

Journal

DiDomenico

November 24, 2016

Contents

1	Dissertação - API HPSM	1
1.1	Orientações	1
1.1.1	<2016-08-23 Ter>	1
1.1.2	<2016-08-30 Ter>	1
1.1.3	<2016-09-08 Qui>	2
1.1.4	<2016-09-23 Sex>	2
1.1.5	<2016-10-20 Qui>	2
1.2	Comandos	2
1.2.1	XKaapi with GCC 4.9 compiler	2
1.3	Texto	2
1.3.1	Estrutura:	2
1.3.2	Tamanho:	3
1.4	Experimentos	3
1.4.1	Calibragem blocos digitalis-IDCIN-2	3
1.4.2	Execuções digitalis-IDCIN-2 - NBody, Hotspot e CFD	9
1.4.3	Rastros	17
1.4.4	Perfmodel StarPU+OpenMP	28
1.4.5	Perfmodel StarPU+Kaapi	30
1.4.6	Overhead da API	31

1 Dissertação - API HPSM

1.1 Orientações

1.1.1 <2016-08-23 Ter> ORIENTACAO

1. Apenas passado o acesso a IDCIN-2.

1.1.2 <2016-08-30 Ter> ORIENTACAO

1. Atualizar gráficos do artigo com escalabilidade forte (variando o número de threads) e fraca (variando o tamanho das entradas).
2. Gerar gráficos com o R.
3. Ver para substituir o ljforces por outro benchmark - SNAP;
4. Testar StarPU com OpenMP - dividindo as tarefas CPU para muitas threads;
5. Métricas de software - lines of code;
6. Criar journal da dissertação.

1.1.3 <2016-09-08 Qui>

ORIENTACAO

1. Definido que a defesa será em dezembro. Prazos:

- (a) Experimentos: até o final de outubro;
- (b) Texto: mês de novembro.

1.1.4 <2016-09-23 Sex>

ORIENTACAO

1. Em conversa com o professor ficou definido:

- (a) Gerar traços das execuções com GPUs e o máximo de threads, a fim de identificar o motivo do melhor desempenho ser obtido com uma thread a menos.
- (b) Testar outros escalonadores além do dmda, como dm, dmdar, dmdas;
- (c) Realizar testes juntando starPU e OpenMP. A princípio utilizar o escalonador pheft.

1.1.5 <2016-10-20 Qui>

ORIENTACAO

1. Rodar a versão StarPU+OMP com Kaapi:

- (a) Ver como compila a versão StarPU com o Kaapi 5.4 ou o Kaapi com a versão 4.9;
- (b) Verificar afinidade, garantindo que o Kaapi ta processando nos núcleos corretos (socket 0 ou 1);

2. Gerar gráficos com o tempo (ao invés de apenas speedup);

3. Verificar se o tempo serial é proporcional quando se aumenta as entradas;

1.2 Comandos

1.2.1 XKaapi with GCC 4.9 compiler

Configure: `##beginsrc sh ac_cvfuncmalloc0nonnull=yes ac_cvfuncrealloc0nonnull=yes \ CFLAGS="-std=gnu99" AM_CFLAGS="-std=gnu99" ../configure \ --prefix=/home/jvlima/daniel/source/xkaapi/install \ --disable-offload --without-cuda --enable-mode=release \ --enable-libkomp --without-perfcounter --with-numa ##endsrc` Servidor França :noexport: Para ver agendamentos:

- chandler

Senha agenda on-line: jvlima xabunfo16@

1.3 Texto

1.3.1 Estrutura:

1. Introdução
2. Programação para aceleradores
 - (a) Arquiteturas Distribuida, compartilhada, aceleradores (GPU e Phi)
 - (b) Ferramentas de programação paralela Clássicas (MPI, OpenMP e CUDA)
 - (c) Ferramentas multi-CPU e multi-GPU
 - (d) Trabalhos relacionados
3. API proposta
4. Resultados experimentais
5. Conclusão
6. Apêndice
 - (a) C++

1.3.2 Tamanho:

Entre 60 a 100 páginas

1.4 Experimentos

1.4.1 Calibragem blocos digitalis-IDCIN-2

1. NBody: existe grande mudança de desempenho a medida que o tamanho do bloco da tarefa é aumentado durante o processamento na GPU. Atráves das figuras percebe-se que com bloco de 2048 há um equilíbrio entre CPU e GPU. Com 4096, as tarefas são processadas somente na GPU. Variando-se o tamanho da entrada, o desempenho permaneceu estavel.

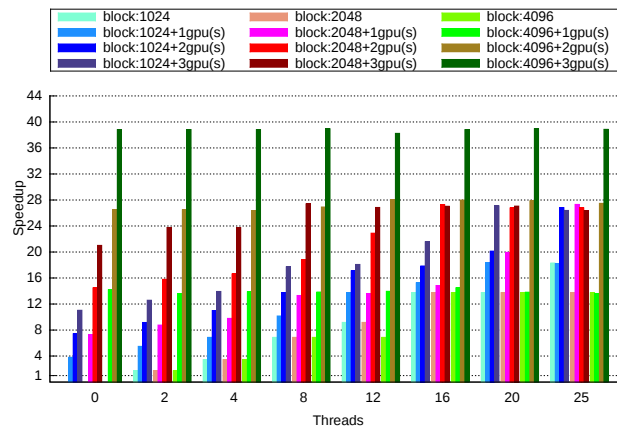


Figure 1: NBody: speedup sobre a execução serial variando threads e tamanho do bloco - Size: 65536.

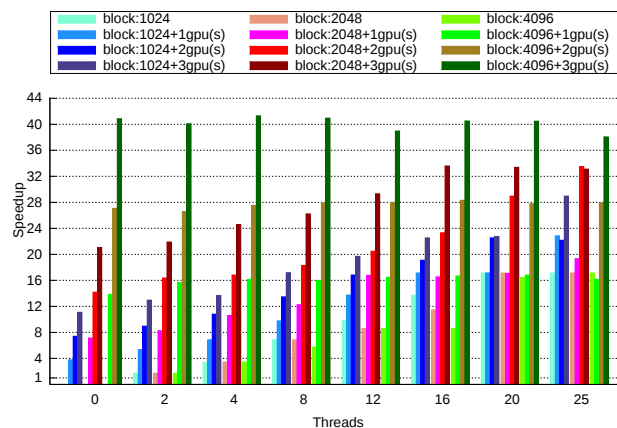


Figure 2: NBody: speedup sobre a execução serial variando threads e tamanho do bloco - Size: 81190.

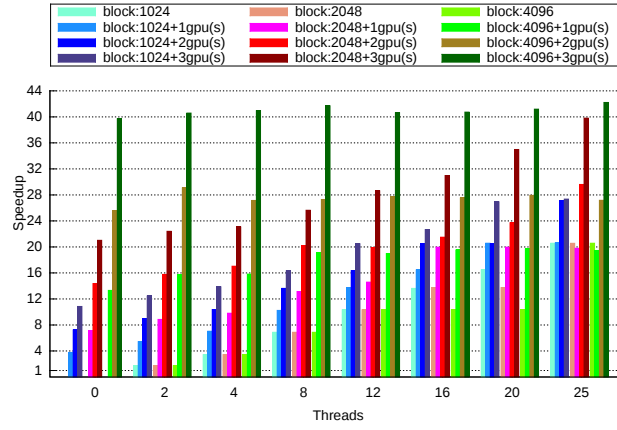


Figure 3: NBody: speedup sobre a execução serial variando threads e tamanho do bloco - Size: 98304.

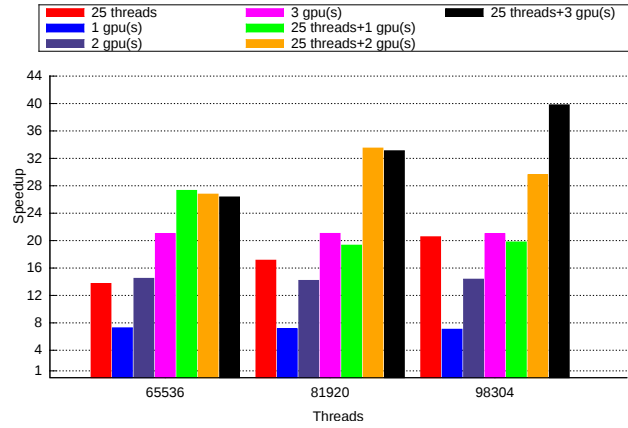


Figure 4: NBody: speedup sobre a execução com máxima configuração variando tamanho da entrada - BS: 2048.

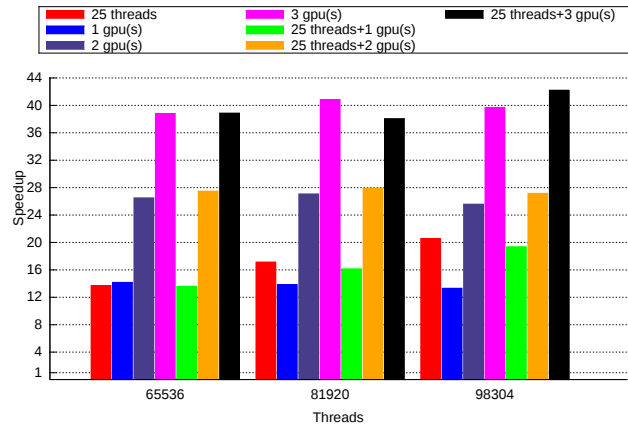


Figure 5: NBody: speedup sobre a execução com máxima configuração variando tamanho da entrada - BS: 4096.

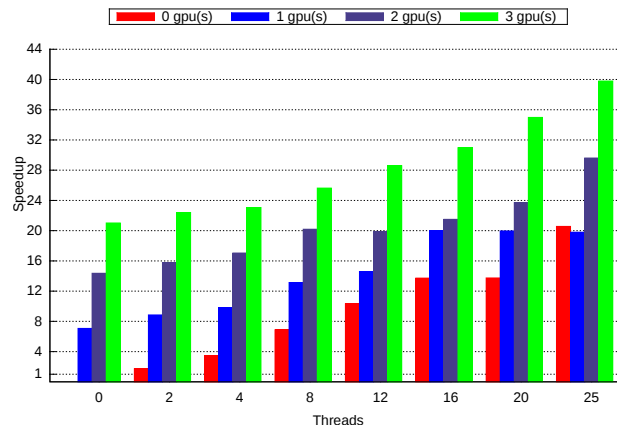


Figure 6: NBody: speedup sobre a execução com gpus variando número de threads - BS: 2048.

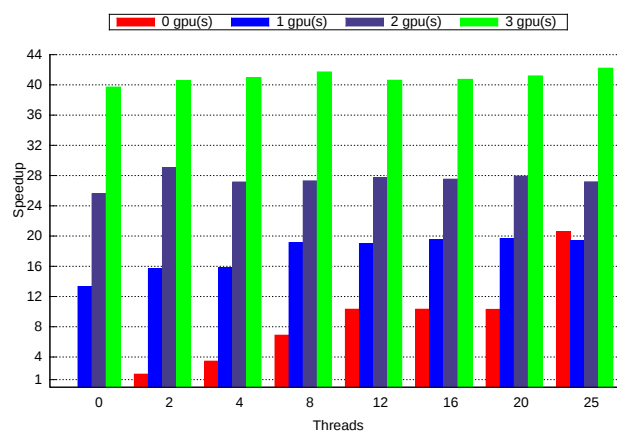


Figure 7: NBody: speedup sobre a execução com gpus variando número de threads - BS: 4096.

2. Hotspot: com o hotspot, a variação do bloco não afeta o desempenho da GPU, indicando que a mesma já está com desempenho máximo. Todavia, blocos maiores acabaram prejudicando o processamento na CPU. Ao variar o tamanho da entrada, ocorreu um ganho de desempenho, porém mínimo.

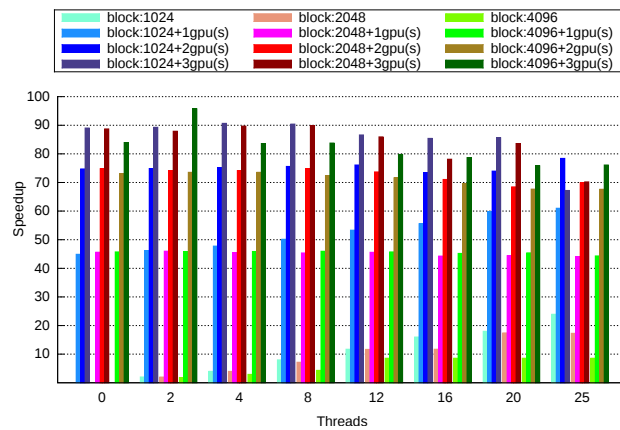


Figure 8: Hotspot: speedup sobre a execução serial variando threads e tamanho do bloco - Size: 12288 x 12288.

