósito do algoritmo, a função prefixPartialSum foi modificada, 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 InputVector (de entrada):
vetor InputVector: [3 1 7 . 0 4 1 . 6 3]
Cada thread faz a soma de valores na sua faixa de números (chunk), produzindo um valor na sua posição do vetor global partialSum
Ou seja, para esse exemplo fica:
A thread 0 faz:
- calcula myPartialSum = 3+1+7
- armazena no vetor global assim: PartialSum[0] = myPartialSum;
A thread 1 faz:
- calcula myPartialSum = 0+4+1
- armazena no vetor global assim: PartialSum[1] = myPartialSum;
A thread 2 faz:
- calcula myPartialSum = 6+3
- armazena no vetor global assim: PartialSum[2] = myPartialSum;
O vetor global PartialSum fica:
PartialSum: [11 5 9]
Após a barreira cada thread faz:
a) calcula a soma do seu prefixo lendo o vetor global PartialSum
e calculando seu "prefixo" na variável local myPrefixSum
b) produz a soma de prefixos correta na usa faixa,
lendo a faixa do vetor de entrada e escrevendo no vetor
de saída, e usando o seu valor calculado em na soma myPrefixSum
Assim temos:
A thread 0 faz:
a) myPrefixSum = 0 (sempre!)
b) produz sua faixa no vetor de saída com APENAS UM "for"
sua faixa fica [3 4 11 ...]
A thread 1 faz:
a) myPrefixSum = 11
b) produz sua faixa no vetor de saída com APENAS UM "for"
sua faixa fica [... 11 15 16 ...]
A thread 2 faz:
a) myPrefixSum = 16 (porque? somou 11+5 do vetor PartialSum)
b) produz sua faixa no vetor de saída com APENAS UM "for"
sua faixa fica [... 22 25]
Nesse caso obtemos o seguinte vetor de saída
OutputVector: [3 4 11 . 11 15 16 . 22 25]

2) descrição do processador:

```
Arquitetura:                         x86_64
Modo(s) operacional da CPU:          32-bit, 64-bit
Ordem dos bytes:                     Little Endian
Tamanhos de endereço:                39 bits physical, 48 bits virtual
CPU(s):                              8
Lista de CPU(s) on-line:             0-7
Thread(s) per núcleo:                2
Núcleo(s) por soquete:               4
Soquete(s):                          1
Nó(s) de NUMA:                       1
ID de fornecedor:                    GenuineIntel
Família da CPU:                      6
Modelo:                              60
Nome do modelo:                      Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
Step:                                3
CPU MHz:                             800.000
CPU MHz máx.:                        3900,0000
CPU MHz mín.:                        800,0000
BogoMIPS:                            6784.83
cache de L1d:                        128 KiB
cache de L1i:                        128 KiB
cache de L2:                         1 MiB
cache de L3:                         8 MiB
CPU(s) de nó0 NUMA:                  0-7
Vulnerability Itlb multihit:         KVM: Mitigation: VMX unsupported
Vulnerability L1tf:                  Mitigation; PTE Inversion
Vulnerability Mds:                   Vulnerable: Clear CPU buffers attempted, no microcode; SMT
vulnerable
Vulnerability Meltdown:              Mitigation; PTI
Vulnerability Mmio stale data:       Unknown: No mitigations
Vulnerability Retbleed:              Not affected
Vulnerability Spec store bypass:     Vulnerable
Vulnerability Spectre v1:            Mitigation; usercopy/swapgs barriers and __user pointer
sanitization
Vulnerability Spectre v2:            Mitigation; Retpolines, STIBP disabled, RSB filling,
PBRSB-eIBRS Not affected
Vulnerability Srbds:                 Vulnerable: No microcode
Vulnerability Tsx async abort:       Not affected
Opções:                              fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp
lm constant_tsc arch_perfmon
                                     pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni
pclmulqdq dtes64 monitor ds_cpl smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1
sse4_2 x2apic movbe pop
                                     cnt aes xsave avx f16c rdrand lahf_lm abm cpuid_fault epb
invpcid_single pti fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid xsaveopt dtherm
ida arat pln pts
```



3) Medições:

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

4) Gráfico: