CAP-399/2019 - Primeira Lista de Exercícios

CONTEÚDO

INFORMAÇÕES

Os arquivos estão em

```
fish://xxxxxxx.xxxxxxx@login.sdumont.lncc.br/prj/padinpe/xxxxxxx.xxxxxxx/
fish://xxxxxxx.xxxxxxx@login.sdumont.lncc.br/scratch/padinpe/xxxxxxx.xxxxxxx/
```

Os fontes (pname.c e stream.c) foram obtidos em

```
fish://xxxxxxx.xxxxxxx@login.sdumont.lncc.br/prj/padinpe/xxxx.xxxx/
```

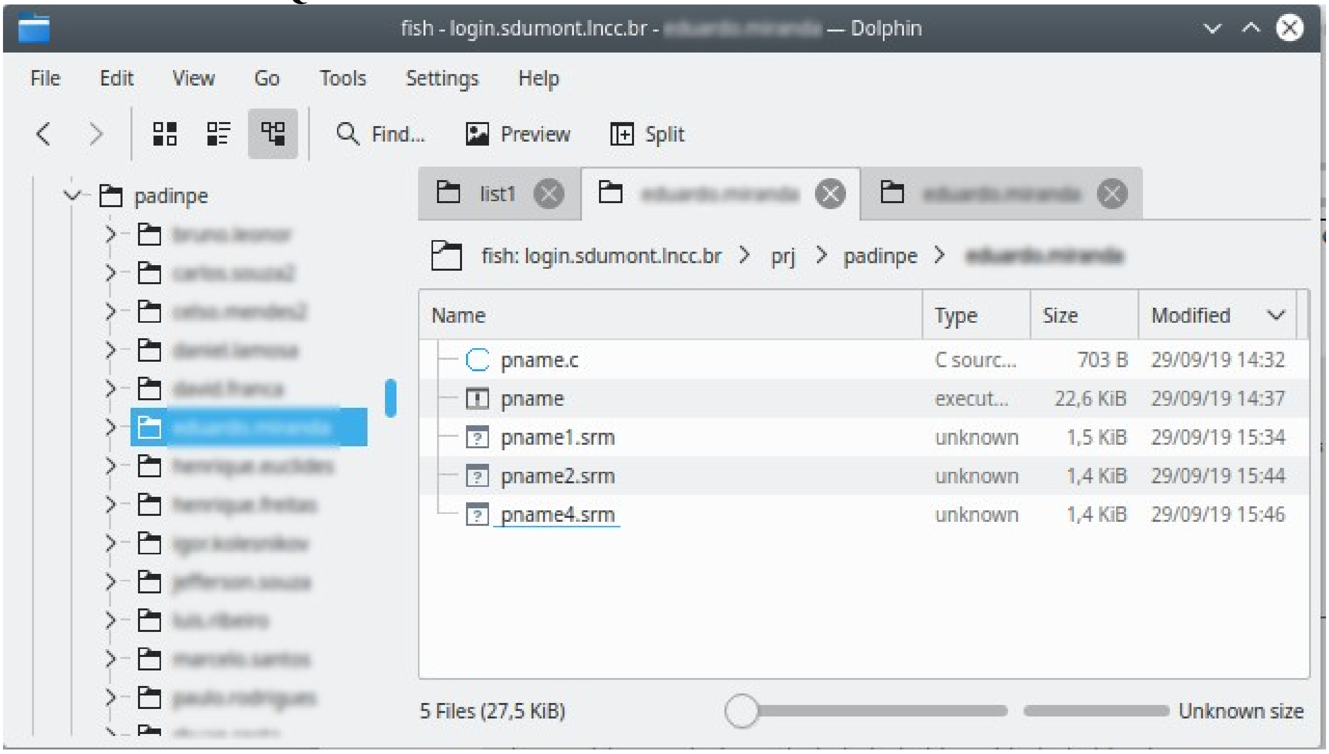
A filha escolhida foi a `cpu_dev`

Fila	Wall-clock máximo (em horas)	Número mínimo de nós (núcleos+ dispositivos)	Número máximo de nós (núcleos+ dispositivos)	Número máximo de tarefas em execução por usuário	Número máximo de tarefas em fila por usuário	Custo em Unidade de Alocação (UA)
cpu (Nós B710)	96	21 (504)	50 (1200)	4	24	1
cpu_dev 1	0:20	1 (24)	4 (96)	1	1	1

fonte: https://sdumont.lncc.br/support_manual.php?pg=support

EXERCÍCIO 1

COPIANDO OS ARQUIVOS

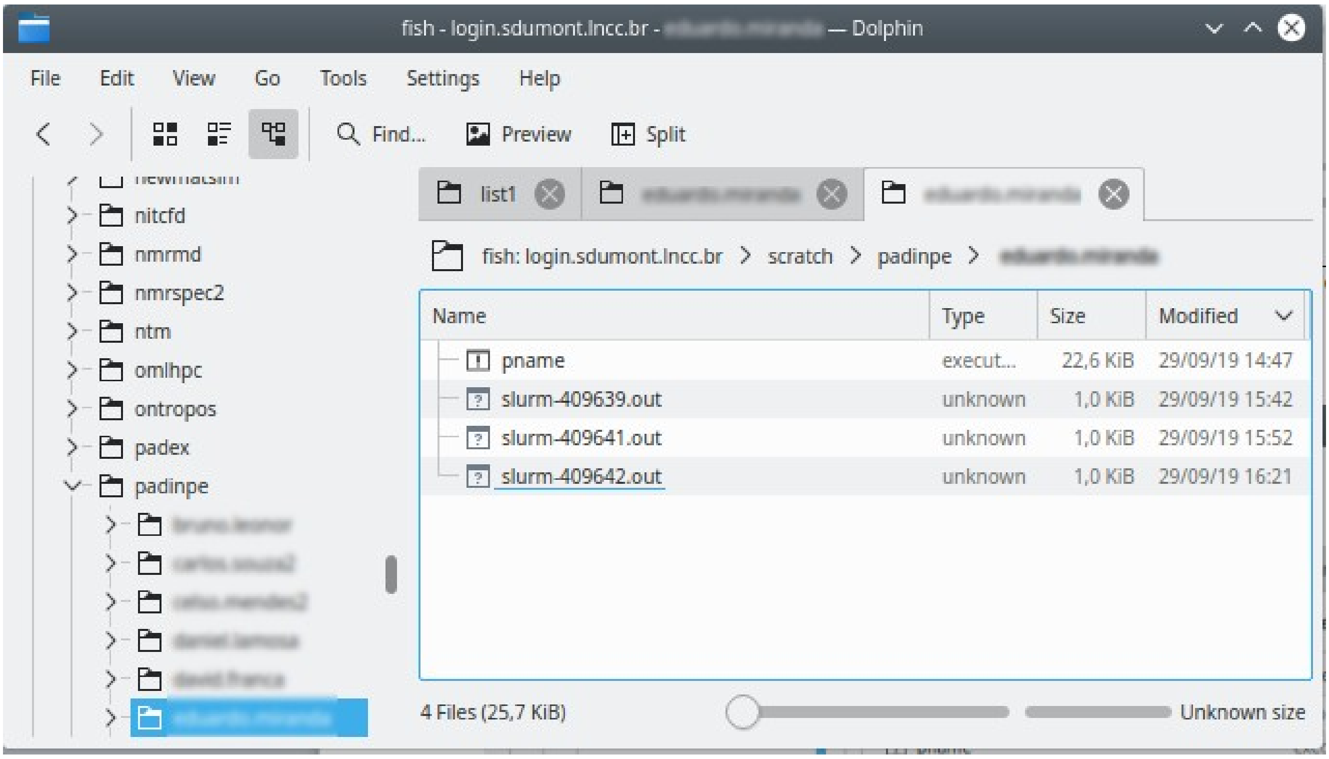


fish: login.sdumont.lncc.br > prj > padinpe > [redacted]

Name	Type	Size	Modified
pname.c	C sourc...	703 B	29/09/19 14:32
pname	execut...	22,6 KiB	29/09/19 14:37
pname1.srm	unknown	1,5 KiB	29/09/19 15:34
pname2.srm	unknown	1,4 KiB	29/09/19 15:44
pname4.srm	unknown	1,4 KiB	29/09/19 15:46

5 Files (27,5 KiB)

Unknown size



fish: login.sdumont.lncc.br > scratch > padinpe > [redacted]

Name	Type	Size	Modified
pname	execut...	22,6 KiB	29/09/19 14:47
slurm-409639.out	unknown	1,0 KiB	29/09/19 15:42
slurm-409641.out	unknown	1,0 KiB	29/09/19 15:52
slurm-409642.out	unknown	1,0 KiB	29/09/19 16:21

4 Files (25,7 KiB)

Unknown size

COMPILANDO

```
$ sudo vpnc /etc/vpnc/sdumont.conf
Enter password for xxxxxxx.xxxxxxx@146.134.0.14:
Connect Banner:
| Todos os acessos a partir de agora estao sendo monitorados
|

VPNC started in background (pid: 23038)...
$ ssh xxxxxxx.xxxxxxx@login.sdumont.lncc.br
xxxxxxx.xxxxxxx@login.sdumont.lncc.br's password:
Last login: Sun Sep 29 14:35:25 2019 from 146.134.223.79
________________________________________________________________
   _____ _____                             _
  / ____|  __ \                           | |
 | (___ | |  | |_   _ _ __ ___   ___  _ __ | |_
  \___ \| |  | | | | | '_ ` _ \ / _ \| '_ \| __|
  ____) | |__| | |_| | | | | | | (_) | | | | |_
 |_____/|_____/ \__,_|_| |_| |_|\___/|_| |_|\__|



Manual: http://sdumont.lncc.br/support_manual.php
_________________________________________________________________
The available softwares can be listed with the command: module avail
If there's something missing, please get in contact with helpdesk-sdumont@lncc.br
_________________________________________________________________

[xxxxxxx.xxxxxxx@sdumont13 ~]$ ls
pname1.srm  pname2.srm  pname4.srm  pname.c
[xxxxxxx.xxxxxxx@sdumont13 ~]$ module load intel_psxe/2019
[xxxxxxx.xxxxxxx@sdumont13 ~]$ mpiicc -o pname pname.c
[xxxxxxx.xxxxxxx@sdumont13 ~]$ ls
pname  pname1.srm  pname2.srm  pname4.srm  pname.c
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sbatch ./pname1.srm
Submitted batch job 409639
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sbatch ./pname2.srm
Submitted batch job 409641
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sbatch ./pname4.srm
Submitted batch job 409642
[xxxxxxx.xxxxxxx@sdumont13 ~]$ squeue --job 409642
             JOBID PARTITION     NAME     USER ST       TIME  NODES
NODELIST(REASON)
            409642   cpu_dev    pname xxxxxxx. PD       0:00      4 (Resources)
```

(a) 16 ranks, 1 nó

SCRIPT pname1.srm

```
#!/bin/bash
# Script baseado em:
#   "6.3. Jobs paralelos (threads/OpenMP)
#   Forma Geral de um Script"
#   <https://sdumont.lncc.br/support_manual.php?pg=support>
# Uso:
#   $ sbatch pname.srm          (roda em prj)(anotar o my_job_no)
#   $ squeue --job my_job_no    (verifica jobs)
#   A saída "slurm-my_job_no.out" aparece em "scracth"

#SBATCH --nodes=1               #Numero de Nós  (a)
#SBATCH --ntasks-per-node=16    #Numero de tarefas por Nó  (b)
#SBATCH --ntasks=16             #Numero total de tarefas MPI  (a x b)
#SBATCH -p cpu_dev              #Fila (partition) a ser utilizada
#SBATCH -J pname                #Nome do job
#SBATCH --time=00:02:00         #Tempo limite
#SBATCH --exclusive             #Utilizacao exclusiva dos nós durante o job

#Exibe os nós alocados para o Job
echo $SLURM_JOB_NODELIST
nodeset -e $SLURM_JOB_NODELIST
```

```
cd $SLURM_SUBMIT_DIR

#Configura os compiladores com intel MPI PSXE
source /scratch/app/modulos/intel-psxe-2019.sh
#Configura I_MPI_PMI_LIBRARY para apontar para a biblioteca \
#"Process Management Interface" do Slurm
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so

#Configura o executavel
EXEC=/scratch/padinpe/xxxxxxx.xxxxxxx/pname

#exibe informações sobre o executável (opcional)
#   /usr/bin/ldd $EXEC

#Dispara a execução
#   $SLURM_NTASKS : Same as --ntasks
#   $SLURM_CPUS_PER_TASK : Same as --cpus-per-task
srun    -n $SLURM_NTASKS $EXEC
```

SAÍDA slurm-409639.out

```
sdumont1405
sdumont1405
Intel(R) Parallel Studio XE 2019 Update 3 for Linux*
Copyright (C) 2009-2019 Intel Corporation. All rights reserved.
Process 12 of 16 on sdumont1405 (hostname sdumont1405)
Process 13 of 16 on sdumont1405 (hostname sdumont1405)
Process 14 of 16 on sdumont1405 (hostname sdumont1405)
Process 15 of 16 on sdumont1405 (hostname sdumont1405)
Process 4 of 16 on sdumont1405 (hostname sdumont1405)
Process 6 of 16 on sdumont1405 (hostname sdumont1405)
Process 8 of 16 on sdumont1405 (hostname sdumont1405)
Process 11 of 16 on sdumont1405 (hostname sdumont1405)
Process 2 of 16 on sdumont1405 (hostname sdumont1405)
Process 5 of 16 on sdumont1405 (hostname sdumont1405)
Process 9 of 16 on sdumont1405 (hostname sdumont1405)
Process 0 of 16 on sdumont1405 (hostname sdumont1405)
Process 1 of 16 on sdumont1405 (hostname sdumont1405)
Process 10 of 16 on sdumont1405 (hostname sdumont1405)
Process 3 of 16 on sdumont1405 (hostname sdumont1405)
Process 7 of 16 on sdumont1405 (hostname sdumont1405)
```

INFORMAÇÕES sacct

```
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sacct --
format=JOBID,NNODES,NCPUS,NTASKS,ELAPSED,CPUTIME -j 409639
       JobID   NNodes      NCPUS   NTasks    Elapsed    CPUTime
------------ -------- ---------- -------- ---------- ----------
409639              1         24            00:00:04   00:01:36
409639.batch        1         24        1   00:00:04   00:01:36
409639.0            1         16       16   00:00:02   00:00:32
```

(b) 16 ranks, 2 nós (8 em cada nó)

SCRIPT pname2.srm

```
#!/bin/bash
# Script baseado em:
#   "6.3. Jobs paralelos (threads/OpenMP)
#   Forma Geral de um Script"
#   <https://sdumont.lncc.br/support_manual.php?pg=support>
# Uso:
#   $ sbatch pname.srm          (roda em prj)(anotar o my_job_no)
#   $ squeue --job my_job_no    (verifica jobs)
#   A saída "slurm-my_job_no.out" aparece em "scracth"

#SBATCH --nodes=2               #Numero de Nós  (a)
#SBATCH --ntasks-per-node=8     #Numero de tarefas por Nó  (b)
#SBATCH --ntasks=16             #Numero total de tarefas MPI  (a x b)
#SBATCH -p cpu_dev              #Fila (partition) a ser utilizada
```

```
#SBATCH -J pname                #Nome do job
#SBATCH --time=00:02:00         #Tempo limite
#SBATCH --exclusive             #Utilizacao exclusiva dos nós durante o job

#Exibe os nós alocados para o Job
echo $SLURM_JOB_NODELIST
nodeset -e $SLURM_JOB_NODELIST

cd $SLURM_SUBMIT_DIR

#Configura os compiladores com intel MPI PSXE
source /scratch/app/modulos/intel-psxe-2019.sh
#Configura I_MPI_PMI_LIBRARY para apontar para a biblioteca \
#"Process Management Interface" do Slurm
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so

#Configura o executavel
EXEC=/scratch/padinpe/xxxxxxx.xxxxxxx/pname

#exibe informações sobre o executável (opcional)
#   /usr/bin/ldd $EXEC

#Dispara a execução
#   $SLURM_NTASKS : Same as --ntasks
#   $SLURM_CPUS_PER_TASK : Same as --cpus-per-task
srun    -n $SLURM_NTASKS $EXEC
```

SAÍDA slurm-409641.out

```
sdumont[1405-1406]
sdumont1405 sdumont1406
Intel(R) Parallel Studio XE 2019 Update 3 for Linux*
Copyright (C) 2009-2019 Intel Corporation. All rights reserved.
Process 1 of 16 on sdumont1405 (hostname sdumont1405)
Process 2 of 16 on sdumont1405 (hostname sdumont1405)
Process 3 of 16 on sdumont1405 (hostname sdumont1405)
Process 4 of 16 on sdumont1405 (hostname sdumont1405)
Process 5 of 16 on sdumont1405 (hostname sdumont1405)
Process 6 of 16 on sdumont1405 (hostname sdumont1405)
Process 7 of 16 on sdumont1405 (hostname sdumont1405)
Process 0 of 16 on sdumont1405 (hostname sdumont1405)
Process 8 of 16 on sdumont1406 (hostname sdumont1406)
Process 10 of 16 on sdumont1406 (hostname sdumont1406)
Process 11 of 16 on sdumont1406 (hostname sdumont1406)
Process 12 of 16 on sdumont1406 (hostname sdumont1406)
Process 13 of 16 on sdumont1406 (hostname sdumont1406)
Process 14 of 16 on sdumont1406 (hostname sdumont1406)
Process 15 of 16 on sdumont1406 (hostname sdumont1406)
Process 9 of 16 on sdumont1406 (hostname sdumont1406)
```

INFORMAÇÕES sacct

```
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sacct --
format=JOBID,NNODES,NCPUS,NTASKS,ELAPSED,CPUTIME -j 409641
       JobID   NNodes      NCPUS   NTasks    Elapsed    CPUTime
------------ -------- ---------- -------- ---------- ----------
409641              2         48            00:00:03   00:02:24
409641.batch        1         24        1   00:00:03   00:01:12
409641.0            2         16       16   00:00:01   00:00:16
```

(c) 16 ranks, 4 nós (4 em cada nó)

SCRIPT pname4.srm

```
#!/bin/bash
# Script baseado em:
#   "6.3. Jobs paralelos (threads/OpenMP)
#   Forma Geral de um Script"
#   <https://sdumont.lncc.br/support_manual.php?pg=support>
# Uso:
```

```
#   $ sbatch pname.srm          (roda em prj)(anotar o my_job_no)
#   $ squeue --job my_job_no    (verifica jobs)
#   A saída "slurm-my_job_no.out" aparece em "scracth"

#SBATCH --nodes=4               #Numero de Nós   (a)
#SBATCH --ntasks-per-node=4     #Numero de tarefas por Nó   (b)
#SBATCH --ntasks=16             #Numero total de tarefas MPI   (a x b)
#SBATCH -p cpu_dev              #Fila (partition) a ser utilizada
#SBATCH -J pname                #Nome do job
#SBATCH --time=00:02:00         #Tempo limite
#SBATCH --exclusive             #Utilizacao exclusiva dos nós durante o job

#Exibe os nós alocados para o Job
echo $SLURM_JOB_NODELIST
nodeset -e $SLURM_JOB_NODELIST

cd $SLURM_SUBMIT_DIR

#Configura os compiladores com intel MPI PSXE
source /scratch/app/modulos/intel-psxe-2019.sh
#Configura I_MPI_PMI_LIBRARY para apontar para a biblioteca \
#"Process Management Interface" do Slurm
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so

#Configura o executavel
EXEC=/scratch/padinpe/xxxxxxx.xxxxxxx/pname

#exibe informações sobre o executável (opcional)
#   /usr/bin/ldd $EXEC

#Dispara a execução
#   $SLURM_NTASKS : Same as --ntasks
#   $SLURM_CPUS_PER_TASK : Same as --cpus-per-task
srun    -n $SLURM_NTASKS $EXEC
```

SAÍDA slurm-409642.out

```
sdumont[1072,1405-1407]
sdumont1072 sdumont1405 sdumont1406 sdumont1407
Intel(R) Parallel Studio XE 2019 Update 3 for Linux*
Copyright (C) 2009-2019 Intel Corporation. All rights reserved.
Process 1 of 16 on sdumont1072 (hostname sdumont1072)
Process 2 of 16 on sdumont1072 (hostname sdumont1072)
Process 3 of 16 on sdumont1072 (hostname sdumont1072)
Process 0 of 16 on sdumont1072 (hostname sdumont1072)
Process 12 of 16 on sdumont1407 (hostname sdumont1407)
Process 9 of 16 on sdumont1406 (hostname sdumont1406)
Process 4 of 16 on sdumont1405 (hostname sdumont1405)
Process 10 of 16 on sdumont1406 (hostname sdumont1406)
Process 5 of 16 on sdumont1405 (hostname sdumont1405)
Process 11 of 16 on sdumont1406 (hostname sdumont1406)
Process 13 of 16 on sdumont1407 (hostname sdumont1407)
Process 7 of 16 on sdumont1405 (hostname sdumont1405)
Process 8 of 16 on sdumont1406 (hostname sdumont1406)
Process 14 of 16 on sdumont1407 (hostname sdumont1407)
Process 6 of 16 on sdumont1405 (hostname sdumont1405)
Process 15 of 16 on sdumont1407 (hostname sdumont1407)
```

INFORMAÇÕES sacct

```
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sacct --
format=JOBID,NNODES,NCPUS,NTASKS,ELAPSED,CPUTIME -j 409642
       JobID   NNodes      NCPUS   NTasks    Elapsed    CPUTime
------------ -------- ---------- -------- ---------- ----------
409642              4         96            00:00:02   00:03:12
409642.batch        1         24        1   00:00:02   00:00:48
409642.0            4         16       16   00:00:01   00:00:16
```

EXERCÍCIO 2

Testando na máquina local

4 THREADS

```
$ gcc -fopenmp -o stream stream.c
$ ./stream
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 4
Number of Threads counted = 4
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 9428 microseconds.
   (= 9428 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           17161.6     0.009529     0.009323     0.009906
Scale:          16961.7     0.009661     0.009433     0.009832
Add:            19587.0     0.012440     0.012253     0.012585
Triad:          19531.5     0.012558     0.012288     0.013081
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

1 THREAD

```
$ export OMP_NUM_THREADS=1
$ ./stream
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 1
Number of Threads counted = 1
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 22201 microseconds.
   (= 22201 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
```

```
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:            7283.7     0.022129     0.021967     0.022497
Scale:           8336.0     0.019357     0.019194     0.019493
Add:            10991.0     0.021936     0.021836     0.022053
Triad:          10375.3     0.023483     0.023132     0.023876
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

COMPILANDO NO NÓ DE ACESSO

```
[xxxxxxx.xxxxxxx@sdumont13 ~]$ module load intel_psxe/2019
[xxxxxxx.xxxxxxx@sdumont13 ~]$ mpiicc -qopenmp -o stream stream.c
[xxxxxxx.xxxxxxx@sdumont13 ~]$ ls
pname  pname1.srm  pname2.srm  pname4.srm  pname.c  stream  stream1.srm  stream.c
```

SCRIPT

```
#!/bin/bash
#Uso:
#   $ sbatch stream1.srm    (roda em prj)(anotar o my_job_no)
#   $ squeue --job my_job_no  (verifica jobs)
#   saída slurm-my_job_no.out em scracth
#SBATCH --nodes=1            # utilizar um único nó
#SBATCH --ntasks-per-node=1  # ntasks(max tasks) invoked on each node
#SBATCH --ntasks=1           # no total de processos MPI
#SBATCH --cpus-per-task=1    # number of threads (max 24)
#SBATCH -p cpu_dev           # Fila (partition) ate 24 cores
#SBATCH -J list06            # Nome do job
#SBATCH --time=00:02:00      # Altera o tempo limite para 2 minutos
#SBATCH --exclusive          # Utilização exclusiva dos nós durante o job

# Exibe os nós alocados para o Job
echo $SLURM_JOB_NODELIST
nodeset -e $SLURM_JOB_NODELIST

cd $SLURM_SUBMIT_DIR

# Config. compiladores p/ suite de compiladores da Intel e MPI compilado com
Intel
module load intel_psxe/2019

# Configura o executável
EXEC1=/scratch/padinpe/xxxxxxx.xxxxxxx/stream

# exibe informações sobre o executável (opcional)
#/usr/bin/ldd $EXEC1

# Configura o número de threads conforme o parâmetro acima do SLURM
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

# Imprime o número de threads
echo "SLURM_CPUS_PER_TASK" $SLURM_CPUS_PER_TASK

# Executa com 1 processo e define número de threads com a opção "-c":
srun -N 1 -c $SLURM_CPUS_PER_TASK $EXEC1
```

(a) 1 THREAD

```
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sbatch ./stream1.srm
Submitted batch job 409689
```

RESULTADO slurm-409689.out

```
sdumont1082
sdumont1082
```

```
SLURM_CPUS_PER_TASK 1
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 1
Number of Threads counted = 1
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 8698 microseconds.
   (= 8698 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:            9214.5     0.017387     0.017364     0.017440
Scale:           7195.8     0.022277     0.022235     0.022345
Add:             9415.8     0.025523     0.025489     0.025586
Triad:           9521.9     0.025298     0.025205     0.025561
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

(b) 2, 4, 8, 12, 16, 20, 24 THREADS

```
[xxxxxxx.xxxxxxx@sdumont13 ~]$ sbatch ./stream2.srm
Submitted batch job 409694
```

RESULTADO slurm-409694.out

```
sdumont1082
sdumont1082
SLURM_CPUS_PER_TASK 2
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 2
Number of Threads counted = 2
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 4908 microseconds.
   (= 4908 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           17588.0     0.009176     0.009097     0.009228
```

```
Scale:          12475.6     0.012885     0.012825     0.012924
Add:            16438.3     0.014661     0.014600     0.014722
Triad:          16586.2     0.014528     0.014470     0.014574
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

RESULTADO COM 4 THREADS slurm-409697.out

```
sdumont1083
sdumont1083
SLURM_CPUS_PER_TASK 4
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 4
Number of Threads counted = 4
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 3397 microseconds.
   (= 3397 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           28174.5     0.005712     0.005679     0.005745
Scale:          22321.3     0.007431     0.007168     0.007747
Add:            28718.3     0.008612     0.008357     0.008921
Triad:          29073.3     0.008563     0.008255     0.008844
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

RESULTADO COM 8 THREADS slurm-409698.out

```
sdumont1083
sdumont1083
SLURM_CPUS_PER_TASK 8
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 8
Number of Threads counted = 8
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 3149 microseconds.
   (= 3149 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
```

```
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           34136.5     0.004757     0.004687     0.004836
Scale:          38138.7     0.004248     0.004195     0.004292
Add:            46360.8     0.005215     0.005177     0.005240
Triad:          46584.0     0.005179     0.005152     0.005209
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

RESULTADO COM 12 THREADS slurm-409699.out

```
sdumont1083
sdumont1083
SLURM_CPUS_PER_TASK 12
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 12
Number of Threads counted = 12
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 2840 microseconds.
   (= 2840 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           34020.5     0.004733     0.004703     0.004785
Scale:          41025.1     0.003953     0.003900     0.004111
Add:            48454.1     0.004995     0.004953     0.005028
Triad:          49039.5     0.005420     0.004894     0.009340
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

RESULTADO COM 16 THREADS slurm-409700.out

```
sdumont1083
sdumont1083
SLURM_CPUS_PER_TASK 16
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 16
Number of Threads counted = 16
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 2443 microseconds.
   (= 2443 clock ticks)
Increase the size of the arrays if this shows that
```

```
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           44310.9     0.004190     0.003611     0.004859
Scale:          26990.4     0.006002     0.005928     0.006074
Add:            36809.6     0.006583     0.006520     0.006689
Triad:          38216.9     0.006441     0.006280     0.006545
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