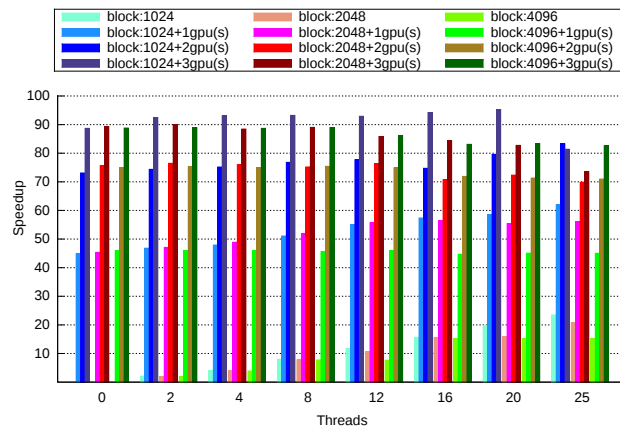


Figure 9: Hotspot: speedup sobre a execução serial variando threads e tamanho do bloco - Size: 16384 x 16384.

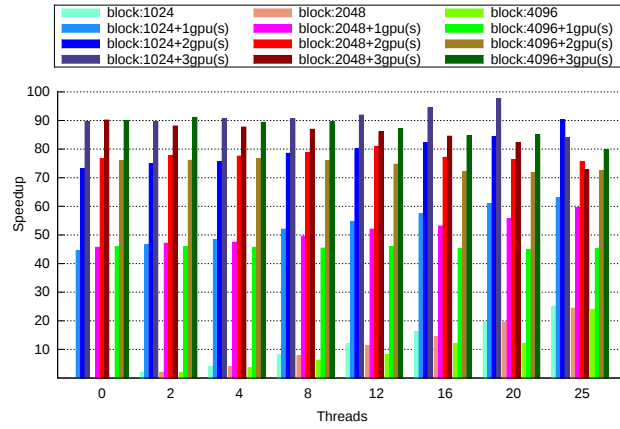


Figure 10: Hotspot: speedup sobre a execução serial variando threads e tamanho do bloco - Size: 20480 x 20480.

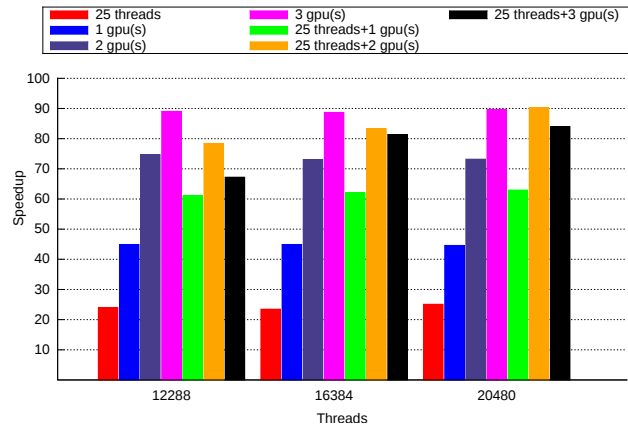


Figure 11: Hotspot: speedup sobre a execução com máxima configuração variando tamanho da entrada - BS 1024 x 1024.

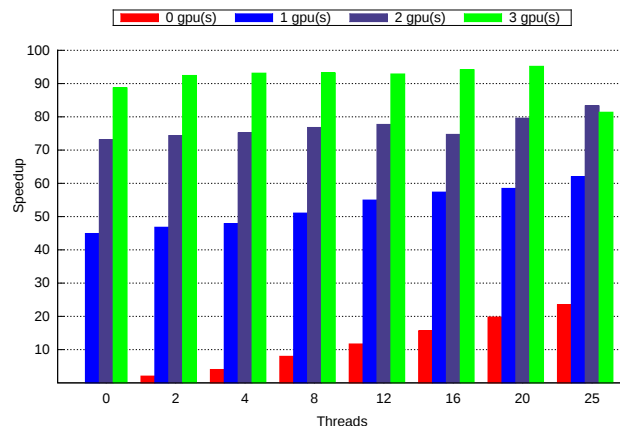


Figure 12: Hostspot: speedup sobre a execução com máxima configuração variando threads - Size: 16384 x 16384 BS: 1024 x 1024.

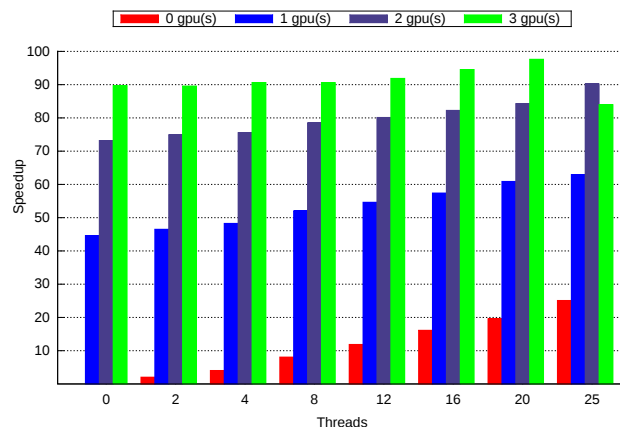


Figure 13: Hostspot: speedup sobre a execução com máxima configuração variando threads - Size: 20480 x 20480 BS: 1024 x 1024.

1.4.2 Execuções digitalis-IDCIN-2 - NBody, Hotspot e CFD

1. Speedup by threads:

Sizes: CFD: 131072; Hostspot: 16384 x 16384; NBody: 98304
Block sizes: CFD: 2048; Hostspot: 1024 x 1024; NBody: 2048

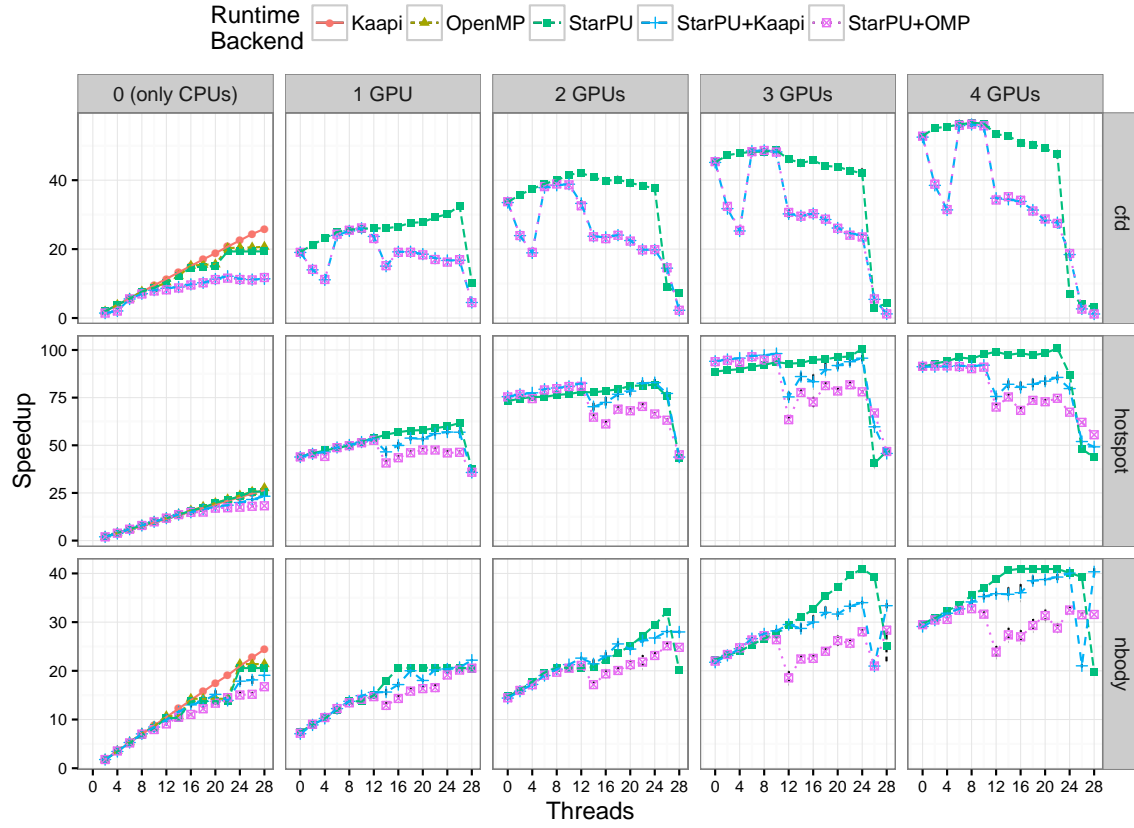


Figure 14: Speedup over sequential version by threads and GPUs.

2. Time by threads:

Sizes: CFD: 131072; Hostspot: 16384 x 16384; NBody: 98304
Block sizes: CFD: 2048; Hostspot: 1024 x 1024; NBody: 2048

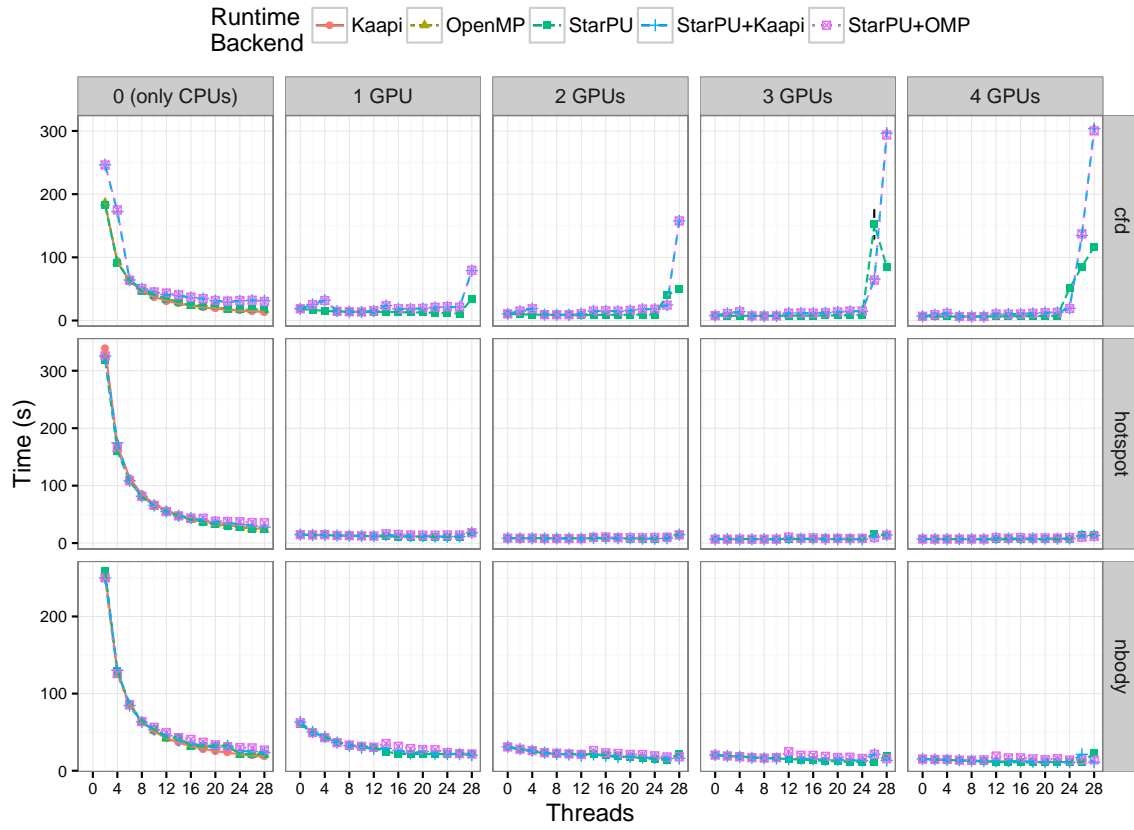


Figure 15: Time by threads and GPUs.

3. Speedup by size (max configuration):

Sizes:

CFD: 98304, 114688, 131072, 147456, 163840;

Hotspot: 12288N, 14336N, 16384N, 18432N, 20480N;

NBody: 65536, 81920, 98304, 114688, 131072;

Block sizes:

CFD: 2048;

Hotspot: 1024 x 1024;

NBody: 2048

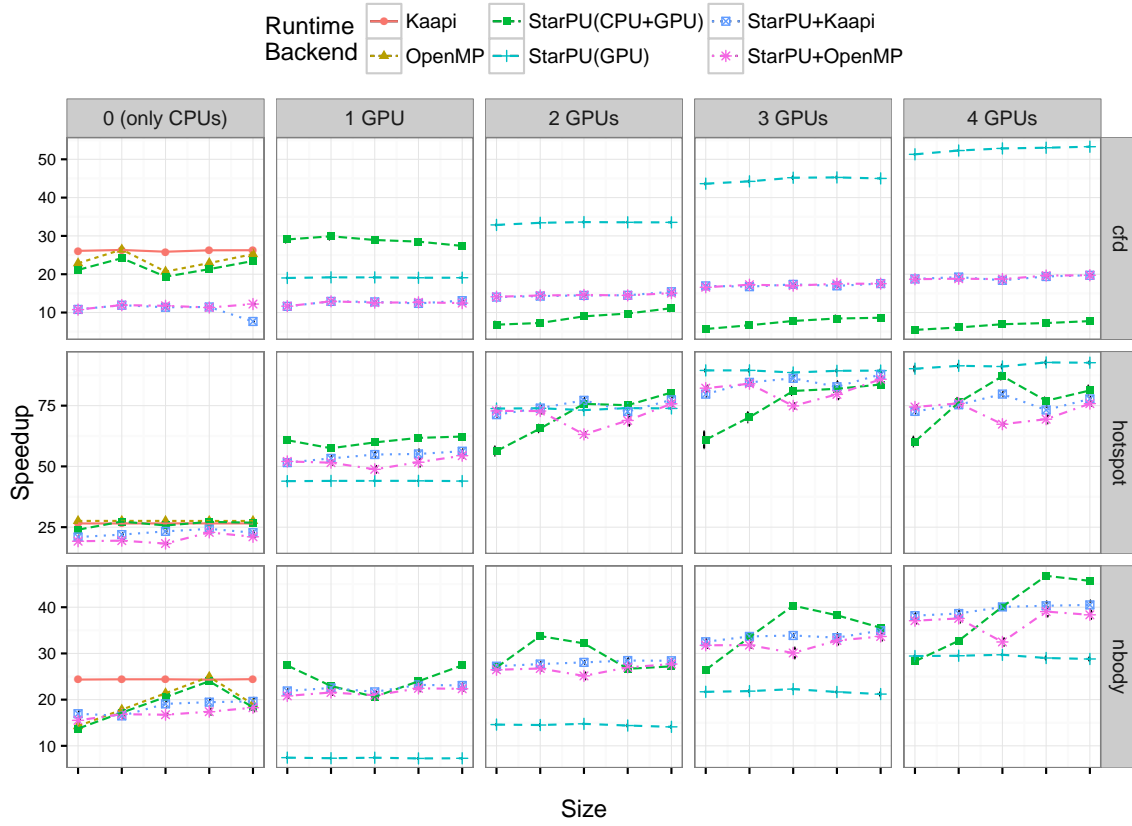


Figure 16: Speedup over sequential version by sizes with maximum threads and GPUs.

4. Speedup by size (best configuration):

Config:

* CFD:

StarPU: 0GPU+28CPUs, 1GPU+26CPUs, 2GPUs+12CPUs,
3GPUs+10CPUs, 4GPUs+10CPUs;

StarU+OpenMP: 0GPU+28CPUs, 1GPU+10CPUs, 2GPUs+10CPUs,
3GPUs+10CPUs, 4GPUs+10CPUs;

StarPU+Kaapi: 0GPU+28CPUs, 1GPU+10CPUs, 2GPUs+10CPUs,
3GPUs+10CPUs, 4GPUs+10CPUs;

* Hotspot

StarPU: 0GPU+28CPUs, 1GPU+26CPUs, 2GPUs+25CPUs,
3GPUs+24CPUs, 4GPUs+23CPUs;

StarU+OpenMP: 0GPU+28CPUs, 1GPU+12CPUs, 2GPUs+12CPUs,
3GPUs+10CPUs, 4GPUs+10CPUs;

StarPU+Kaapi: 0GPU+28CPUs, 1GPU+12CPUs, 2GPUs+12CPUs,
3GPUs+10CPUs, 4GPUs+10CPUs;

* NBody:

StarPU: 0GPU+28CPUs, 1GPU+27CPUs, 2GPUs+26CPUs,
3GPUs+25CPUs, 4GPUs+24CPUs;

StarU+OpenMP: 0GPU+28CPUs, 1GPU+27CPUs, 2GPUs+26CPUs,
3GPUs+25CPUs, 4GPUs+24CPUs;

StarPU+Kaapi: 0GPU+28CPUs, 1GPU+27CPUs, 2GPUs+26CPUs,
3GPUs+25CPUs, 4GPUs+24CPUs;

Sizes:

CFD: 98304, 114688, 131072, 147456, 163840;

Hostspot: 12288N, 14336N, 16384N, 18432N, 20480N;

NBody: 65536, 81920, 98304, 114688, 131072;
 Block sizes:
 CFD: 2048;
 Hostspot: 1024 x 1024;
 NBody: 2048

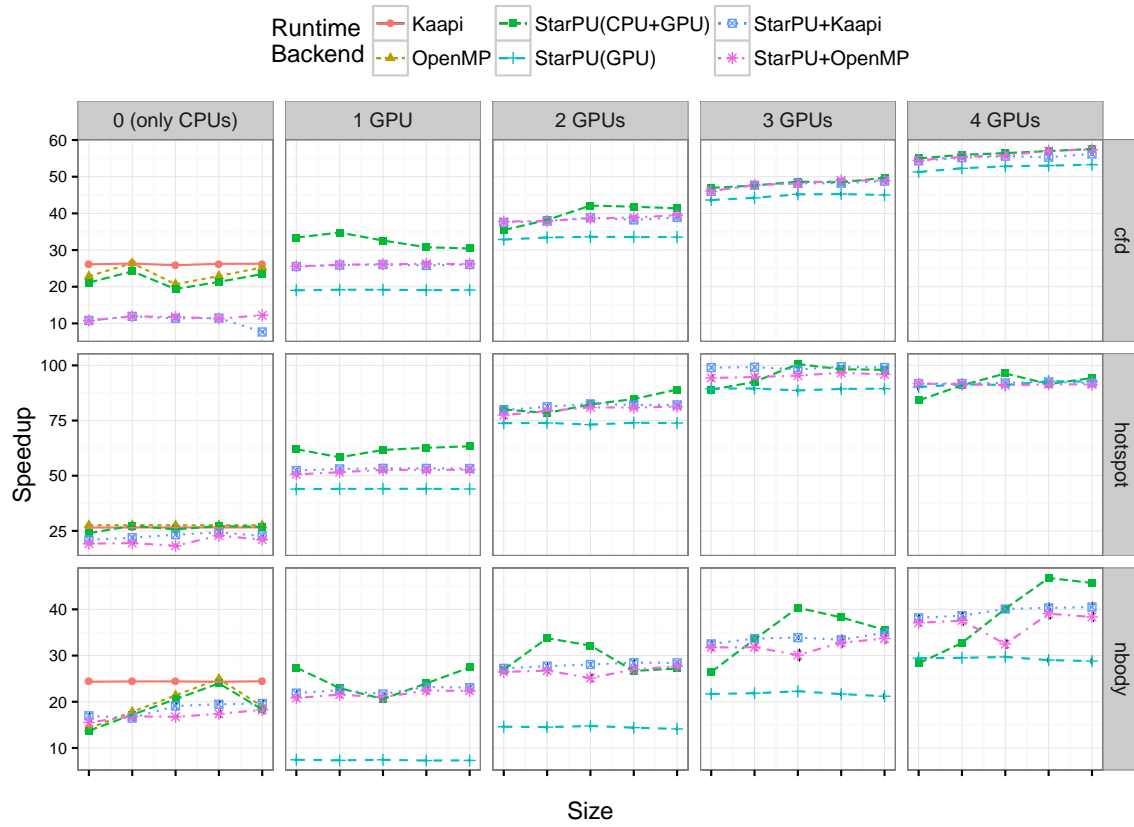


Figure 17: Speedup over sequential version by sizes with maximum threads and GPUs.

5. Speedup - calibragem melhor configuração CPUs + GPUs:

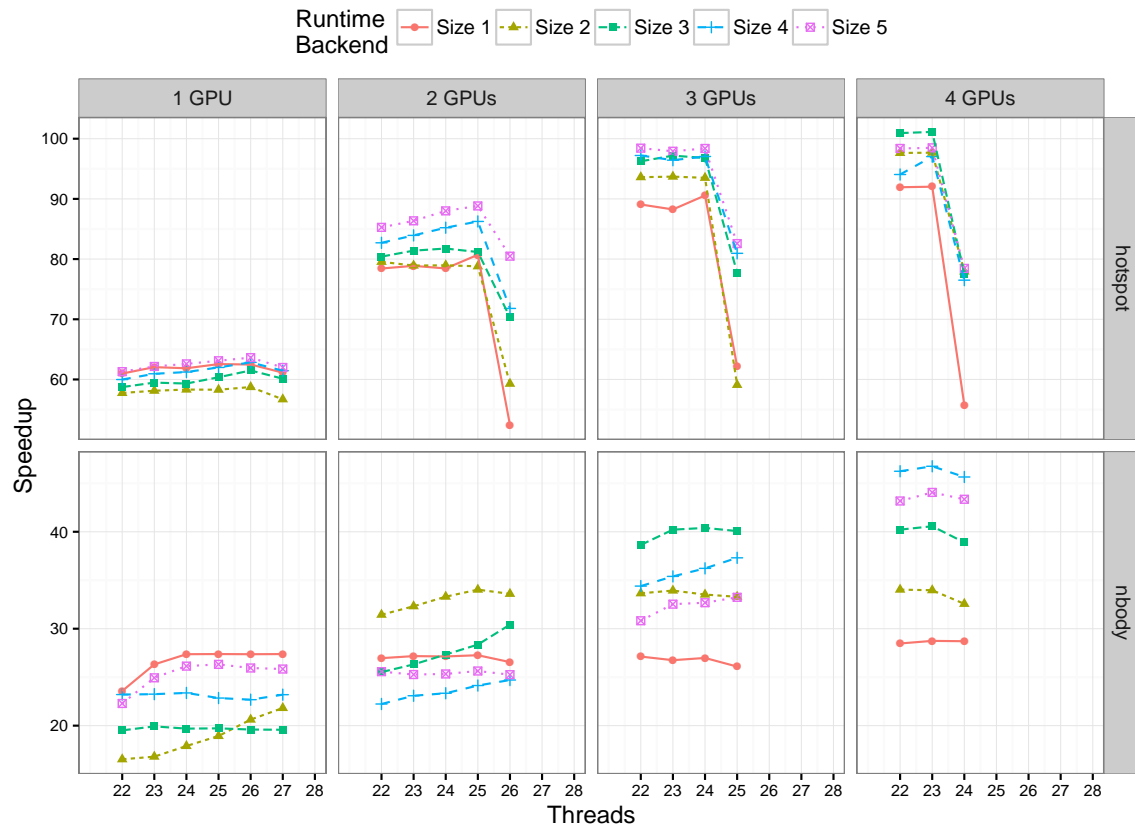


Figure 18: Speedup over sequential version by sizes calibrating CPUs + GPUs.

6. Time - calibragem melhor configuração CPUs + GPUs:

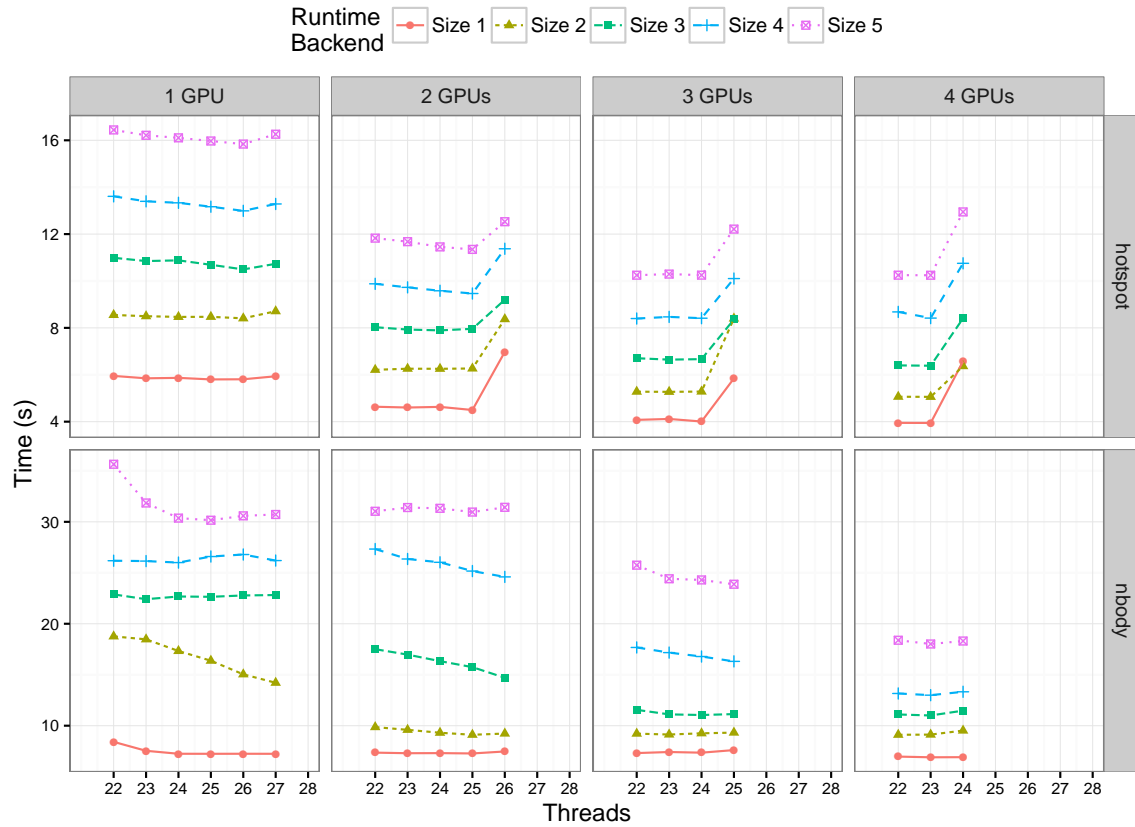


Figure 19: Time by sizes calibrating CPUs + GPUs.

7. Speedup - calibragem escalonador CFD, Hotspot e NBody com dmda, dm, dmdar, dmdas: realizados testes para identificar se modificando o escalonador da StarPU resultava em mudança no comportamento das execuções. No geral, o escalonamento DMDA é o que obteve os melhores desempenhos.

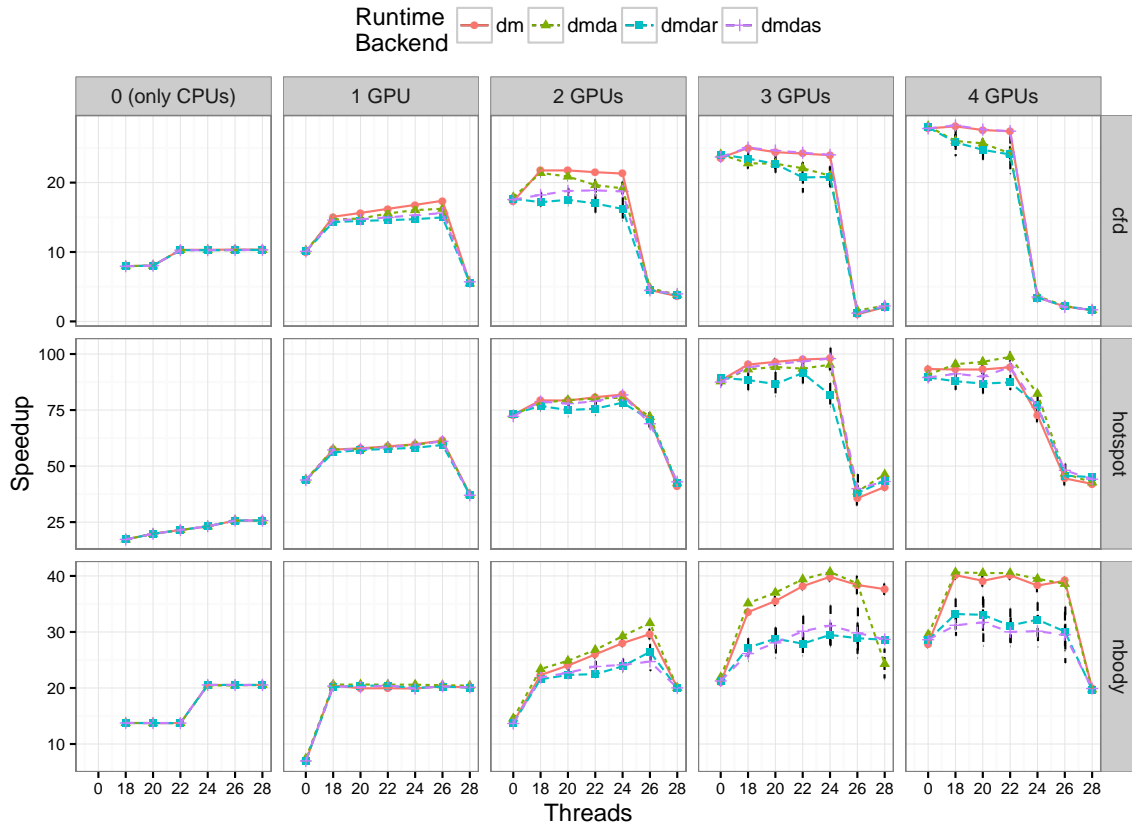


Figure 20: Speedup over sequential version by sizes calibrating schedules.

8. Speedup comparing two block sizes with StarPU, StarPU+OpenMP e StarPU+Kaapi: teste realizado para avaliar o desempenho com um bloco maior (dobro) que o utilizado nos testes anteriores. Espera-se que o desempenho dos back-ends StarPU+OMP e StarPU+Kaapi alcancem resultados melhores com o bloco maior, pois haverá mais paralelismo disponível para as tarefas.

- (a) CFD: block size 2048 and 4096 (size: 131072);
- (b) Hostpot: block size 1024 and 2048 (size: 16384);
- (c) NBody: block size 2048 and 4096 (size: 98304);

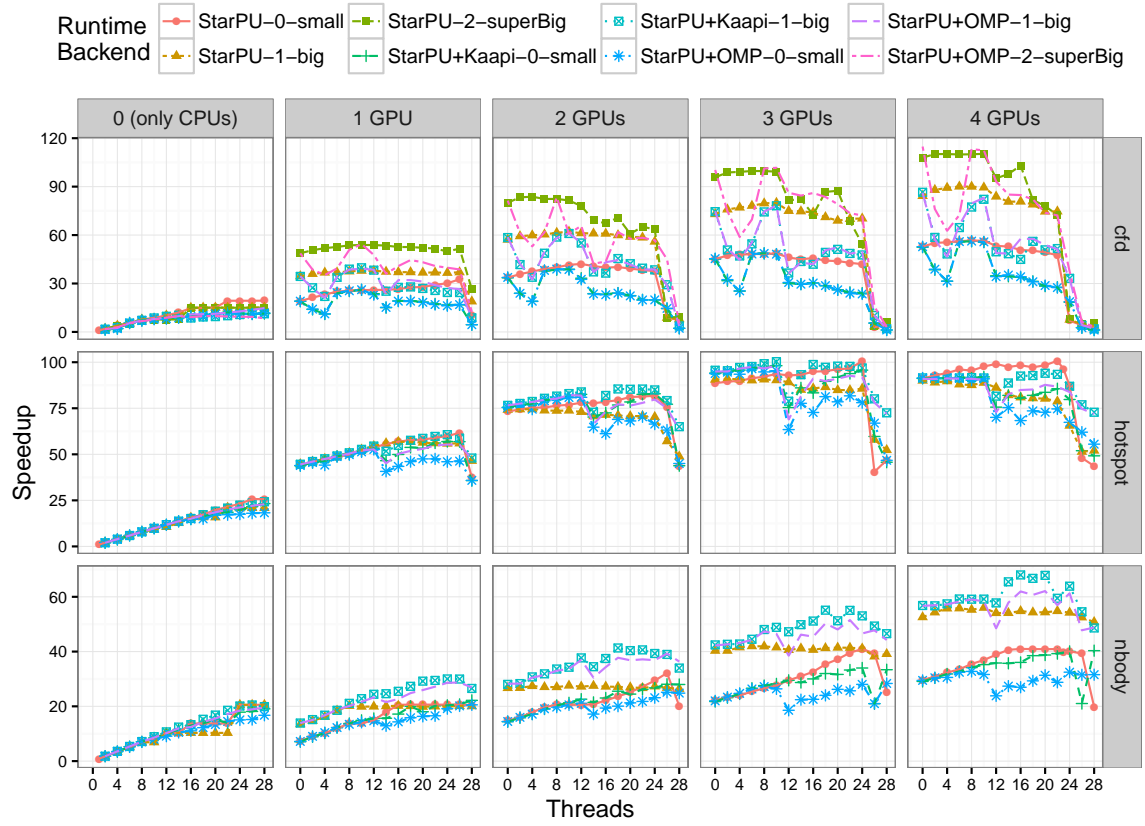


Figure 21: Speedup over sequential version using StarPU/StarPU+OMP/StarPU+Kaapi with different block sizes.

9. Time sequencial: tempo das execuções seriais variando o tamanho da entrada:

Sizes:

CFD: 98304, 114688, 131072, 147456, 163840;

Hostspot: 12288N, 14336N, 16384N, 18432N, 20480N;

NBody: 65536, 81920, 98304, 114688, 131072;

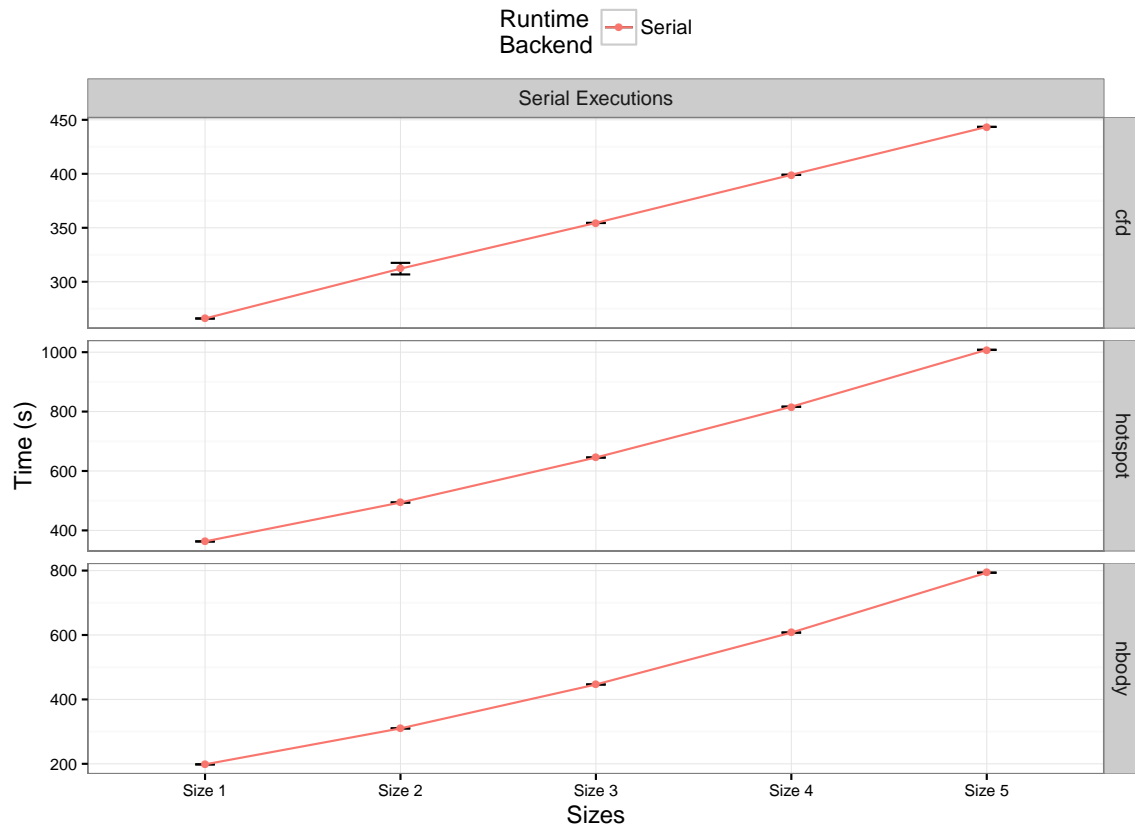


Figure 22: Time by sizes.

1.4.3 Rastros

1. Hotspot com GPUs mais máximo de threads:

- Traços: traços para identificar a razão da execução com uma thread a menos que o máximo possível resultar no melhor desempenho.

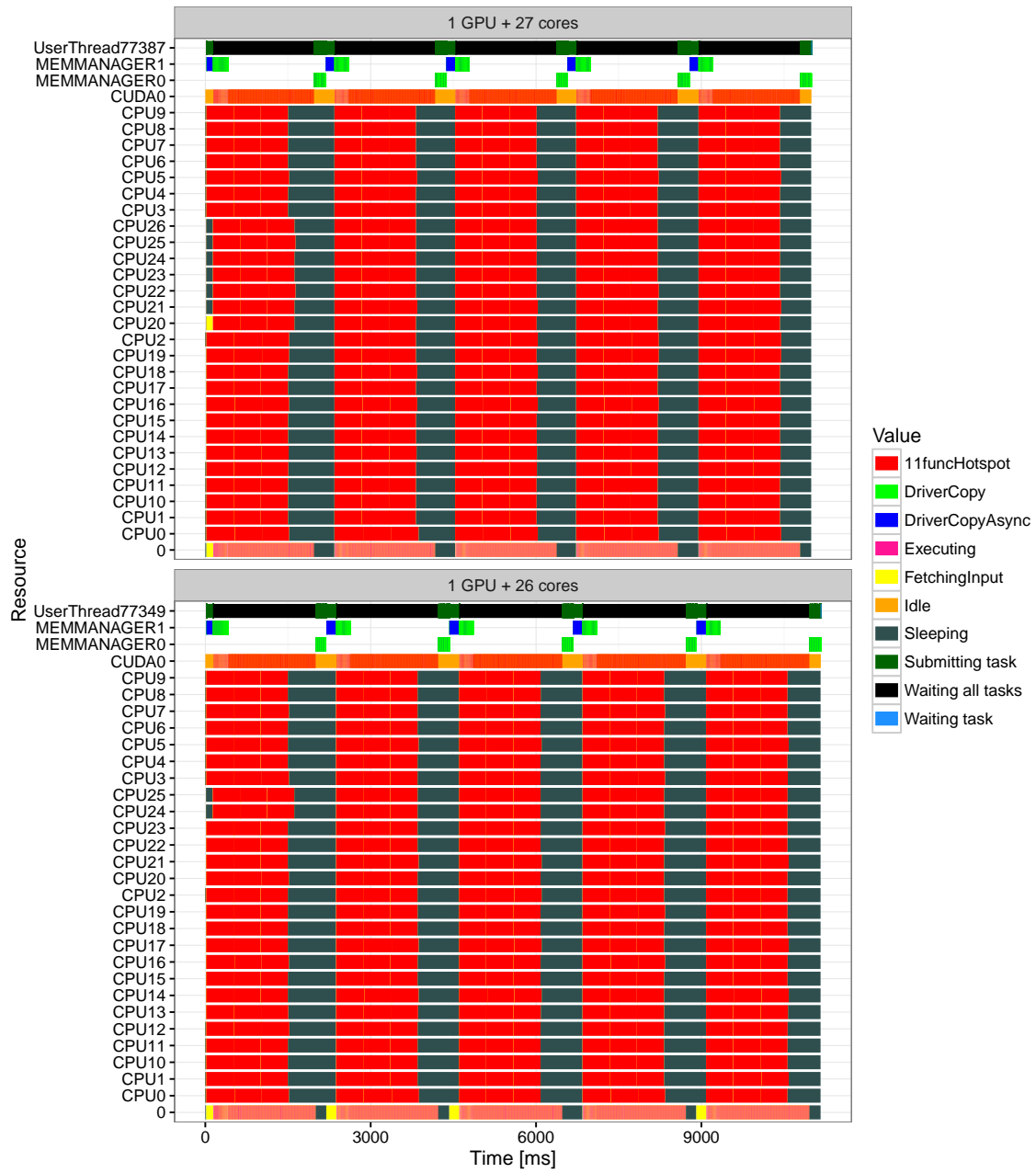


Figure 23: Trace from Hotspot benchmark with 1 GPU + 26/27 threads.

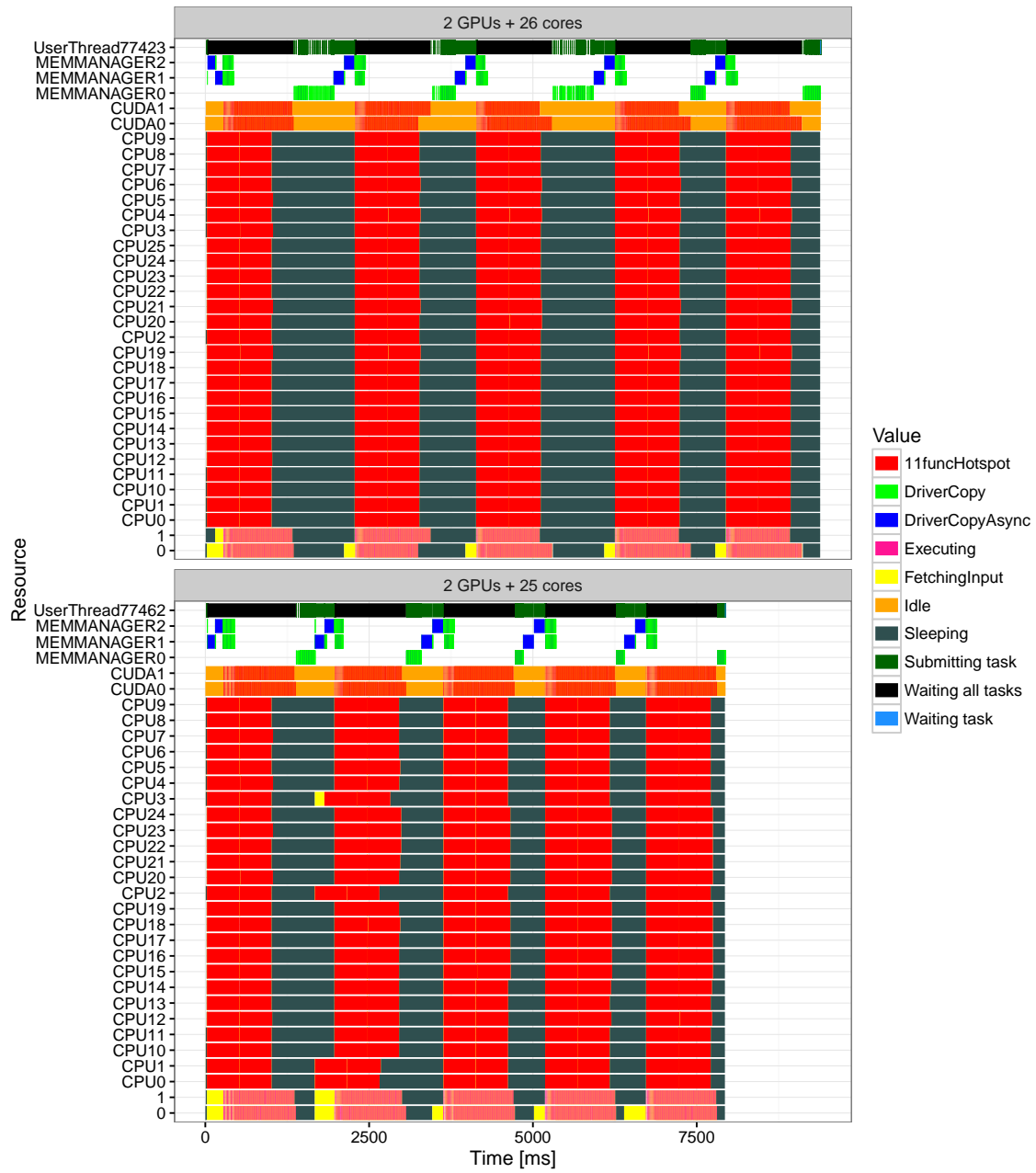


Figure 24: Trace from Hotspot benchmark with 2 GPUs + 25/26 threads.

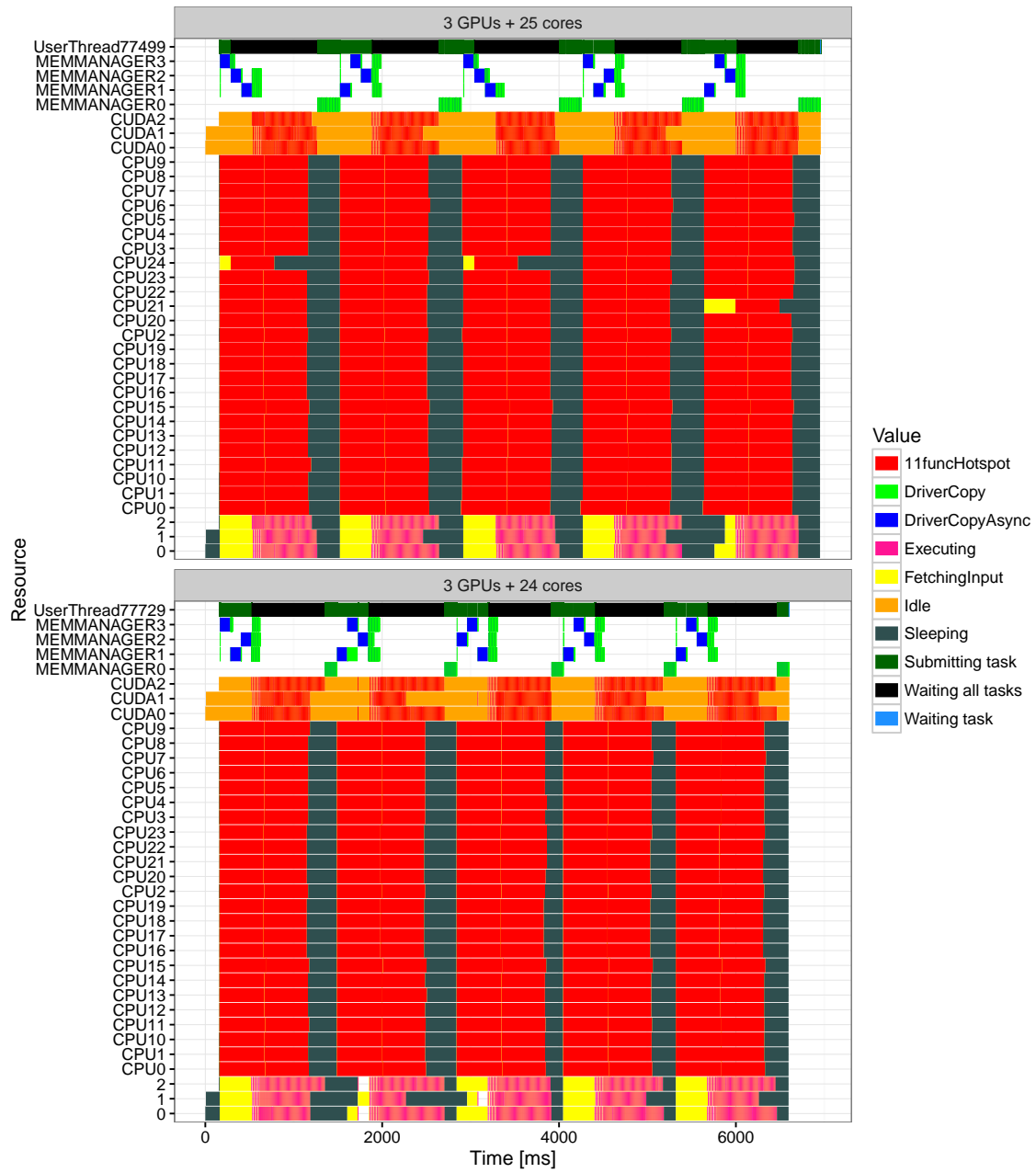


Figure 25: Trace from Hotspot benchmark with 3 GPUs + 24/25 threads.

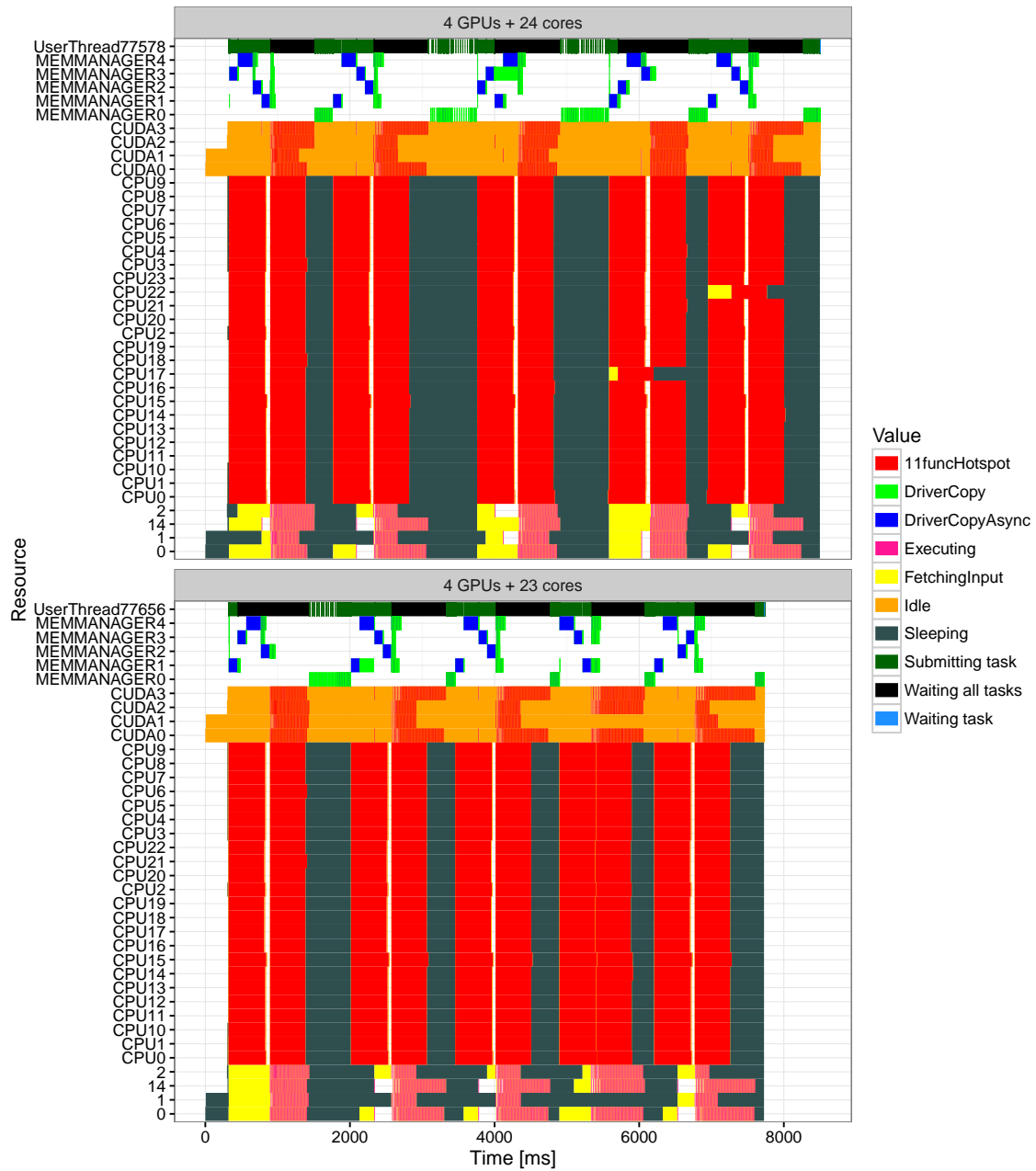


Figure 26: Trace from Hotspot benchmark with 4 GPUs + 23/24 threads.

2. NBody com 1 GPUs mais 16 a 26 threads:

- (a) Traços: indicam que o motivo da estabilização dos resultados a partir de 16 threads é que o escalonador opta por não utilizá-las, visto que elas ficam ociosas. Desta forma, as tarefas passam a ser processadas apenas pelas GPUs.

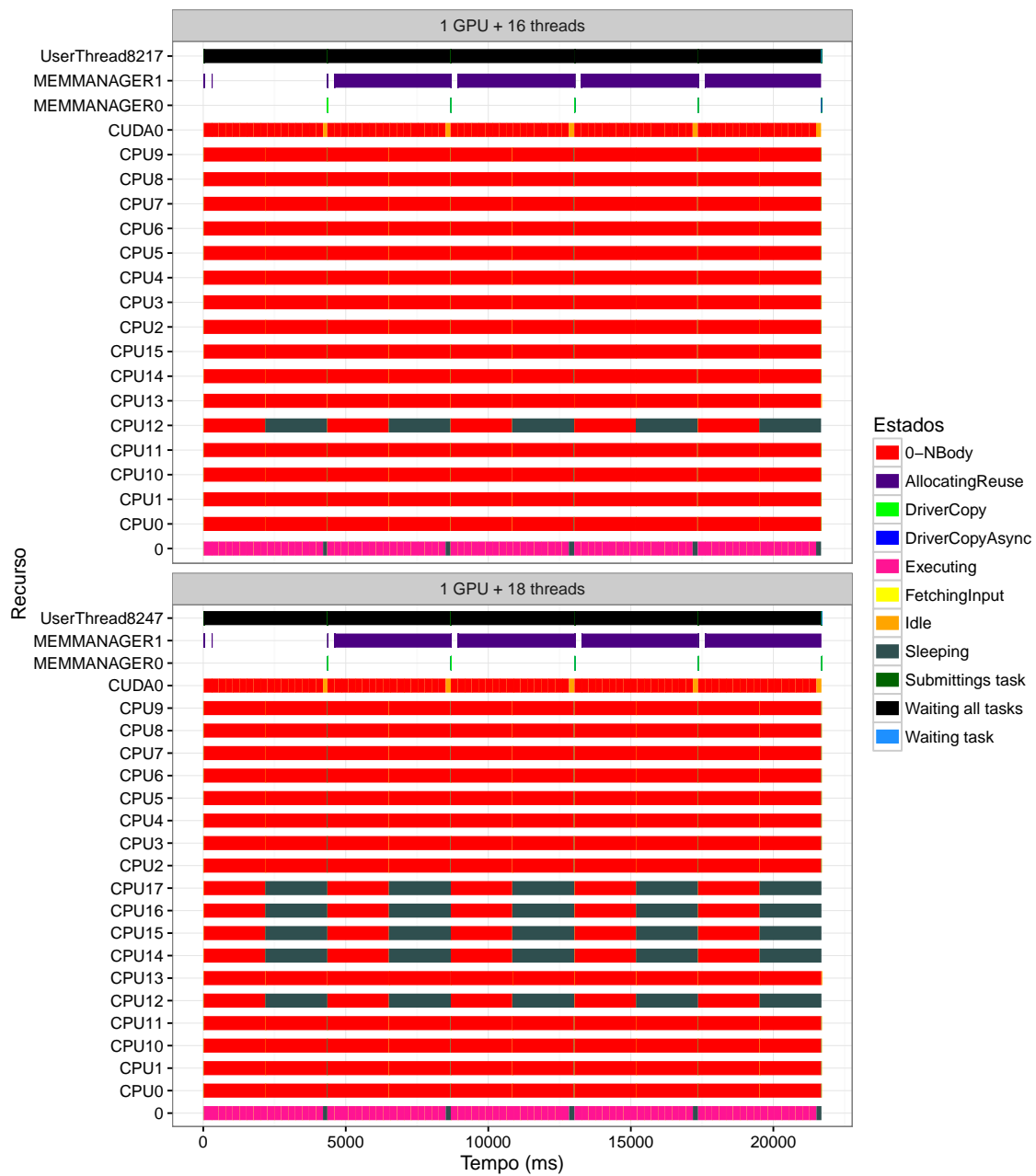


Figure 27: Trace from NBody benchmark with 1 GPU + 16/18 threads.

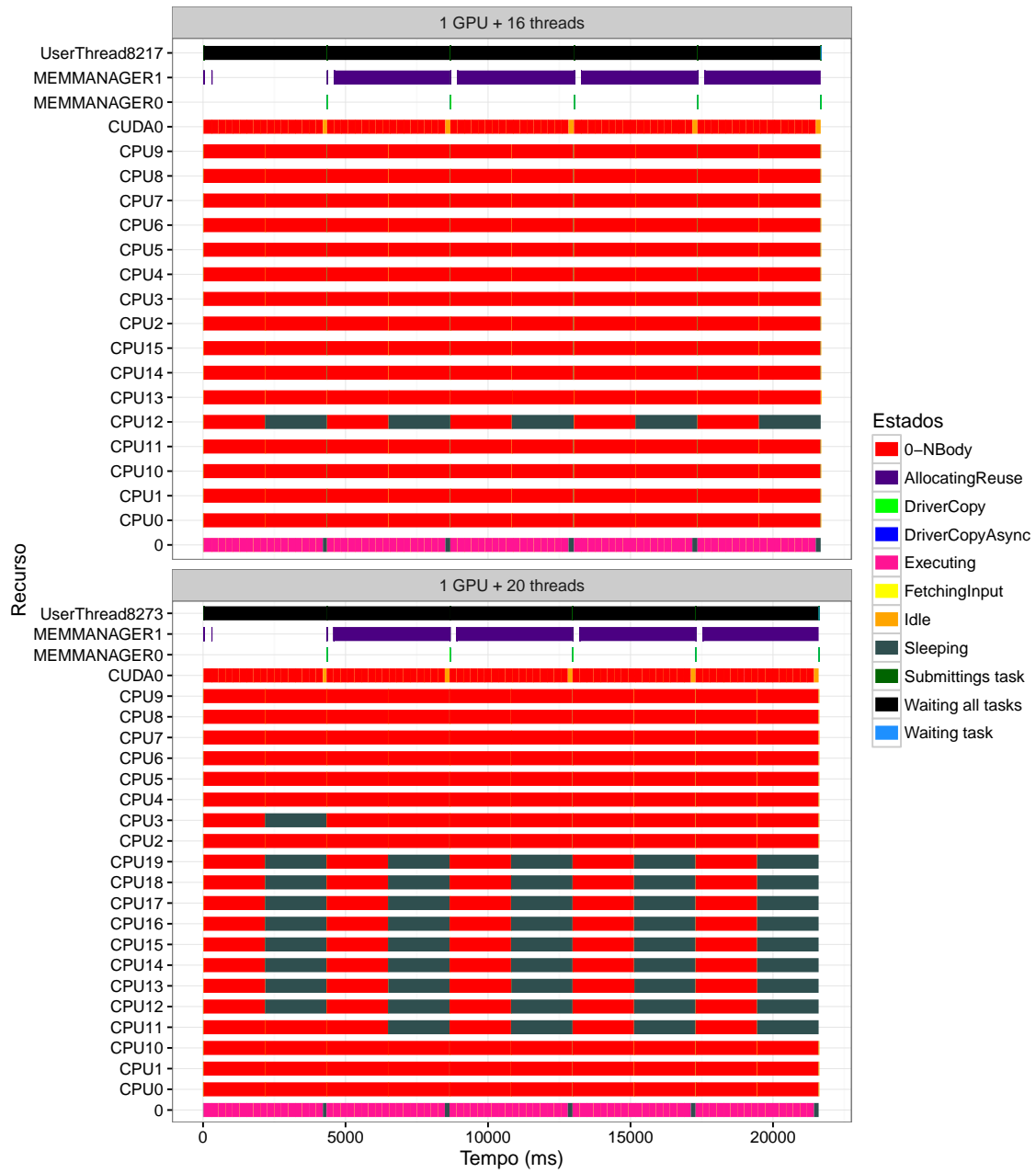


Figure 28: Trace from NBody benchmark with 1 GPU + 16/20 threads.

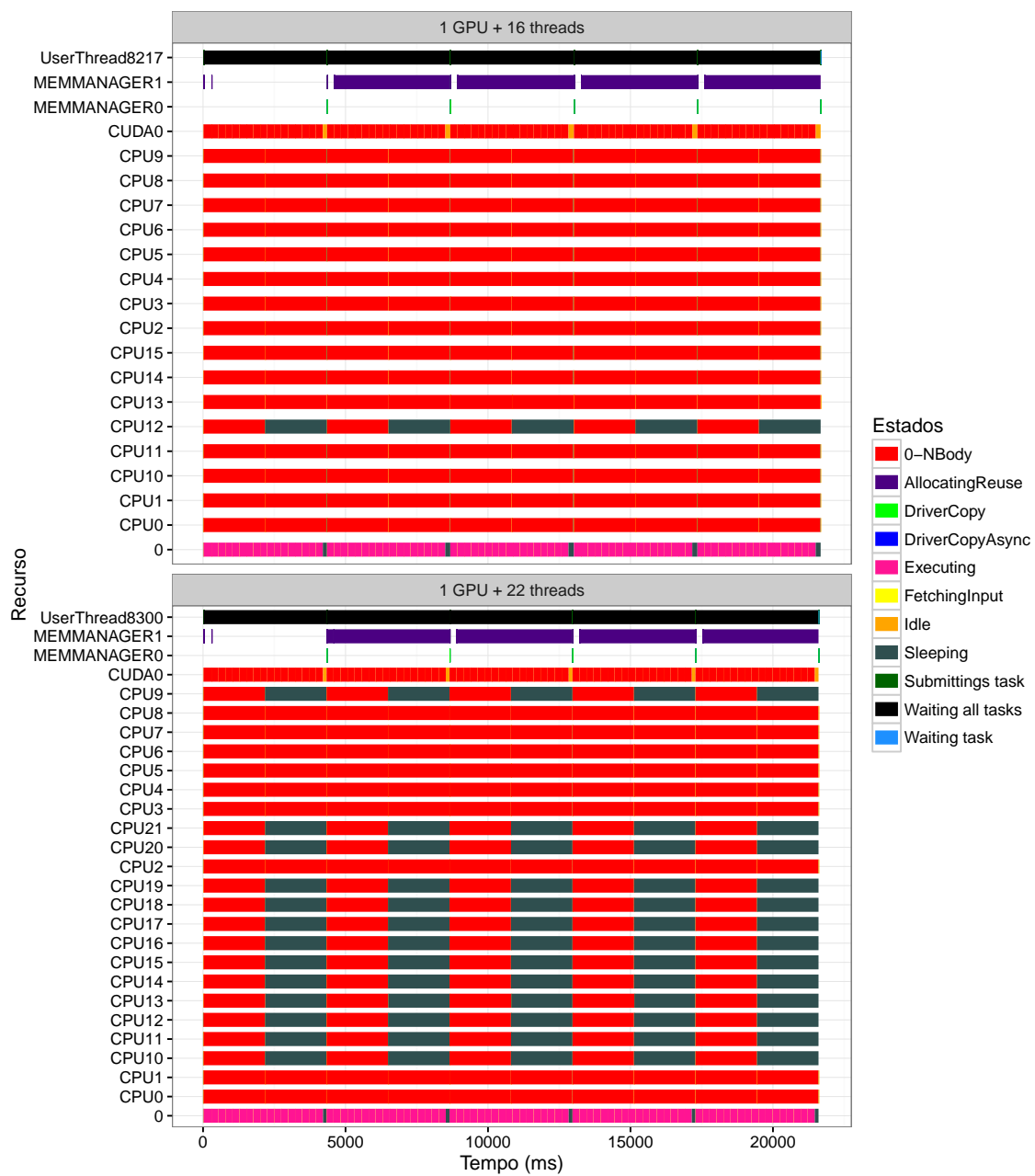


Figure 29: Trace from NBody benchmark with 1 GPU + 16/22 threads.

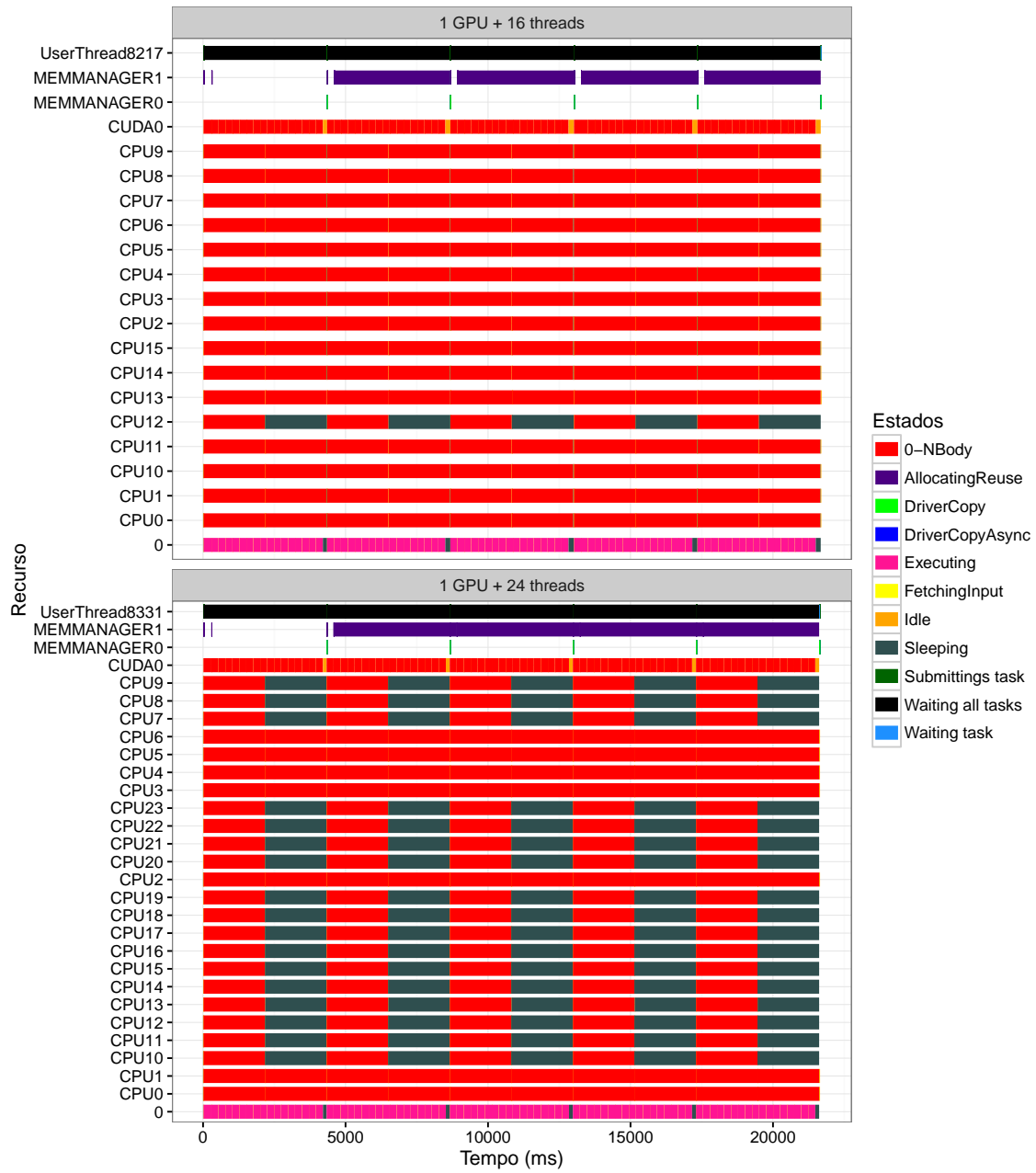


Figure 30: Trace from NBody benchmark with 1 GPU + 16/24 threads.

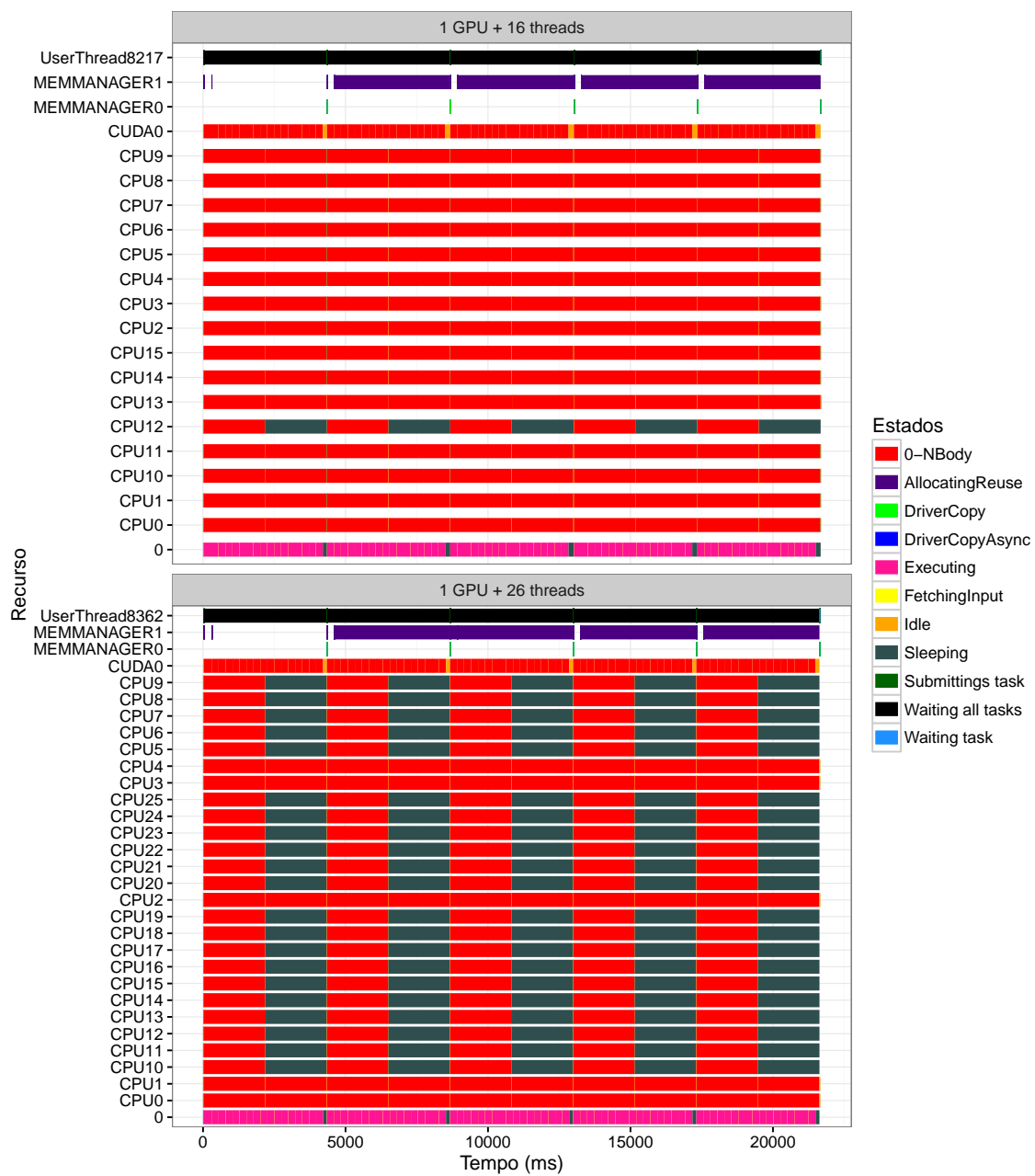


Figure 31: Trace from NBody benchmark with 1 GPU + 16/26 threads.

3. StarPU+OpenMP - resultados estranhos:

(a) Hotspot: queda de rendimento de 10 para 12 threads.

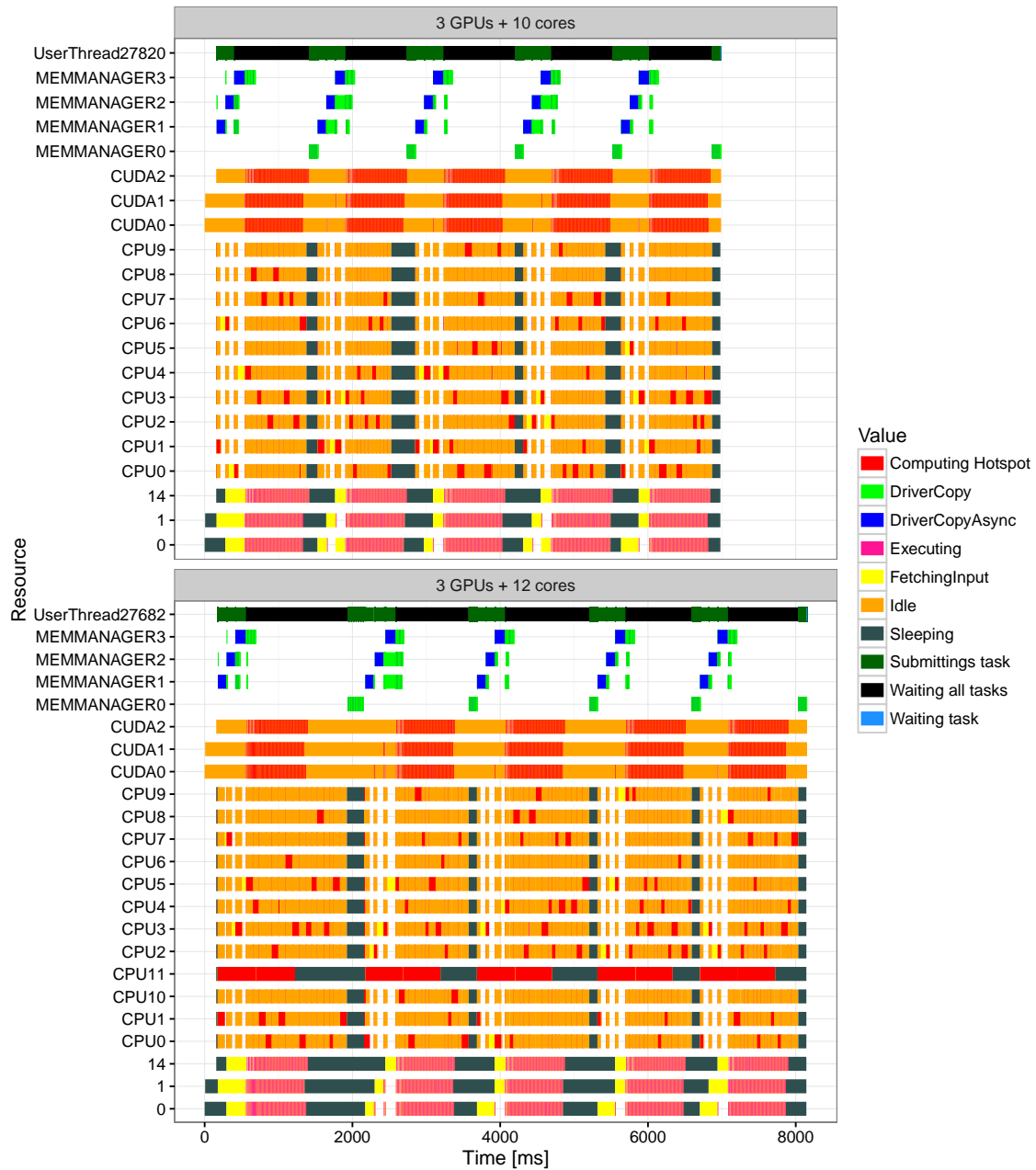


Figure 32: Trace from Hotspot benchmark (StarPU+OpenMP) with 3 GPU + 10/12 threads.

(b) CFD: desempenho bem baixo com 4 threads em relação à 6 threads.

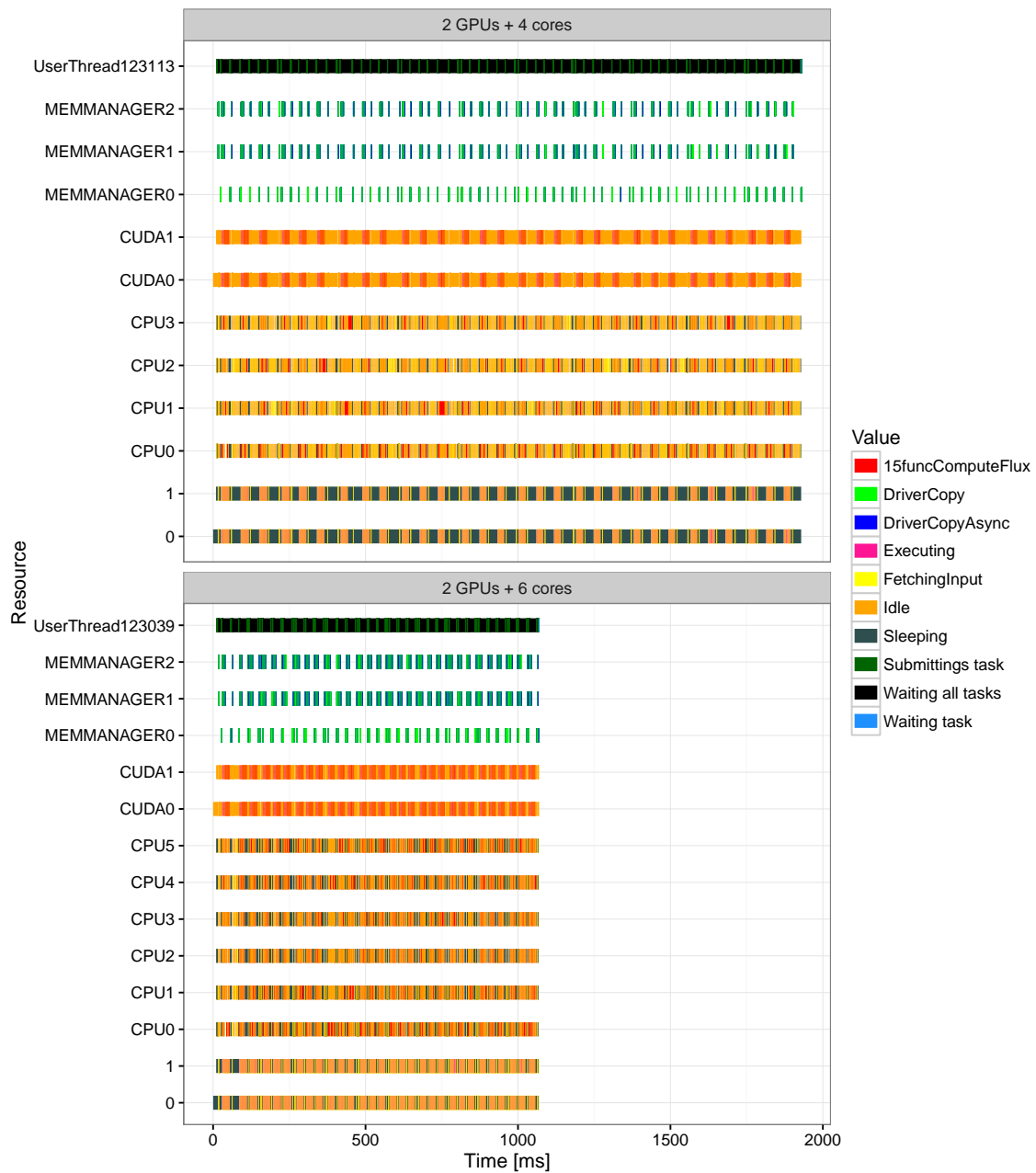


Figure 33: Trace from CFD benchmark (StarPU+OpenMP) with 2 GPU + 4/6 threads.

