Introdução ao ambiente SDUMONT/SLURM

Escola Santos Dumont
22 de janeiro 2024
LNCC
evento remoto online

Laboratório Nacional de Computação Científica

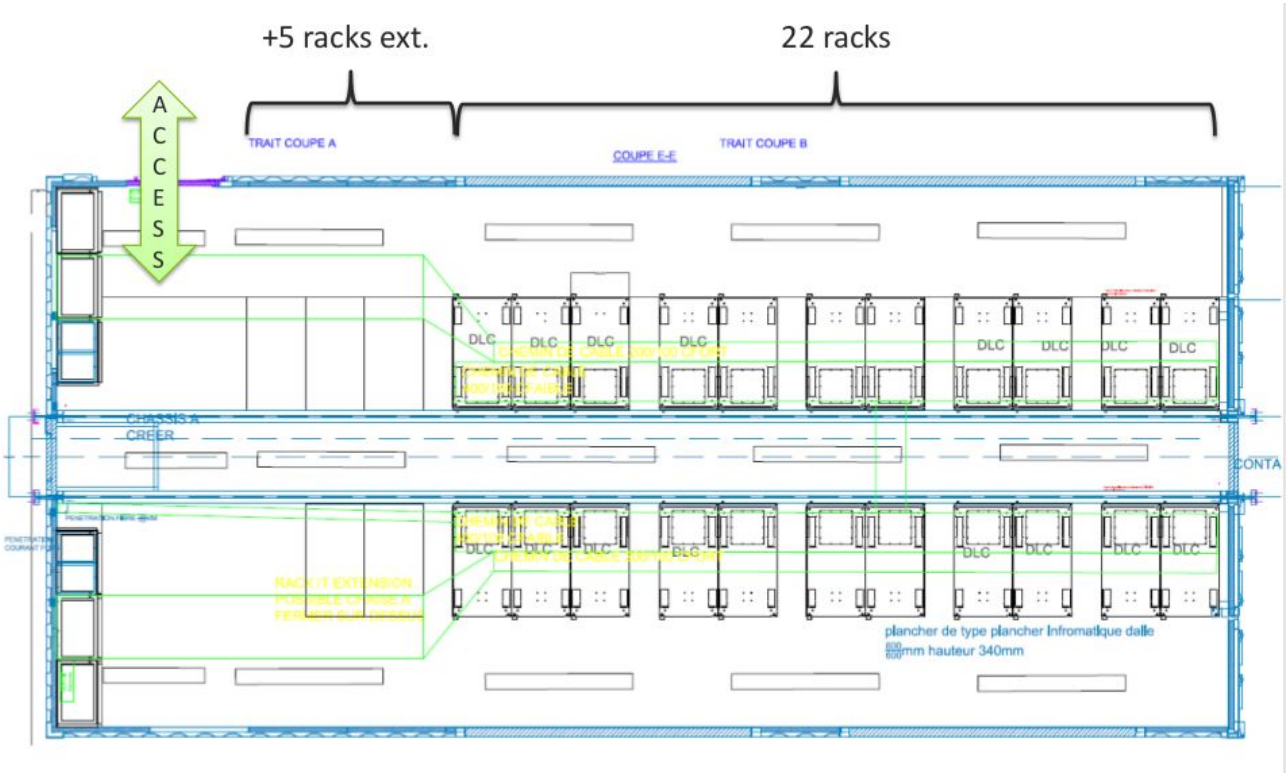
A máquina

Mobull - solução para datacenter baseada em containers.

Plug & Boot

2 containers com 22 racks 42U

A máquina



Laboratório Nacional de Computação Científica

Arquitetura

Cluster de propósito geral

3 tipos de nodes:

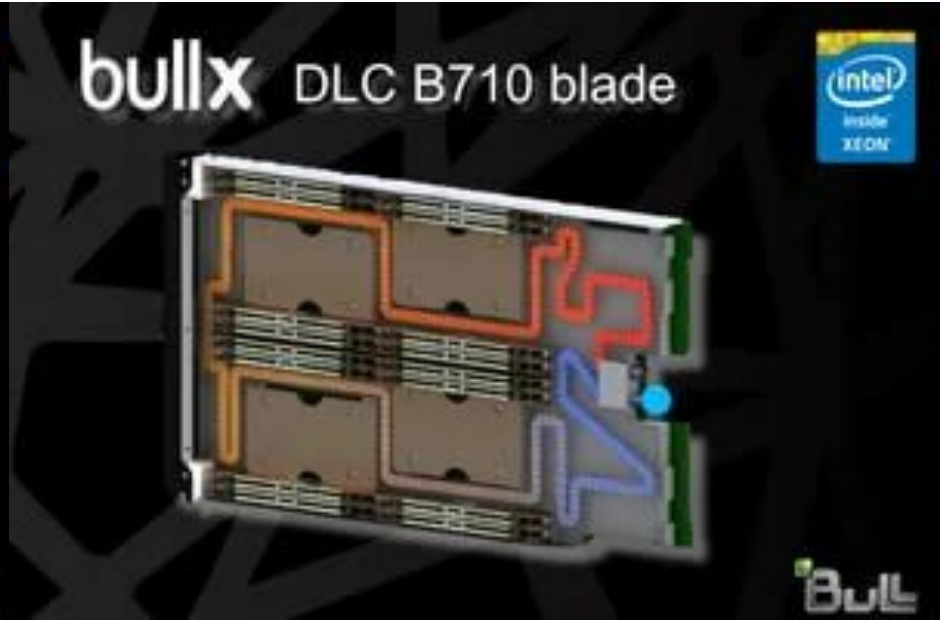
- thin nodes
- hybrid nodes
- fat-node

Arquitetura

Thin nodes (B710)

- 7 racks completos (504 nós computacionais)
- 2 nós computacionais por blade
- Configuração
  - 2x Intel Xeon E5-2695v2 (12c, 2.4Ghz)
  - 64 GB DDR3 RAM (8x 8GB DIMM)
  - 1x 120GB SSD disk
  - 1x Infiniband FDR ConnectX3
  - 1x GbE

Arquitetura



Arquitetura

Thin nodes (Bull Sequana) - Machine Learning/Deep Learning

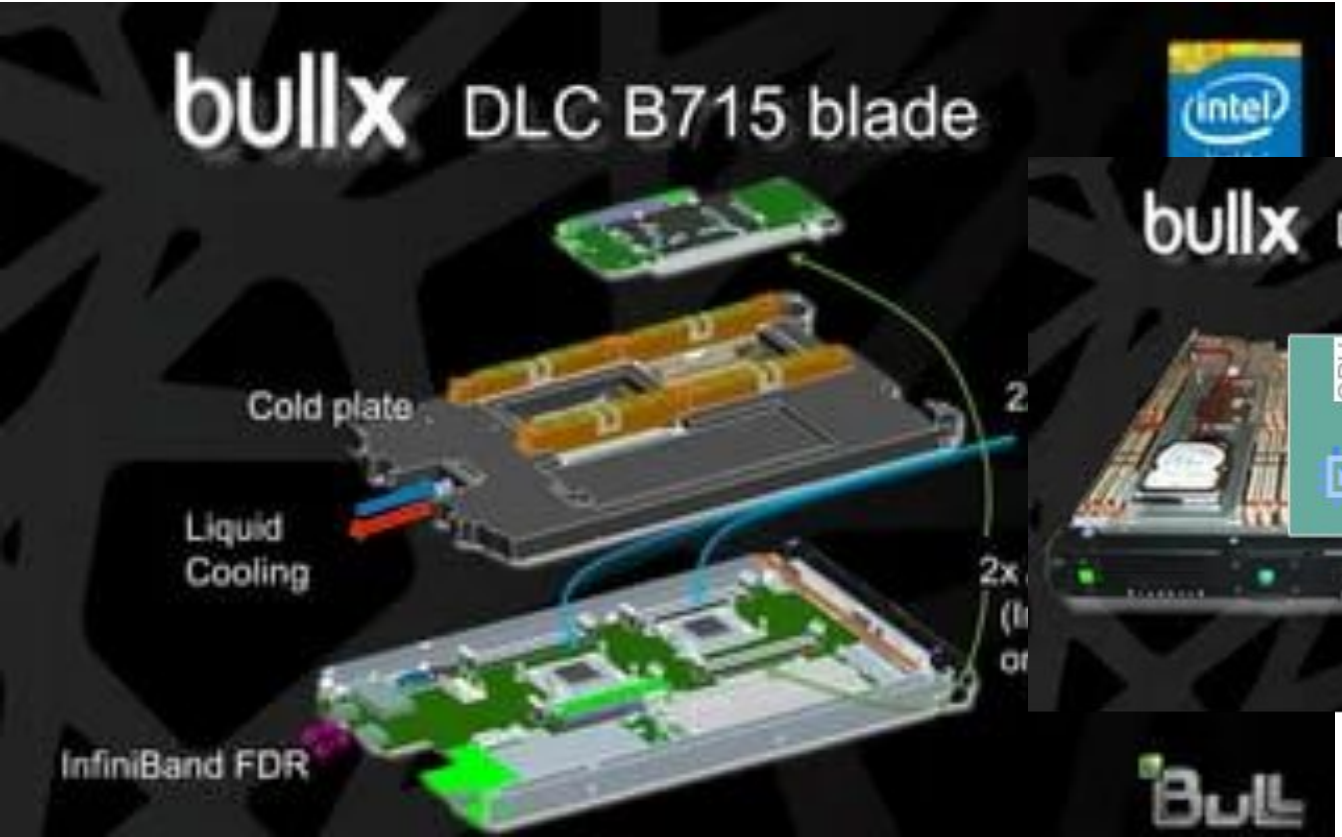
- 1 nó computacional
- Configuração
  - 2x Intel Skylake GOLD 6148, 2,4Ghz (20c)
  - 384 GB DDR4
  - 4x Infiniband EDR 100Gbps
  - 8x NVidia V100 com NVLink