RESULTADO COM 20 THREADS slurm-409701.out

```
sdumont1083
sdumont1083
SLURM_CPUS_PER_TASK 20
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 20
Number of Threads counted = 20
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 4302 microseconds.
   (= 4302 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           40795.7     0.005159     0.003922     0.005829
Scale:          39938.6     0.005436     0.004006     0.006129
Add:            46074.4     0.007321     0.005209     0.008309
Triad:          48328.4     0.006630     0.004966     0.007233
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

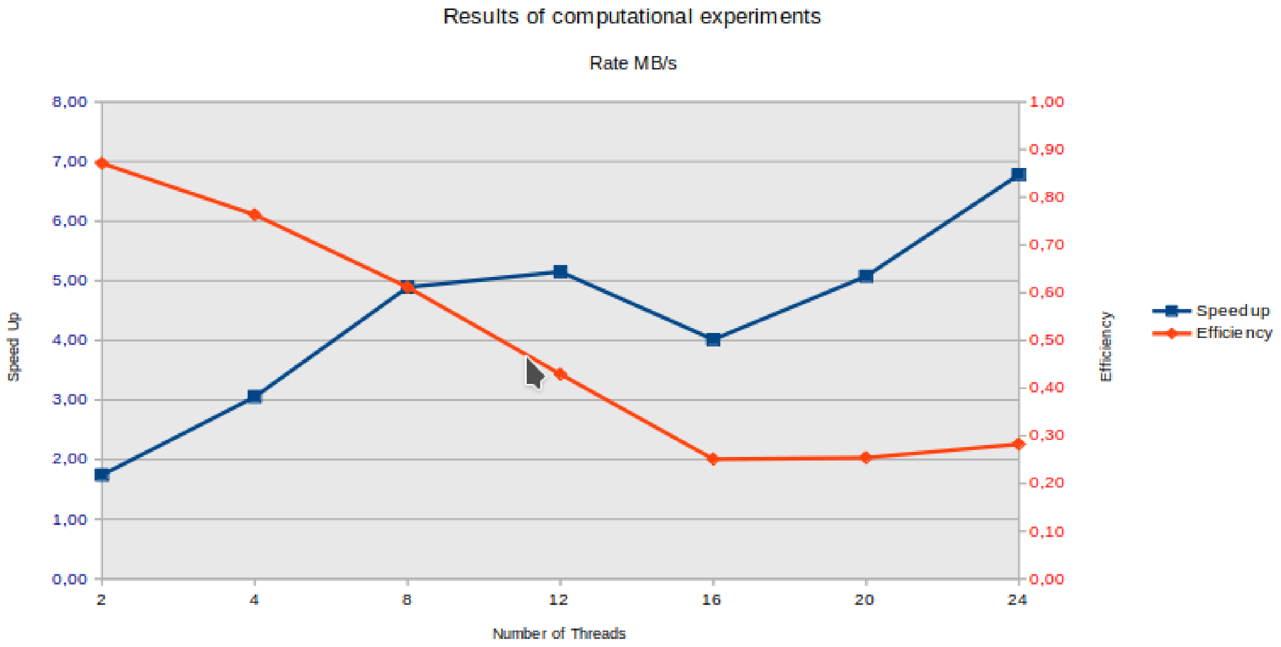
RESULTADO COM 24 THREADS slurm-409702.out

```
sdumont1083
sdumont1083
SLURM_CPUS_PER_TASK 24
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Number of Threads requested = 24
Number of Threads counted = 24
-------------------------------------------------------------
```

```
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 1614 microseconds.
   (= 1614 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           67650.1     0.002864     0.002365     0.004326
Scale:          54533.5     0.003324     0.002934     0.003680
Add:            64639.6     0.004064     0.003713     0.004326
Triad:          64552.6     0.003921     0.003718     0.004440
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

(c) TAXAS DE DESEMPENHO

Function	1	2			4			8			12			16			20			24		
	Thread	Rate	Speed Up	Efficiency	Rate	Speed Up	Efficiency	Rate	Speed Up	Efficiency	Rate	Speed Up	Efficiency	Rate	Speed Up	Efficiency	Rate	Speed Up	Efficiency	Rate	Speed Up	Efficiency
Triad	9522	16586	1,74	0,87	29073	3,05	0,76	46584	4,89	0,61	49040	5,15	0,43	38217	4,01	0,25	48328	5,08	0,25	64553	6,78	0,28



REFERÊNCIAS

- http://www.lac.inpe.br/~xxxx/cap399-2019/
- http://www.lac.inpe.br/~stephan/
- https://sdumont.lncc.br/support_manual.php
- https://slurm.schedmd.com/documentation.html
- https://stackoverflow.com
- https://software.intel.com/en-us/cpp-compiler-developer-guide-and-reference-compiler-reference