1.4.4 Perfmodel StarPU+OpenMP

1. Calibragem das tarefas CFD - Size: 131072 - comprovação do efeito NUMA, com piora no tempo a partir de 14 threads para os dois tamanhos de bloco:

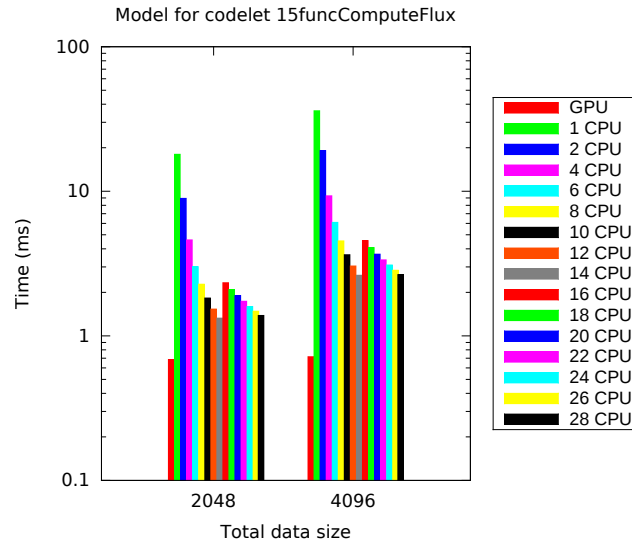


Figure 34: StarPU+OpenMP perfmodel from CFD benchmark - Size: 131072.

2. Calibragem das tarefas Hotspot - Size: 16384 x 16384 - não há ocorrência do efeito NUMA:

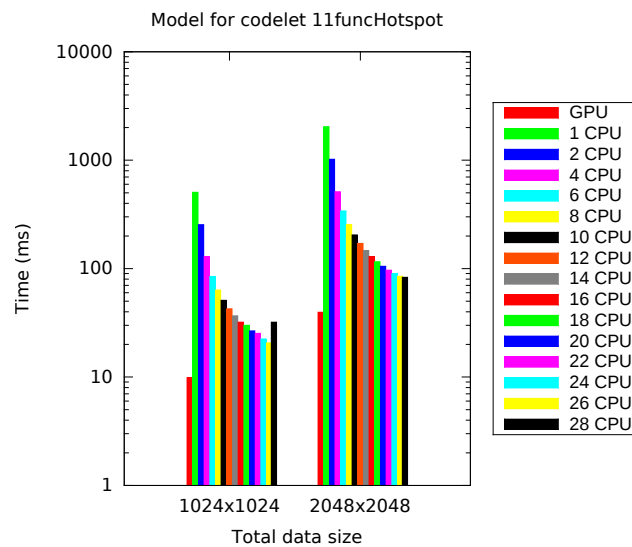


Figure 35: StarPU+OpenMP perfmodel from Hotspot benchmark - Size: 16384 x 16384.

3. Calibragem das tarefas NBody - Size: 98304 - não há ocorrência do efeito NUMA:

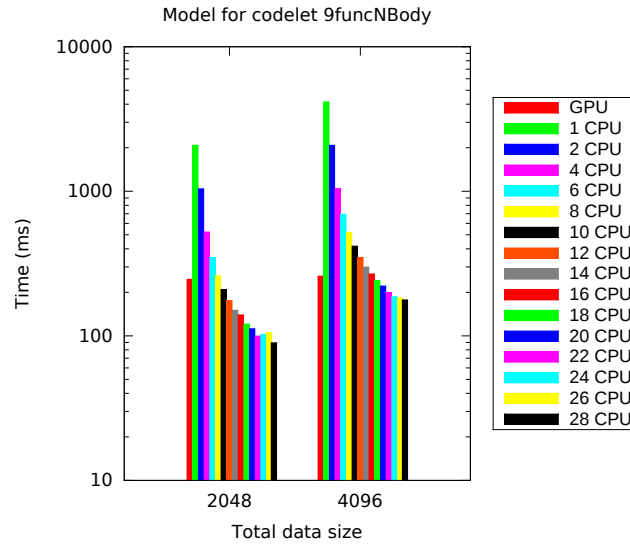


Figure 36: StarPU+OpenMP perfmodel from NBody benchmark - Size: 98304.

1.4.5 Perfmodel StarPU+Kaapi

1. Calibragem das tarefas CFD - Size: 131072 - comprovação do efeito NUMA, com piora no tempo a partir de 14 threads para os dois tamanhos de bloco:

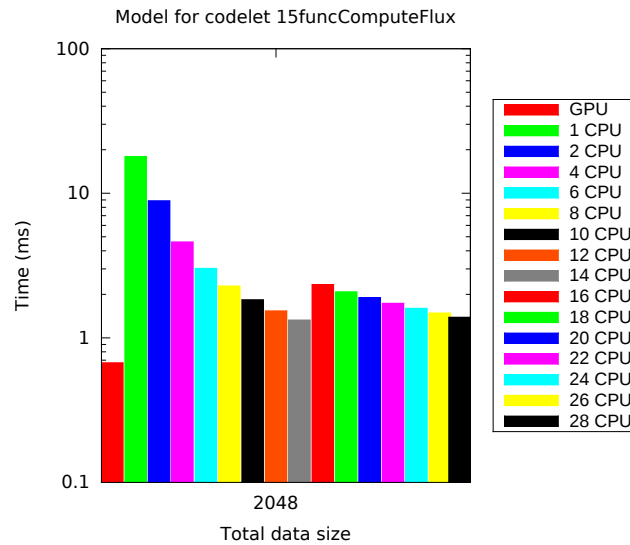


Figure 37: StarPU+Kaapi perfmodel from CFD benchmark - Size: 131072.

2. Calibragem das tarefas Hotspot - Size: 16384 x 16384 - não há ocorrência do efeito NUMA:

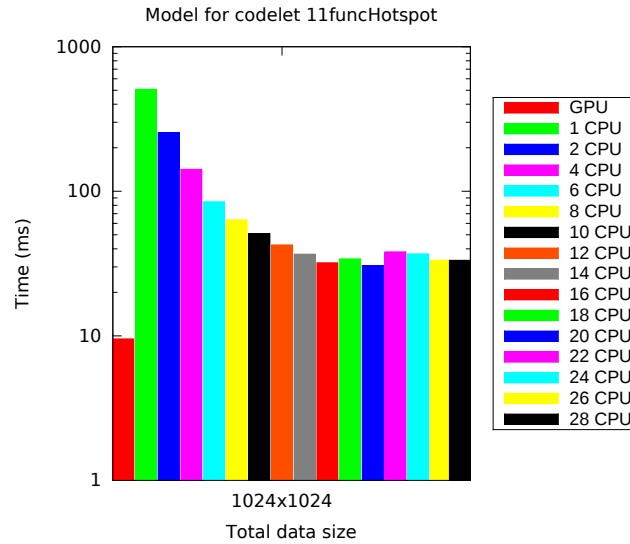


Figure 38: StarPU+Kaapi perfmodel from Hotspot benchmark - Size: 16384 x 16384.

3. Calibragem das tarefas NBody - Size: 98304 - não há ocorrência do efeito NUMA:

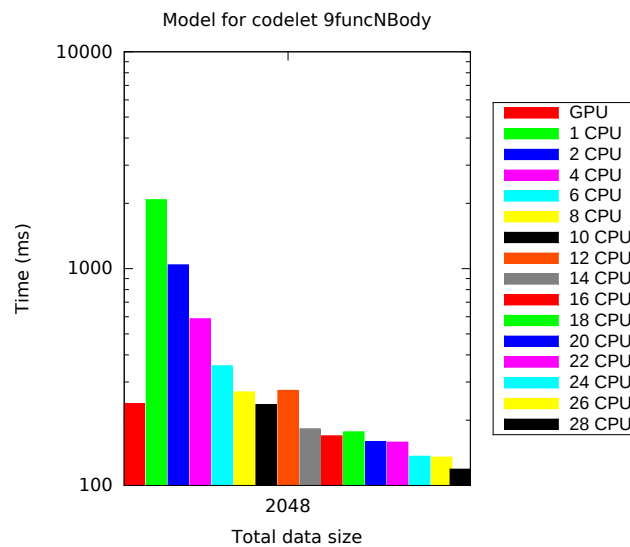


Figure 39: StarPU+Kaapi perfmodel from NBody benchmark - Size: 98304.

1.4.6 Overhead da API

Comparado o tempo sequencial com o tempo obtido pela API executando com 1 thread. O gráfico de overhead demonstra que o uso da HPSM traz um custo para a execução que varia de acordo com a aplicação. Com a Hotspot o sobrecusto foi praticamente 0, enquanto que com a Hotspot o custo é maior, com os tempos de execução aumento até 14%.

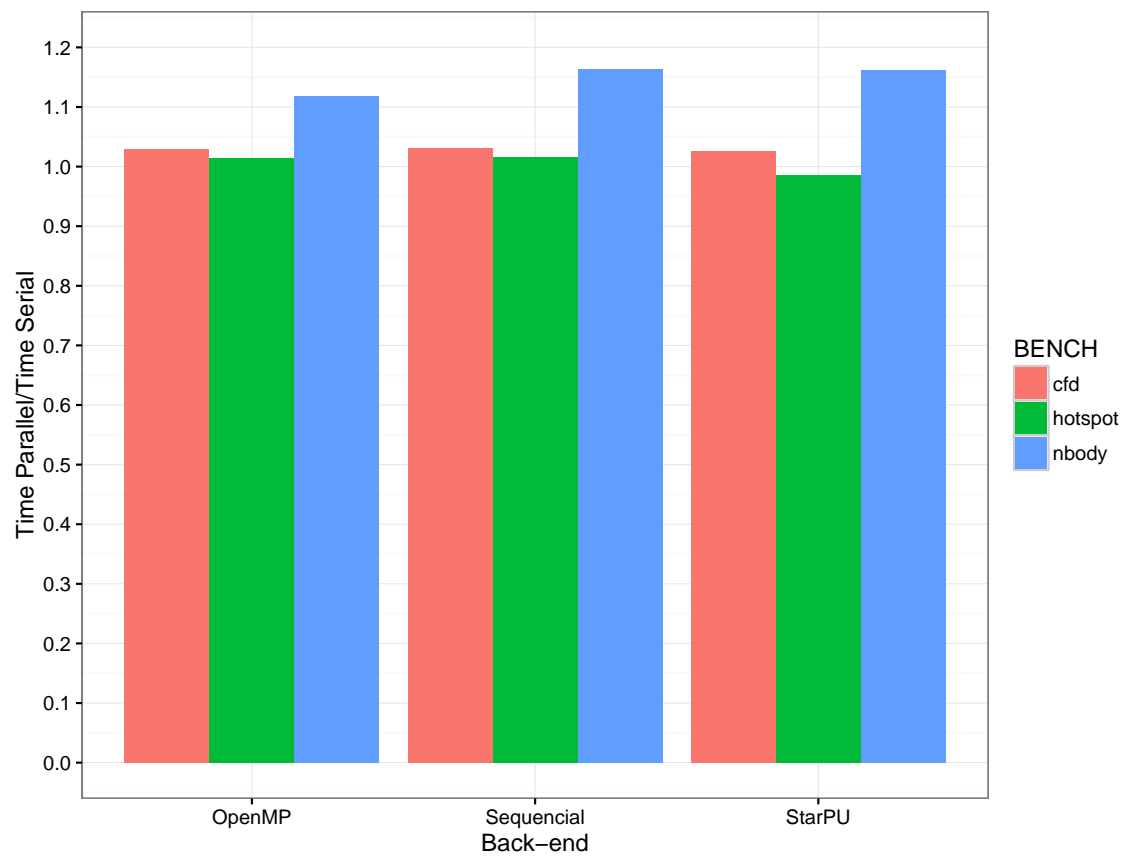


Figure 40: API overhead.