Arquitetura

Hybrid nodes (B715)

- 7 racks completos (252 nós computacionais e 504 aceleradores)
- 198 nodes e 396 nVidia K40
- 54 nodes e 108 Intel Phi 7120P -> nodes CPU

Arquitetura



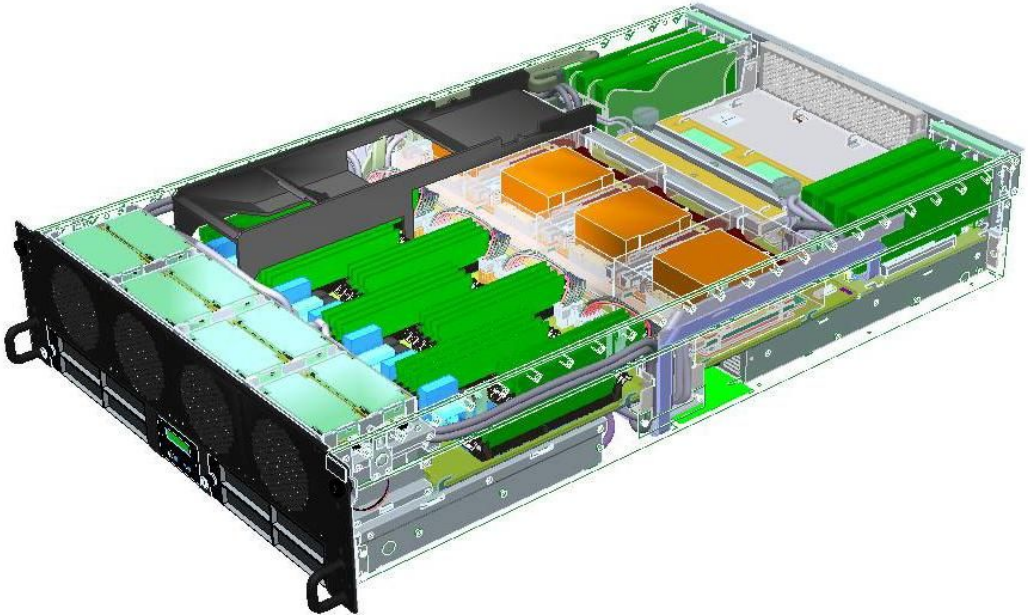
Laboratório Nacional de Computação Científica

Arquitetura

Fat-node (S6130)

- 16x Intel Ivy Bridge E7 2870v2 15c 2.3Ghz
- 6TB DDR3 RAM
- 1x 120GB SSD disk
- 1x Infiniband FDR ConnectX3
- 1x GbE

Arquitetura



Arquitetura

Login nodes

- 4x bullx R423-E3
- Linux Virtual Server (LVS)
- Cada login node possui:
  - 2x E5-2695v2 12c, 2.4GHz
  - 128 GB DDR3@1866RAM
  - 2x 500GB 7.2krpm SATA2 RAID1
  - 1x GbE network port
  - 1x IB FDR network port
  - 1x Ethernet BMC network port
  - 2x 10GbE network ports

Arquitetura

Armazenamento

- Lustre - Seagate ClusterStor 9000 - /scratch
  - Total 1,7 Petabytes
- DellEMC Isilon - /prj
  - Total 650 Terabytes

Estrutura de diretórios

Diretório home ($HOME):

- NFS - Acessível apenas nos login nodes
- /prj/**PROJETO**/login.name

Diretório de scratch ($SCRATCH):

- Lustre - Acessível a todos os nodes do cluster
- /scratch/**PROJETO**/login.name

Desempenho

- GPU - 456,8 TFlop/s
- PHI - 363,2 TFlop/s
- CPU - 321,2 TFlop/s
- Total - 1.141,2 TFlop/s

TOP 500	Total	GPU	PHI	CPU
Jun/15	55	145	177	207
Nov/15	63	200	265	310
Jun/16	75	265	364	433
Nov/16	91	364	476	
Jun/17	107	472		
Nov/17	128			
Jun/18	192			
Nov/18	316			

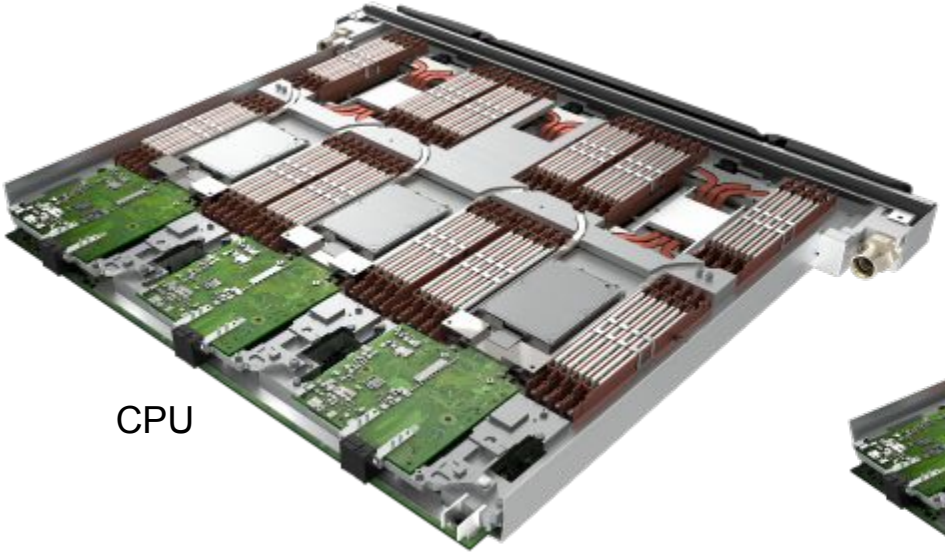
https://www.top500.org

Expansão SDumont - 2018/2019

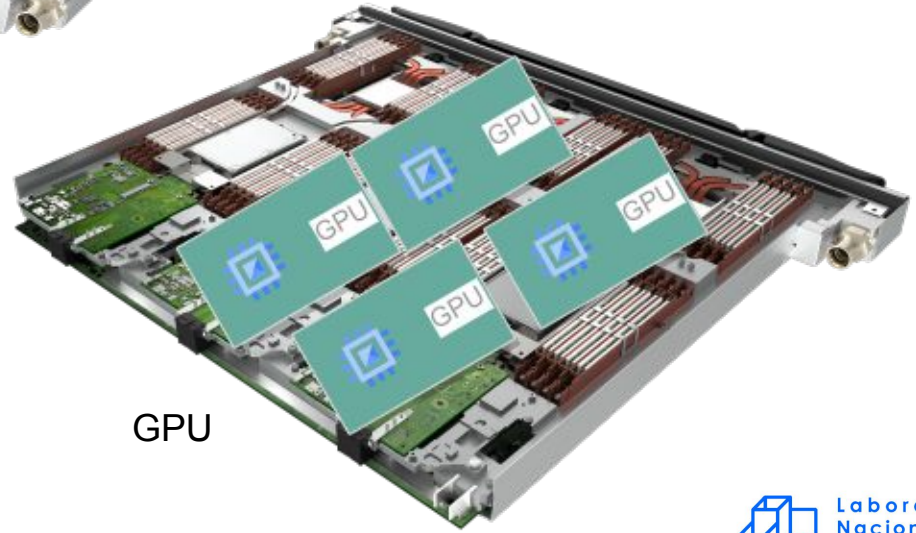
- 1 Célula Sequana X1000 CPU
  - 82 Blades X1120 - 384 GB
  - 12 Blades X1120 - 768 GB
  - 282 nós computacionais
  - 2x Intel CascadeLake Gold 6252 (24c)

- 1 Célula Sequena X1000 GPU
  - 94 Blades X1125 - 384 GB
  - 1 nó computacional e 4 aceleradores NVIDIA Volta V100 GPU por blade
  - 2x Intel CascadeLake Gold 6252 (24c)

Expansão SDumont - 2018/2019



CPU



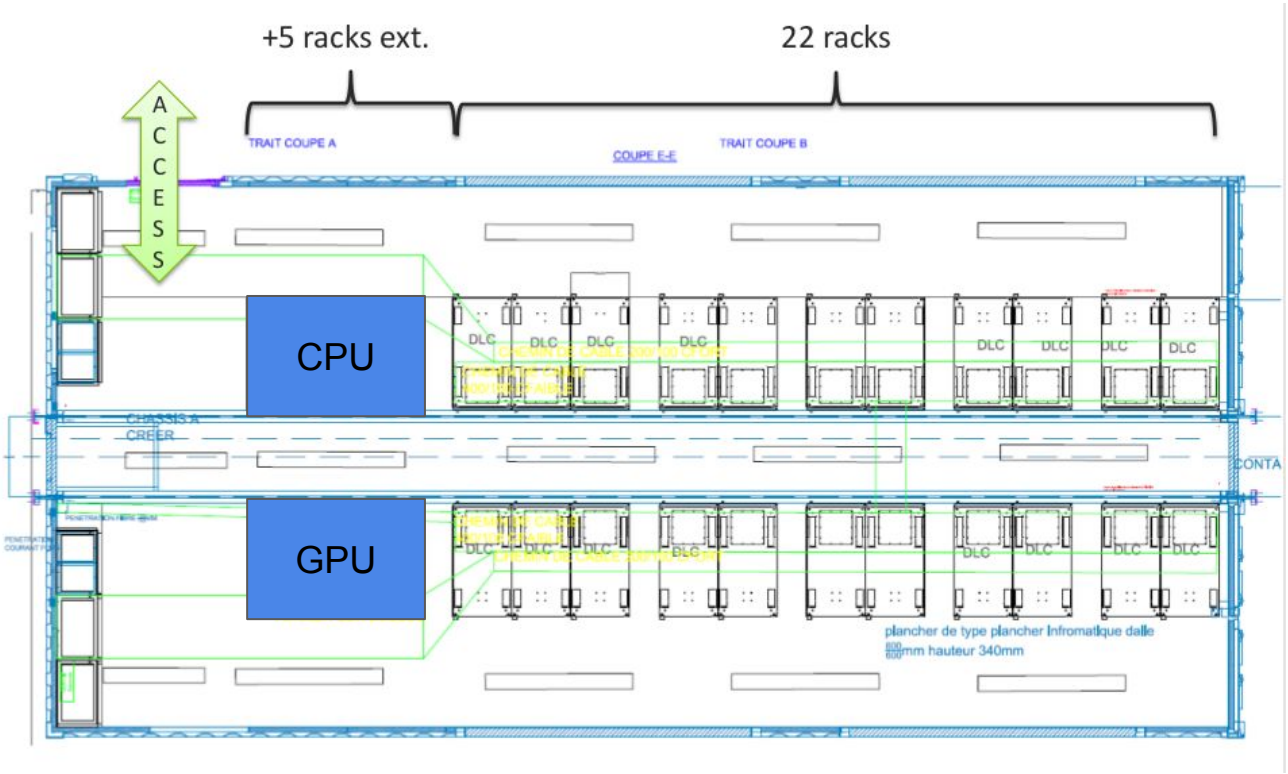
GPU

Expansão SDumont

Login nodes

- 4x Login nodes:
  - 2x Intel Xeon Gold 6152 22c, 2.1GHz
  - 756 GB DDR3@1866RAM
  - 2x SSD MZ7LM960
  - 1x GbE network port
  - 2x IB FDR network port
  - 2x 10GbE network ports

Expansão SDumont



Laboratório Nacional de Computação Científica

Desempenho

Novembro 2019

			Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
193	Laboratório Nacional de Computação Científica Brazil	**Santos Dumont (SDumont)** - Bull Sequana X1000, Xeon Gold 6252 24C 2.1GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2 Atos	33,856	1,849.0	2,727.0

Junho 2020

			Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
240	Laboratório Nacional de Computação Científica Brazil	**Santos Dumont (SDumont)** - Bull Sequana X1000, Xeon Gold 6252 24C 2.1GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Atos	33,856	1,849.0	2,727.0

Novembro 2020

			Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
276	Laboratório Nacional de Computação Científica Brazil	**Santos Dumont (SDumont)** - Bull Sequana X1000, Xeon Gold 6252 24C 2.1GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Atos	33,856	1,849.0	2,727.0

Desempenho

Novembro 2022

		Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
462	**Santos Dumont (SDumont)** - Bull Sequana X1000, Xeon Gold 6252 24C 2.1GHz, Mellanox InfiniBand EDR, NVIDIA Tesla V100 SXM2, Atos Laboratório Nacional de Computação Científica Brazil	33,856	1.85	2.73

Novembro 2023

1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
500	**TX-Green2** - PowerEdge C6420, Xeon Platinum 8260 24C 2.4GHz, 25G Ethernet, ACTION MIT Lincoln Laboratory Supercomputing Center United States	43,200	2.02	53.08	

Filas

Filas SDBASE (24 núcleos/nó)	Wall-clock	Nodes	Núcleos	Execução	Na fila
cpu_dev	20 min	1-4	1-96	1	1
nvidia_dev	20 min	1-4	1-96	1	1

Filas SEQUANA (48 núcleos/nó)	Wall-clock	Nodes	Núcleos	Execução	Na fila
sequana_cpu_dev	20 min	1-4	1-192	1	1
sequana_gpu_dev	20 min	1-4	1-192	1	1

Módulos de ambiente

module avail : Lista todos as aplicações (módulos) disponíveis

module whatis/help <app>/<versão> : Exibe uma ajuda sobre a aplicação (módulo)

module load <app>/<versão> : Carrega o módulo (já carrega as dependências)

module unload <app>/<versão> : Descarrega o módulo

module list : Lista os módulos carregados

Intel Parallel Studio

source /scratch/app/modulos/intel-psxe-20[16|17|18|19|20].sh
(também tem módulo = intel_psxe/<versão>)

Compiladores

GNU

Versões: 4.8.5, 6.5, 7.4, 8.3, 9.3, 10.2 e 11.1

INTEL

Versões: 2016, 2017, 2018, 2019 e 2020

PGI

Versão 2016.5 e 2019.10 (Community) - **Expirado**! Possível instalar novas versões, caso necessário.

Implementações MPI

OpenMPI (Bull)

Versão 4.0.3 - Implementação MPI compilada pela Bull

OpenMPI

Versões: 1.8.6, 1.10.7, 2.0.x, 2.1.x, 3.1.x, 4.0.x, 4.1.x e 5.0.0

Intel MPI

Versões 5.1, 2016, 2017, 2018, 2019 e 2020

Slurm

Versão 20.11.8

Onde encontrar referências?

http://sdumont.lncc.br/

https://slurm.schedmd.com/archive/slurm-20.11.8

Política de escalonamento

Backfill: prioridade, tempo na fila, recursos solicitados e etc.

Acesso

Somente quem já possuir conta no SDumont poderá acessar o ambiente.

```
$ ssh meu.login@login.sdumont.lncc.br
```

Não serão distribuídas credenciais “genéricas”.

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

`sacctmgr:` lista acesso às filas

Lista as filas que o usuário tem acesso (entre outras coisas):

```
$ sacctmgr list user $USER -s format=account,partition%30,maxjobs,maxnodes,maxcpus,maxsubmit,maxwall

   Account                      Partition MaxJobs MaxNodes  MaxCPUs MaxSubmit     MaxWall
---------- ------------------------------ ------- -------- -------- --------- -----------
      xpto             sequana_gpu_shared       4       15      720        24  4-00:00:00
      xpto                sequana_gpu_dev       1        4      192         1    00:20:00
      xpto                   nvidia_small       4       20      480        24    01:00:00
      xpto                    nvidia_long       2       10      240         4 31-00:00:00
      xpto                    nvidia_scal       1      128     3072         8    18:00:00
      xpto                     nvidia_dev       1        4       96         1    00:20:00
      xpto                         nvidia       4       50     1200        24  2-00:00:00
      xpto             sequana_cpu_shared       4       50     2400        24  4-00:00:00
      xpto                sequana_cpu_dev       1        4      192         1    00:20:00
      xpto                     cpu_shared      16       20      480        96  3-00:00:00
      xpto                      cpu_small      16       20      480        96  3-00:00:00
      xpto                       cpu_long       3       10      240        18 31-00:00:00
      xpto                       cpu_scal       1      128     3072         8    18:00:00
      xpto                        cpu_dev       1        4       96         1    00:20:00
      xpto                            cpu       4       50     1200        24  4-00:00:00
```

*[1]Lista resumida: sacctmgr list user $USER -s format=partition%30
*[2]A variável de ambiente "**$USER**" possui o "**login**" do próprio usuário executando o comando.

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

Comandos básicos

sinfo

```
$ sinfo -s

PARTITION       AVAIL  TIMELIMIT   NODES(A/I/O/T)  NODELIST
cpu_dev            up   infinite   124/180/38/342  sdumont[1000-1009,...,1414-1503]
nvidia_dev         up   infinite     182/11/0/193  sdumont[3000-3035,...,3190-3197]

sequana_cpu_dev    up      20:00      61/51/0/112  sdumont[6068-6084,...,6279-6287]
sequana_gpu_dev    up      20:00       31/10/1/42  sdumont[8044-8055,...,8093-8095]
```

Comandos básicos

sinfo

```
$ sinfo -s

PARTITION       AVAIL  TIMELIMIT   NODES(A/I/O/T)  NODELIST
cpu*               up   infinite    350/229/7/586  sdumont[1000-1503,3110-3159,5012-5043]
nvidia             up   infinite     119/71/2/192  sdumont[3006-3197]
phi                up   infinite       13/19/0/32  sdumont[5012-5043]
mesca2             up   infinite          0/1/0/1  sdumont57
cpu_small          up   infinite    350/229/7/586  sdumont[1000-1503,3110-3159,5012-5043]
nvidia_small       up   infinite     119/71/2/192  sdumont[3006-3197]
cpu_dev            up   infinite    287/220/7/514  sdumont[1000-1503,5044-5053]
nvidia_dev         up   infinite     119/71/2/192  sdumont[3006-3197]
phi_dev            up   infinite       13/19/0/32  sdumont[5012-5043]
cpu_scal           up   infinite    350/229/7/586  sdumont[1000-1503,3110-3159,5012-5043]
cpu_long           up   infinite    300/229/7/536  sdumont[1000-1503,5012-5043]
nvidia_scal        up   infinite     119/71/2/192  sdumont[3006-3197]
nvidia_long        up   infinite     119/71/2/192  sdumont[3006-3197]
all             inact 2-00:00:00    419/328/9/756  sdumont[1000-1503,3000-3197,5000-5053]
```

Comandos básicos

sinfo

Outras opções:

- -s
- --long
- --state
- -R

Comandos básicos

squeue

Outras opções

- -s
- -u (user)
- -A (account)
- -p

Comandos básicos

squeue

```
$ squeue

JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
 1767       cpu mpiblast  labinfo  R    9:40:13      1 sdumont1128
 1769       cpu mpiblast  labinfo  R    9:35:10      1 sdumont1130
 1770       cpu mpiblast  labinfo  R    9:33:36      2 sdumont[1000-1001]
 1772       cpu mpiblast  labinfo  R    9:29:48      2 sdumont[1106-1107]
 1777       cpu brams-5. xrpsouto  R    1:12:10      1 sdumont1126
 1776    mesca2 TEST_bla  labinfo  R    8:21:18      1 sdumont57
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

srun

```
$ srun -p cpu_dev --nodes=1 --ntasks=6 --cpus-per-task=1 sleep 60

$ squeue -u $USER
      JOBID PARTITION  NAME     USER     ST       TIME  NODES NODELIST(REASON)
   10756971 cpu_dev    sleep    rpsouto  R        0:40      3 sdumont1189

$ scontrol --details show job 10756932

   NumNodes=1 NumCPUs=6 NumTasks=6 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   Nodes=sdumont1189 CPU_IDs=0-5 Mem=64000
```

srun

```
$ srun -p cpu_dev   -N1        -n6       -c1    sleep 60 (forma compacta)

$ squeue -u $USER
      JOBID PARTITION  NAME     USER     ST       TIME  NODES NODELIST(REASON)
   10756971 cpu_dev    sleep    rpsouto  R        0:40      3 sdumont1189

$ scontrol --details show job 10756932

   NumNodes=1 NumCPUs=6 NumTasks=6 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   Nodes=sdumont1189 CPU_IDs=0-5 Mem=64000
```

srun

```
$ scontrol --details show job 10756971
JobId=10756971 JobName=sleep
   UserId=rpsouto(60879) GroupId=cenapadrjsd(61071) MCS_label=N/A
   Priority=5116 Nice=0 Account=lncc QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=0 Reboot=0 ExitCode=0:0
   DerivedExitCode=0:0
   RunTime=00:00:26 TimeLimit=00:20:00 TimeMin=N/A
   SubmitTime=2023-01-16T02:23:26 EligibleTime=2023-01-16T02:23:26
   AccrueTime=2023-01-16T02:23:26
   StartTime=2023-01-16T02:23:33 EndTime=2023-01-16T02:43:33 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2023-01-16T02:23:33
   Partition=cpu_dev AllocNode:Sid=sdumont11:6575
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=sdumont1189
   BatchHost=sdumont1189
   NumNodes=1 NumCPUs=6 NumTasks=6 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=6,mem=62.50G,node=1,billing=6
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   JOB_GRES=(null)
   Nodes=sdumont1189 CPU_IDs=0-5 Mem=64000 GRES=
   MinCPUsNode=1 MinMemoryNode=62.50G MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=sleep
   WorkDir=/prj/cenapadrjsd/rpsouto
   Power=
   NtasksPerTRES:0
```

Mapeamento (*mapping*) e vinculação (*binding*)

Mapping define como as tarefas são distribuídas:
- no nível de núcleos
- no nível de sockets
- no nível de nós

Binding define a afinidade das tarefas:
- por núcleo
- por socket
- por nó (sem *binding*)

SDBASE


NUMANode P#0 (32GB)
Package P#0
L3 (30MB)
L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#0 PU P#0
Core P#1 PU P#1
Core P#2 PU P#2
Core P#3 PU P#3
Core P#4 PU P#4
Core P#5 PU P#5
Core P#8 PU P#6
Core P#9 PU P#7
Core P#10 PU P#8
Core P#11 PU P#9
Core P#12 PU P#10
Core P#13 PU P#11
NUMANode P#1 (32GB)
Package P#1
L3 (30MB)
L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB) L2 (256KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#0 PU P#12
Core P#1 PU P#13
Core P#2 PU P#14
Core P#3 PU P#15
Core P#4 PU P#16
Core P#5 PU P#17
Core P#8 PU P#18
Core P#9 PU P#19
Core P#10 PU P#20
Core P#11 PU P#21
Core P#12 PU P#22
Core P#13 PU P#23

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N1 -n6 -c1 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

2: cpu-bind=MASK - sdumont1227, task  2  2 [2318]: mask 0x4 set
1: cpu-bind=MASK - sdumont1227, task  1  1 [2317]: mask 0x2 set
3: cpu-bind=MASK - sdumont1227, task  3  3 [2319]: mask 0x8 set
4: cpu-bind=MASK - sdumont1227, task  4  4 [2320]: mask 0x10 set
0: cpu-bind=MASK - sdumont1227, task  0  0 [2316]: mask 0x1 set
5: cpu-bind=MASK - sdumont1227, task  5  5 [2321]: mask 0x20 set


0: Cpus_allowed_list:    0
1: Cpus_allowed_list:    1
2: Cpus_allowed_list:    2
3: Cpus_allowed_list:    3
4: Cpus_allowed_list:    4
5: Cpus_allowed_list:    5
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n6 -c1 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont1227, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont1228, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont1228, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont1228, task  5  2 [25465]: mask 0x4 set


0: Cpus_allowed_list:    0
1: Cpus_allowed_list:    1
2: Cpus_allowed_list:    2
3: Cpus_allowed_list:    0
4: Cpus_allowed_list:    1
5: Cpus_allowed_list:    2
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n6 -c2 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont1227, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont1228, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont1228, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont1228, task  5  2 [25465]: mask 0x4 set


0: Cpus_allowed_list:    0-1
1: Cpus_allowed_list:    2-3
2: Cpus_allowed_list:    4-5
4: Cpus_allowed_list:    2-3
3: Cpus_allowed_list:    0-1
5: Cpus_allowed_list:    4-5
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n6 -c4 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont1227, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont1228, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont1228, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont1228, task  5  2 [25465]: mask 0x4 set


0: Cpus_allowed_list:    0-3
1: Cpus_allowed_list:    4-7
2: Cpus_allowed_list:    8-11
3: Cpus_allowed_list:    0-3
4: Cpus_allowed_list:    4-7
5: Cpus_allowed_list:    8-11
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n6 -c8 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont1227, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont1228, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont1228, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont1228, task  5  2 [25465]: mask 0x4 set


0: Cpus_allowed_list:    0-7
1: Cpus_allowed_list:    8-15
2: Cpus_allowed_list:    16-23
3: Cpus_allowed_list:    0-7
5: Cpus_allowed_list:    16-23
4: Cpus_allowed_list:    8-15
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n6 -c8 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont1227, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont1228, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont1228, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont1228, task  5  2 [25465]: mask 0x4 set


0: Cpus_allowed_list:    0-7
1: Cpus_allowed_list:    8-15  -> núcleos em diferentes sockets
2: Cpus_allowed_list:    16-23
3: Cpus_allowed_list:    0-7
5: Cpus_allowed_list:    16-23
4: Cpus_allowed_list:    8-15  -> núcleos em diferentes sockets
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n8 -c6 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [25051]: mask 0x3f set
1: cpu-bind=MASK - sdumont1227, task  1  1 [25052]: mask 0xfc0 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [25053]: mask 0x3f000 set
3: cpu-bind=MASK - sdumont1227, task  3  3 [25054]: mask 0xfc0000 set
4: cpu-bind=MASK - sdumont1228, task  4  0 [28964]: mask 0x3f set
5: cpu-bind=MASK - sdumont1228, task  5  1 [28965]: mask 0xfc0 set
6: cpu-bind=MASK - sdumont1228, task  6  2 [28966]: mask 0x3f000 set
7: cpu-bind=MASK - sdumont1228, task  7  3 [28967]: mask 0xfc0000 set
0: Cpus_allowed_list:    0-5
1: Cpus_allowed_list:    6-11
2: Cpus_allowed_list:    12-17
3: Cpus_allowed_list:    18-23
4: Cpus_allowed_list:    0-5
5: Cpus_allowed_list:    6-11
6: Cpus_allowed_list:    12-17
7: Cpus_allowed_list:    18-23
```

`srun`: distribuição das tarefas

```
$ srun -p cpu_dev -N2 -n8 -c6 --cpu_bind=cores,verbose --label cat /proc/self/status | grep
Cpus_allowed_list | grep Cpus_allowed_list

0: cpu-bind=MASK - sdumont1227, task  0  0 [25051]: mask 0x3f set
1: cpu-bind=MASK - sdumont1227, task  1  1 [25052]: mask 0xfc0 set
2: cpu-bind=MASK - sdumont1227, task  2  2 [25053]: mask 0x3f000 set
3: cpu-bind=MASK - sdumont1227, task  3  3 [25054]: mask 0xfc0000 set
4: cpu-bind=MASK - sdumont1228, task  4  0 [28964]: mask 0x3f set
5: cpu-bind=MASK - sdumont1228, task  5  1 [28965]: mask 0xfc0 set
6: cpu-bind=MASK - sdumont1228, task  6  2 [28966]: mask 0x3f000 set
7: cpu-bind=MASK - sdumont1228, task  7  3 [28967]: mask 0xfc0000 set
0: Cpus_allowed_list:    0-5
1: Cpus_allowed_list:    6-11
2: Cpus_allowed_list:    12-17
3: Cpus_allowed_list:    18-23
4: Cpus_allowed_list:    0-5
5: Cpus_allowed_list:    6-11
6: Cpus_allowed_list:    12-17
7: Cpus_allowed_list:    18-23
```

Tarefas com núcleos nos mesmos sockets.

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd $SCRATCH -> vai para o diretório de sua conta na partição do lustre (/scratch)
TODA SUBMISSÃO DE JOB DEVE SER FEITA COM EXECUTÁVEIS INSTALADOS NA PARTIÇÃO /scratch
$ pwd $SCRATCH -> verifica o caminho deste diretório

$ module load git/2.23
$ git clone https://github.com/robertopsouto/ESD2024.git
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
ESD2024/
├── README.md
├── sdbase
│   ├── env_openmpi
│   └── NPB3.4.2-MZ
└── sequana
    ├── env_openmpi
    └── NPB3.4.2-MZ
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
ESD2024/
├── README.md
├── sdbase
│   ├── env_openmpi
│   └── NPB3.4.2-MZ
│       ├── NPB3.4-MZ-MPI
│       │   ├── bin
│       │   │   └── BULL_srun_openmpi.sh
│       │   ├── BT-MZ
│       │   ├── config
│       │   │   ├── make.def
│       │   │   ├── suite.def
│       │   ├── Makefile
│       └── README
└── sequana
    ├── env_openmpi
    └── NPB3.4.2-MZ
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd ESD2024/sdbase/
$ ls -A1
env_openmpi
NPB3.4.2-MZ

$ cat env_openmpi
module load openmpi/gnu/4.1.2+cuda-11.2

$ source env_openmpi
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd NPB3.4.2-MZ/NPB3.4-MZ-MPI/config/
$ ls -A1
  make.def
  make.def.template
  NAS.samples/
  suite.def
  suite.def.template
$ cat make.def
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cat suite.def

# config/suite.def
# This file is used to build several benchmarks with a single command.
# Typing "make suite" in the main directory will build all the benchmarks
# specified in this file.
# Each line of this file contains a benchmark name, and class.
# The name is one of "sp-mz", "bt-mz", and "lu-mz".
# The class is one of "S", "W", and "A" through "F".
# No blank lines.
# The following example builds sample sizes of all benchmarks.
bt-mz     W
bt-mz     A
bt-mz     B
bt-mz     C
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd ../
$ pwd
  ../ESD2024/sdbase/NPB3.4.2-MZ/NPB3.4-MZ-MPI
$ make suite --> compila o benchmark BT-MZ
$ cd bin
$ ls -A1
bt-mz.A.x
bt-mz.B.x
bt-mz.C.x
bt-mz.W.x
BULL_srun_openmpi.sh
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p cpu_dev -N1 -n1 ./bt-mz.W.x  --> submete o job com srun

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas

$ salloc -p cpu_dev -N1 -n1  mpirun ./bt-mz.W.x  --> submete o job com salloc
salloc: Granted job allocation 105518

 Class           =                        W
 Size            =              64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.82
 Total processes =                        1
 Total threads   =                        1
 Mop/s total     =                  2464.61
 Mop/s/thread    =                  2464.61
salloc: Relinquishing job allocation 105518
salloc: Job allocation 105518 has been revoked.
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p sequana_cpu_dev -N1 -n1 ./bt-mz.W.x  --> submete o job com srun

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas

$ salloc -p sequana_cpu_dev -N1 -n1  mpirun ./bt-mz.W.x  --> submete o job com salloc
salloc: Granted job allocation 105518

 Class           =                        W
 Size            =              64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.82
 Total processes =                        1
 Total threads   =                        1
 Mop/s total     =                  2464.61
 Mop/s/thread    =                  2464.61
salloc: Relinquishing job allocation 105518
salloc: Job allocation 105518 has been revoked.
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ salloc -p cpu_dev -N1 -n2  mpirun -n 1 ./bt-mz.W.x --> submete o job com salloc
salloc: Granted job allocation 105519
Class           =                        W
 Size           =              64x   64x  8
 Iterations     =                      200
 Time in seconds =                    2.96
 Total processes =                       2
 Total threads   =                       2
 Mop/s total     =                 4851.15
 Mop/s/thread    =                 2425.57

salloc: Relinquishing job allocation 105519
salloc: Job allocation 105519 has been revoked.
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p cpu_dev -N1 -n2      ./bt-mz.W.x      --> submete o job com srun

Class           =                        W
 Size           =              64x   64x  8
 Iterations     =                      200
 Time in seconds =                    2.96
 Total processes =                       2
 Total threads   =                       2
 Mop/s total     =                 4851.15
 Mop/s/thread    =                 2425.57
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p cpu_dev -N1 -n1 -c2 ./bt-mz.W.x --> para execução multi-thread

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas
-c2 (--cpus-per-task=2): define o número de cpus por tarefa (MPI rank)

BT-MZ Benchmark Completed.
 Class           =                        W
 Size            =             64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.85  --> desempenho aquém do esperado
 Total processes =                        1
 Total threads   =                        1
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p cpu_dev -N1 -n1 -c2 ./bt-mz.W.x --> para execução multi-thread

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas
-c2 (--cpus-per-task=2): define o número de cpus por tarefa (MPI rank)

BT-MZ Benchmark Completed.
 Class           =                        W
 Size            =             64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.85  --> definir var. de amb. OMP_NUM_THREADS
 Total processes =                        1
 Total threads   =                        1

$ export OMP_NUM_THREADS=2 --> (igual a --cpus-per-task)
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p cpu_dev -N1 -n1 -c2 ./bt-mz.W.x --> para execução multi-thread

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas
-c2 (--cpus-per-task=2): define o número de cpus por tarefa (MPI rank)

BT-MZ Benchmark Completed.
 Class           =                        W
 Size            =             64x   64x  8
 Iterations      =                      200
 Time in seconds =                     3.03 --> redução de tempo
 Total processes =                        1
 Total threads   =                        2
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

SLURM: comandos básicos

sbatch

- parâmetros na linha de comando
- parâmetros no script

principais opções de configuração:
--time (-t)
--nodes (-N)
--ntasks (-n)
--ntasks-per-node
--cpus-per-task (-c)
--partition (-p)

Exemplo: NAS Parallel Benchmark (BT-MZ)

BULL_srun_openmpi.sh

```
#!/bin/bash

#SBATCH --nodes=1                       # here the number of nodes
#SBATCH --ntasks=1                      # here total number of mpi tasks
#SBATCH --cpus-per-task=1               # number of cores per node
#SBATCH -p cpu_dev                      # target partition
#SBATCH -J NPB_BT-MZ                    # job name
#SBATCH --time=00:05:00                 # time limit
#SBATCH --exclusive                     # to have exclusive use of your nodes

echo "Cluster configuration:"
echo "==="
echo "Partition: " $SLURM_JOB_PARTITION
echo "Number of nodes: " $SLURM_NNODES
echo "Number of MPI processes: " $SLURM_NTASKS " (" $SLURM_NNODES " nodes)"
echo "Number of MPI processes per node: " $SLURM_NTASKS_PER_NODE
echo "Number of threads per MPI process: " $SLURM_CPUS_PER_TASK
echo "NPB Benchmark: " $1
echo "Bechmark class problem: " $2
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

BULL_srun_openmpi.sh (cont.)

```
###################################
#           COMPILER              #
###################################
module load openmpi/gnu/4.1.2+cuda-11.2

DIR=$PWD

bench=${1}
class=${2}
execfile="${bench}.${class}.x"
BIN=$DIR/${execfile}

export OMPI_MCA_opal_warn_on_missing_libcuda=0
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

cd $DIR

srun -n $SLURM_NTASKS $BIN

dirdest="${bench}_${class}_MPI-${SLURM_NTASKS}_OMP-${SLURM_CPUS_PER_TASK}_JOBID-${SLURM_JOBID}"
mkdir $dirdest
cp slurm-${SLURM_JOBID}.out $dirdest/
```

Variáveis de ambiente do SLURM

Alguns exemplos:

SLURM_JOB_PARTITION
SLURM_NNODES
SLURM_NTASKS
SLURM_NTASKS_PER_NODE
SLURM_CPUS_PER_TASK
SLURM_JOBID
SLURM_JOB_NODELIST
SLURM_SUBMIT_DIR

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ sbatch BULL_srun_openmpi.sh bt-mz W
Submitted batch job 105539
$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            105539 sequana_c NPB_BT-M professo  R       0:05      1 sdumont5000

$ ls bt-mz_W_MPI-1_OMP-1_JOBID-105539/
slurm-105539.out     --> arquivo gerado pelo SLURM com a saída da aplicação

$ sbatch -N2 -n2 -c1 ./BULL_srun_openmpi.sh bt-mz W
Os parâmetros por linha de comando têm precedência sobre os definidos no script

$ ls bt-mz_W_MPI-2_OMP-1_JOBID-105540
slurm-105540.out
```

`scontrol`: alterando parâmetro do job

```
--dependency (-d): adia o início do job até que a dependência especificada seja satisfeita

$ sbatch BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105558
$ sbatch -d afterany:105558 BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105559
$ sbatch -d afterany:105559 BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105560
$ sbatch -d afterany:105560 BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105561
$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            105559 treinamen NPB_BT-M professo PD       0:00      1 (Dependency)
            105560 treinamen NPB_BT-M professo PD       0:00      1 (Dependency)
            105561 treinamen NPB_BT-M professo PD       0:00      1 (Dependency)
            105558 treinamen NPB_BT-M professo  R       0:26      1 sdumont5000
```

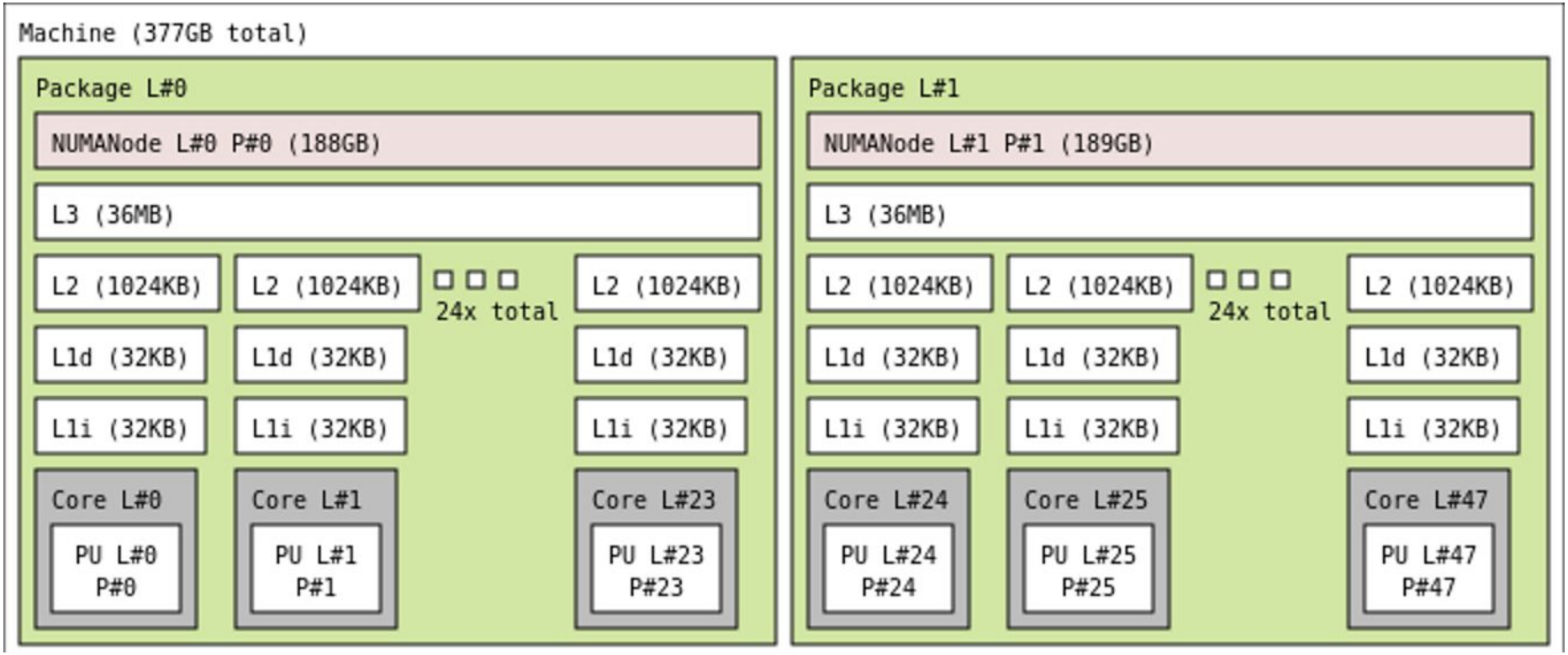
`scontrol:` alterando parâmetro do job

```
$ scontrol show jobid 105561
JobId=105561 JobName=NPB_BT-MZ
   UserId=professor(63001) GroupId=treinamento(61052)
   Priority=1791 Nice=0 Account=treinamento QOS=normal
   JobState=PENDING Reason=Dependency Dependency=afterany:105560
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=00:05:00 TimeMin=N/A
   SubmitTime=2017-07-30T17:42:04 EligibleTime=Unknown
   StartTime=Unknown EndTime=Unknown
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=cpu AllocNode:Sid=sdumont14:19952
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=(null)
   NumNodes=1-1 NumCPUs=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:1
   Socks/Node=* NtasksPerN:B:S:C=1:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   Shared=0 Contiguous=0 Licenses=(null) Network=(null)
```

`scontrol:` alterando parâmetro do job

```
$ scontrol update JobId=105561 Partition=cpu_dev
$ scontrol show jobid 105561
JobId=105561 JobName=NPB_BT-MZ
   UserId=professor(63001) GroupId=treinamento(61052)
   Priority=1791 Nice=0 Account=treinamento QOS=normal
   JobState=PENDING Reason=Dependency Dependency=afterany:105560
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=00:05:00 TimeMin=N/A
   SubmitTime=2017-07-30T17:42:04 EligibleTime=Unknown
   StartTime=Unknown EndTime=Unknown
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=cpu_dev AllocNode:Sid=sdumont14:19952
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=(null)
   NumNodes=1-1 NumCPUs=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:1
   Socks/Node=* NtasksPerN:B:S:C=1:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   Shared=0 Contiguous=0 Licenses=(null) Network=(null)
```

SEQUANA


Machine (377GB total)
Package L#0
NUMANode L#0 P#0 (188GB)
L3 (36MB)
L2 (1024KB)
L2 (1024KB)
24x total
L2 (1024KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core L#0
PU L#0 P#0
Core L#1
PU L#1 P#1
Core L#23
PU L#23 P#23
Package L#1
NUMANode L#1 P#1 (189GB)
L3 (36MB)
L2 (1024KB)
L2 (1024KB)
24x total
L2 (1024KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1i (32KB)
L1i (32KB)
L1i (32KB)
Core L#24
PU L#24 P#24
Core L#25
PU L#25 P#25
Core L#47
PU L#47 P#47

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
ESD2024/
├── README.md
├── sdbase
│   ├── env_openmpi
│   └── NPB3.4.2-MZ
└── sequana
    ├── env_openmpi
    └── NPB3.4.2-MZ
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
ESD2024/
├── README.md
├── sdbase
│   ├── env_openmpi
│   └── NPB3.4.2-MZ
└── sequana
    ├── env_openmpi
    └── NPB3.4.2-MZ
        ├── NPB3.4-MZ-MPI
        │   ├── bin
        │   │   └── BULL_srun_openmpi.sh
        │   ├── BT-MZ
        │   ├── config
        │   │   ├── make.def
        │   │   ├── suite.def
        │   ├── Makefile
        └── README
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd $SCRATCH -> vai para o diretório de sua conta na partição do lustre (/scratch)
TODA SUBMISSÃO DE JOB DEVE SER FEITA COM EXECUTÁVEIS INSTALADOS NA PARTIÇÃO /scratch
$ pwd $SCRATCH -> verifica o caminho deste diretório

$ ssh sdumont18
$ module load sequana/current
Loading SEQUANA Software environment
$ module load git/2.23_sequana
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd ESD2024/sequana/
$ ls -A1
env_openmpi
NPB3.4.2-MZ

$ cat env_openmpi
module load openmpi/gnu/4.1.2+cuda-11.2_sequana

$ source env_openmpi
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd NPB3.4.2-MZ/NPB3.4-MZ-MPI/config/
$ ls -A1
  make.def
  make.def.template
  NAS.samples/
  suite.def
  suite.def.template
$ cat make.def
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cat suite.def

# config/suite.def
# This file is used to build several benchmarks with a single command.
# Typing "make suite" in the main directory will build all the benchmarks
# specified in this file.
# Each line of this file contains a benchmark name, and class.
# The name is one of "sp-mz", "bt-mz", and "lu-mz".
# The class is one of "S", "W", and "A" through "F".
# No blank lines.
# The following example builds sample sizes of all benchmarks.
bt-mz     W
bt-mz     A
bt-mz     B
bt-mz     C
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ cd ../
$ pwd
  ../ESD2024/sequana/NPB3.4.2-MZ/NPB3.4-MZ-MPI
$ make suite --> compila o benchmark BT-MZ
$ cd bin
$ ls -A1
bt-mz.A.x
bt-mz.B.x
bt-mz.C.x
bt-mz.W.x
BULL_srun_openmpi.sh
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p sequana_cpu_dev -N1 -n1 ./bt-mz.W.x  --> submete o job com srun

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas

$ salloc -p sequana_cpu_dev -N1 -n1  mpirun ./bt-mz.W.x  --> submete o job com salloc
salloc: Granted job allocation 105518

 Class           =                        W
 Size            =              64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.82
 Total processes =                        1
 Total threads   =                        1
 Mop/s total     =                  2464.61
 Mop/s/thread    =                  2464.61
salloc: Relinquishing job allocation 105518
salloc: Job allocation 105518 has been revoked.
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ salloc -p sequana_cpu_dev -N1 -n2  mpirun -n 1 ./bt-mz.W.x --> submete o job com salloc
salloc: Granted job allocation 105519
Class            =                        W
 Size            =              64x   64x  8
 Iterations      =                      200
 Time in seconds =                     2.96
 Total processes =                        2
 Total threads   =                        2
 Mop/s total     =                  4851.15
 Mop/s/thread    =                  2425.57

salloc: Relinquishing job allocation 105519
salloc: Job allocation 105519 has been revoked.
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p sequana_cpu_dev -N1 -n2      ./bt-mz.W.x      --> submete o job com srun

Class           =                        W
 Size            =              64x   64x  8
 Iterations      =                      200
 Time in seconds =                     2.96
 Total processes =                        2
 Total threads   =                        2
 Mop/s total     =                  4851.15
 Mop/s/thread    =                  2425.57
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p sequana_cpu_dev -N1 -n1 -c2 ./bt-mz.W.x --> para execução multi-thread

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas
-c2 (--cpus-per-task=2): define o número de cpus por tarefa (MPI rank)

BT-MZ Benchmark Completed.
 Class           =                        W
 Size            =             64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.85  --> desempenho aquém do esperado
 Total processes =                        1
 Total threads   =                        1
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p sequana_cpu_dev -N1 -n1 -c2 ./bt-mz.W.x --> para execução multi-thread

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas
-c2 (--cpus-per-task=2): define o número de cpus por tarefa (MPI rank)

BT-MZ Benchmark Completed.
 Class           =                        W
 Size            =             64x   64x  8
 Iterations      =                      200
 Time in seconds =                     5.85  --> definir var. de amb. OMP_NUM_THREADS
 Total processes =                        1
 Total threads   =                        1

$ export OMP_NUM_THREADS=2 --> (igual a --cpus-per-task)
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ srun -p sequana_cpu_dev -N1 -n1 -c2 ./bt-mz.W.x --> para execução multi-thread

-N1 (--nodes=1): define o número de nós a serem alocados
-n1 (--ntasks=1): define o número total de tarefas (MPI ranks) a serem executadas
-c2 (--cpus-per-task=2): define o número de cpus por tarefa (MPI rank)

BT-MZ Benchmark Completed.
 Class           =                        W
 Size            =             64x   64x  8
 Iterations      =                      200
 Time in seconds =                     3.03 --> redução de tempo
 Total processes =                        1
 Total threads   =                        2
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

SLURM: comandos básicos

sbatch

- parâmetros na linha de comando
- parâmetros no script

principais opções de configuração:
--time (-t)
--nodes (-N)
--ntasks (-n)
--ntasks-per-node
--cpus-per-task (-c)
--partition (-p)

Exemplo: NAS Parallel Benchmark (BT-MZ)

BULL_srun_openmpi.sh

```
#!/bin/bash

#SBATCH --nodes=1                       # here the number of nodes
#SBATCH --ntasks=1                      # here total number of mpi tasks
#SBATCH --cpus-per-task=1               # number of cores per node
#SBATCH -p sequana_cpu_dev                      # target partition
#SBATCH -J NPB_BT-MZ                    # job name
#SBATCH --time=00:05:00                 # time limit
#SBATCH --exclusive                     # to have exclusive use of your nodes

echo "Cluster configuration:"
echo "==="
echo "Partition: " $SLURM_JOB_PARTITION
echo "Number of nodes: " $SLURM_NNODES
echo "Number of MPI processes: " $SLURM_NTASKS " (" $SLURM_NNODES " nodes)"
echo "Number of MPI processes per node: " $SLURM_NTASKS_PER_NODE
echo "Number of threads per MPI process: " $SLURM_CPUS_PER_TASK
echo "NPB Benchmark: " $1
echo "Bechmark class problem: " $2
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

BULL_srun_openmpi.sh (cont.)

```
###################################
#           COMPILER              #
###################################
module load sequana/current
module load openmpi/gnu/4.1.2+cuda-11.2_sequana

DIR=$PWD

bench=${1}
class=${2}
execfile="${bench}.${class}.x"
BIN=$DIR/${execfile}

export OMPI_MCA_opal_warn_on_missing_libcuda=0
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

cd $DIR

srun -n $SLURM_NTASKS $BIN

dirdest="${bench}_${class}_MPI-${SLURM_NTASKS}_OMP-${SLURM_CPUS_PER_TASK}_JOBID-${SLURM_JOBID}"
mkdir $dirdest
cp slurm-${SLURM_JOBID}.out $dirdest/
```

Variáveis de ambiente do SLURM

Alguns exemplos:

SLURM_JOB_PARTITION
SLURM_NNODES
SLURM_NTASKS
SLURM_NTASKS_PER_NODE
SLURM_CPUS_PER_TASK
SLURM_JOBID
SLURM_JOB_NODELIST
SLURM_SUBMIT_DIR

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
$ sbatch BULL_srun_openmpi.sh bt-mz W
Submitted batch job 105539
$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            105539 sequana_c NPB_BT-M professo  R       0:05      1 sdumont5000

$ ls bt-mz_W_MPI-1_OMP-1_JOBID-105539/
slurm-105539.out     --> arquivo gerado pelo SLURM com a saída da aplicação

$ sbatch -N2 -n2 -c1 ./BULL_srun_openmpi.sh bt-mz W
Os parâmetros por linha de comando têm precedência sobre os definidos no script

$ ls bt-mz_W_MPI-2_OMP-1_JOBID-105540
slurm-105540.out
```

`scontrol`: alterando parâmetro do job

```
--dependency (-d): adia o início do job até que a dependência especificada seja satisfeita

$ sbatch BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105558
$ sbatch -d afterany:105558 BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105559
$ sbatch -d afterany:105559 BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105560
$ sbatch -d afterany:105560 BULL_srun_openmpi.sh bt-mz A
Submitted batch job 105561
$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            105559 treinamen NPB_BT-M professo PD       0:00      1 (Dependency)
            105560 treinamen NPB_BT-M professo PD       0:00      1 (Dependency)
            105561 treinamen NPB_BT-M professo PD       0:00      1 (Dependency)
            105558 treinamen NPB_BT-M professo  R       0:26      1 sdumont5000
```

`scontrol:` alterando parâmetro do job

```
$ scontrol show jobid 105561
JobId=105561 JobName=NPB_BT-MZ
   UserId=professor(63001) GroupId=treinamento(61052)
   Priority=1791 Nice=0 Account=treinamento QOS=normal
   JobState=PENDING Reason=Dependency Dependency=afterany:105560
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=00:05:00 TimeMin=N/A
   SubmitTime=2017-07-30T17:42:04 EligibleTime=Unknown
   StartTime=Unknown EndTime=Unknown
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=sequana_cpu_dev AllocNode:Sid=sdumont14:19952
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=(null)
   NumNodes=1-1 NumCPUs=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:1
   Socks/Node=* NtasksPerN:B:S:C=1:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   Shared=0 Contiguous=0 Licenses=(null) Network=(null)
```

`scontrol`: alterando parâmetro do job

```
$ scontrol update JobId=105561 Partition=sequana_cpu_dev
$ scontrol show jobid 105561
JobId=105561 JobName=NPB_BT-MZ
   UserId=professor(63001) GroupId=treinamento(61052)
   Priority=1791 Nice=0 Account=treinamento QOS=normal
   JobState=PENDING Reason=Dependency Dependency=afterany:105560
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:00 TimeLimit=00:05:00 TimeMin=N/A
   SubmitTime=2017-07-30T17:42:04 EligibleTime=Unknown
   StartTime=Unknown EndTime=Unknown
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=sequana_cpu_dev AllocNode:Sid=sdumont14:19952
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=(null)
   NumNodes=1-1 NumCPUs=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:1
   Socks/Node=* NtasksPerN:B:S:C=1:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   Shared=0 Contiguous=0 Licenses=(null) Network=(null)
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

`sacct:` mostra informação de jobs já submetidos

```
$ sacct
158796            mpirun treinamen+ treinamen+          1     FAILED      1:0
158798            mpirun treinamen+ treinamen+          2  COMPLETED      0:0
158798.0           orted             treinamen+          2  COMPLETED      0:0
158802       bt-mz.A.2 treinamen+ treinamen+          2 CANCELLED+      0:0
158803       bt-mz.A.2 treinamen+ treinamen+          2  COMPLETED      0:0
158804       bt-mz.W.2 treinamen+ treinamen+          2  COMPLETED      0:0
158809            mpirun treinamen+ treinamen+          2  COMPLETED      0:0
158809.0           orted             treinamen+          2  COMPLETED      0:0
158810       bt-mz.W.2 treinamen+ treinamen+          2  COMPLETED      0:0
158811       bt-mz.W.2 treinamen+ treinamen+          2  COMPLETED      0:0
```

`sacct:` mostra informação de jobs já submetidos

```
$ sacct -j 158811
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
158811        bt-mz.W.2 treinamen+ treinamen+          2  COMPLETED      0:0
```

`sacct:` mostra informação de jobs já submetidos

```
$ sacct -e
AllocCPUS           AllocGRES           Account             AssocID
AveCPU              AveCPUFreq          AveDiskRead         AveDiskWrite
AvePages            AveRSS              AveVMSize           BlockID
Cluster             Comment             ConsumedEnergy      ConsumedEnergyRaw
CPUTime             CPUTimeRAW          DerivedExitCode     Elapsed
Eligible            End                 ExitCode            GID
Group               JobID               JobIDRaw            JobName
Layout              MaxDiskRead         MaxDiskReadNode     MaxDiskReadTask
MaxDiskWrite        MaxDiskWriteNode    MaxDiskWriteTask    MaxPages
MaxPagesNode        MaxPagesTask        MaxRSS              MaxRSSNode
MaxRSSTask          MaxVMSize           MaxVMSizeNode       MaxVMSizeTask
MinCPU              MinCPUNode          MinCPUTask          NCPUS
NNodes              NodeList            NTasks              Priority
Partition           QOS                 QOSRAW              ReqCPUFreq
ReqCPUS             ReqGRES             ReqMem              Reservation
ReservationId       Reserved            ResvCPU             ResvCPURAW
Start               State               Submit              Suspended
SystemCPU           Timelimit           TotalCPU            UID
User                UserCPU             WCKey               WCKeyID
```

`sacct:` mostra informação de jobs já submetidos

```
$ sacct -j 158811 --format=JobID,JobName,Elapsed,NodeList
       JobID    JobName    Elapsed        NodeList
------------ ---------- ---------- ---------------
158811        bt-mz.W.2   00:00:04     sdumont5000
```

SLURM: comandos básicos

- **sacctmgr:** lista acesso às filas
- **sinfo:** visualiza informação sobre os nós e partições do SLURM
- **squeue:** visualiza informação sobre o status dos jobs e escalonamento
- **srun:** alocação de recursos e distribuição de tarefas
- **scontrol:** ferramenta para visualizar e/ou modificar o estado de um job
- **salloc:** obtém uma alocação para o job
- **sbatch:** submete um script para alocação em uma partição do SLURM
- **scancel:** cancela um job
- **sacct:** mostra informação de jobs já submetidos
- **sreport:** mostra os recursos consumidos

`sreport:` mostra os recursos consumidos

Exibe a quantidade de horas utilizada pelos Projetos durante um período, delimitado pelos parâmetros *start* e *end*.

```
$ sreport -t hours cluster AccountUtilizationByUser start=AAAA-MM-DD end=AAAA-MM-DD \
Accounts=PROJETO Tree
```

```
$ sreport -t hours cluster AccountUtilizationByUser start=2023-05-01 end=2023-06-01 Accounts=lncc Tree
--------------------------------------------------------------------------------
Cluster/Account/User Utilization 2023-05-01T00:00:00 - 2023-05-31T23:59:59 (2678400 secs)
Usage reported in CPU Hours
--------------------------------------------------------------------------------
  Cluster              Account     Login     Proper Name     Used    Energy
--------- -------------------- --------- --------------- -------- ---------
  sdumont lncc                                            15643     45565 <-- Total do Projeto
  sdumont  prjssisd                                        1521     38303 <-- Total do Projeto vinculado
  sdumont   prjssisd           andrercsd Andre Ramos Ca+      658         0
  sdumont   prjssisd           brunoafsd Bruno Alvez Fa+      863     38303
  sdumont  prjstasd                                        9508      3822 <-- Total do Projeto vinculado
  sdumont   prjstasd           caio.san+ Caio Graco Per+      118         0
  sdumont   prjstasd           caio.san+ Caio Graco Per+     9390      3822
```

`sreport:` mostra os recursos consumidos

Exibe a quantidade de horas utilizada pelo projeto durante um período, delimitado pelos parâmetros *start* e *end*. Útil para o coordenador do Projeto

```
$ sreport -t hours cluster UserUtilizationByAccount start=AAAA-MM-DD end=AAAA-MM-DD \
Accounts=PROJETO
```

```
$ sreport -t hours cluster UserUtilizationByAccount start=2023-05-01 end=2023-06-01 Accounts=prjssisd
--------------------------------------------------------------------------------
Cluster/User/Account Utilization 2023-05-01T00:00:00 - 2023-05-31T23:59:59 (2678400 secs)
Usage reported in CPU Hours
--------------------------------------------------------------------------------
  Cluster     Login     Proper Name         Account     Used   Energy
--------- --------- --------------- --------------- -------- --------
  sdumont brunoafsd Bruno Alves Fa+        prjssisd     8509        5
  sdumont andrercsd Andre Ramos Ca+        prjssisd     8083        0
  sdumont   jpassos Jeferson Passos        prjssisd     7660        0
  sdumont carlos.a+ Carlos Daniel +        prjssisd     5964       51
  sdumont bruno.fa+ Bruno Fagundes+        prjssisd     3404        0
  sdumont fabio.so+ Fabio Moreira +        prjssisd      149        0
  sdumont regio.pi+     Regio Pires        prjssisd       33      121
```

Uso das filas com GPU

SDBASE

```
nvidia
nvidia_dev
nvidia_long
nvidia_scal
```

SEQUANA

```
sequana_gpu
sequana_gpu_dev
sequana_gpu_long
sequana_gpu_shared
```

Uso das filas com GPU - SDBASE

- Carregar o ambiente com o NVIDIA CUDA Toolkit v11.2
  `$ module load cuda/11.2`

- Verificar a configuração de GPU do nó
  `$ srun -p nvidia_dev nvidia-smi`

```
srun: job 10857658 queued and waiting for resources
srun: job 10857658 has been allocated resources
Sat Jan 20 13:57:26 2024
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.82.01    Driver Version: 470.82.01    CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla K40t          On   | 00000000:01:00.0 Off |                    0 |
| N/A   41C    P8    21W / 235W |      0MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla K40t          On   | 00000000:81:00.0 Off |                    0 |
| N/A   41C    P8    20W / 235W |      0MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Uso das filas com GPU - SDBASE

- Copiar os exemplos do NVIDIA CUDA Toolkit v11.2

```
$ cd $SCRATCH
$ cuda-install-samples-11.2.sh $SCRATCH
Copying samples to
/scratch/cenapadrjsd/rpsouto/NVIDIA_CUDA-11.2_Samples now...
Finished copying samples.
```

- Verificar exemplo de multiplicação de matrizes

```
$ cd $SCRATCH/NVIDIA_CUDA-11.2_Samples/0_Simple/matrixMul
$ ls
Makefile  matrixMul.cu  NsightEclipse.xml  readme.txt
```

Uso das filas com GPU - SDBASE

- Compilar exemplo de multiplicação de matrizes

```
$ make
/usr/local/cuda-11.2/bin/nvcc -ccbin g++ -I../../common/inc  -m64    --threads 0 -gencode
arch=compute_35,code=sm_35 -gencode arch=compute_37,code=sm_37 -gencode arch=compute_50,code=sm_50
-gencode arch=compute_52,code=sm_52 -gencode arch=compute_60,code=sm_60 -gencode
arch=compute_61,code=sm_61 -gencode arch=compute_70,code=sm_70 -gencode arch=compute_75,code=sm_75
-gencode arch=compute_80,code=sm_80 -gencode arch=compute_86,code=sm_86 -gencode
arch=compute_86,code=compute_86 -o matrixMul.o -c matrixMul.cu
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50'
architectures are deprecated, and may be removed in a future release (Use
-Wno-deprecated-gpu-targets to suppress warning).
/usr/local/cuda-11.2/bin/nvcc -ccbin g++   -m64      -gencode arch=compute_35,code=sm_35 -gencode
arch=compute_37,code=sm_37 -gencode arch=compute_50,code=sm_50 -gencode arch=compute_52,code=sm_52
-gencode arch=compute_60,code=sm_60 -gencode arch=compute_61,code=sm_61 -gencode
arch=compute_70,code=sm_70 -gencode arch=compute_75,code=sm_75 -gencode arch=compute_80,code=sm_80
-gencode arch=compute_86,code=sm_86 -gencode arch=compute_86,code=compute_86 -o matrixMul
matrixMul.o
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50'
architectures are deprecated, and may be removed in a future release (Use
-Wno-deprecated-gpu-targets to suppress warning).
mkdir -p ../../bin/x86_64/linux/release
cp matrixMul ../../bin/x86_64/linux/release

$ ls
Makefile  matrixMul  matrixMul.cu  matrixMul.o  NsightEclipse.xml  readme.txt
```

Uso das filas com GPU - SDBASE

- Executar exemplo de multiplicação de matrizes

```
$ srun -p nvidia_dev ./matrixMul -wA=1024 -hA=256 -wB=256 -hB=1024
srun: job 10857663 queued and waiting for resources
srun: job 10857663 has been allocated resources
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Kepler" with compute capability 3.5

MatrixA(1024,256), MatrixB(256,1024)
Computing result using CUDA Kernel...
done
Performance= 310.19 GFlop/s, Time= 0.433 msec, Size= 134217728 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performancemeasurements. Results may
vary when GPU Boost is enabled.
```

Uso das filas com GPU - SDBASE

- Executar exemplo de multiplicação de matrizes

```
$ srun -p nvidia_dev ./matrixMul -wA=2048 -hA=256 -wB=256 -hB=2048
srun: job 10857664 queued and waiting for resources
srun: job 10857664 has been allocated resources
[Matrix Multiply Using CUDA] - Starting…
GPU Device 0: "Kepler" with compute capability 3.5

MatrixA(2048,256), MatrixB(256,2048)
Computing result using CUDA Kernel…
done
Performance= 311.69 GFlop/s, Time= 0.861 msec, Size= 268435456 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performancemeasurements. Results may
vary when GPU Boost is enabled.
```

Uso das filas com GPU - SDBASE

- Executar exemplo de multiplicação de matrizes

```
$ srun -p nvidia_dev ./matrixMul -wA=4096 -hA=256 -wB=256 -hB=4096
srun: job 10857665 queued and waiting for resources
srun: job 10857665 has been allocated resources
[Matrix Multiply Using CUDA] - Starting…
GPU Device 0: "Kepler" with compute capability 3.5

MatrixA(4096,256), MatrixB(256,4096)
Computing result using CUDA Kernel…
done
Performance= 312.00 GFlop/s, Time= 1.721 msec, Size= 536870912 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performancemeasurements. Results may
vary when GPU Boost is enabled.
```

Uso das filas com GPU - SDBASE

- Executar exemplo de multiplicação de matrizes

```
$ srun -p nvidia_dev ./matrixMul -wA=8192 -hA=256 -wB=256 -hB=8192
srun: job 10857667 queued and waiting for resources
srun: job 10857667 has been allocated resources
[Matrix Multiply Using CUDA] - Starting…
GPU Device 0: "Kepler" with compute capability 3.5

MatrixA(8192,256), MatrixB(256,8192)
Computing result using CUDA Kernel…
done
Performance= 306.13 GFlop/s, Time= 3.507 msec, Size= 1073741824 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performancemeasurements. Results may
vary when GPU Boost is enabled.
```