Introdução ao ambiente SDUMONT/SLURM

Escola Santos Dumont
27 de janeiro 2025
LNCC
evento remoto online

Laboratório Nacional de Computação Científica

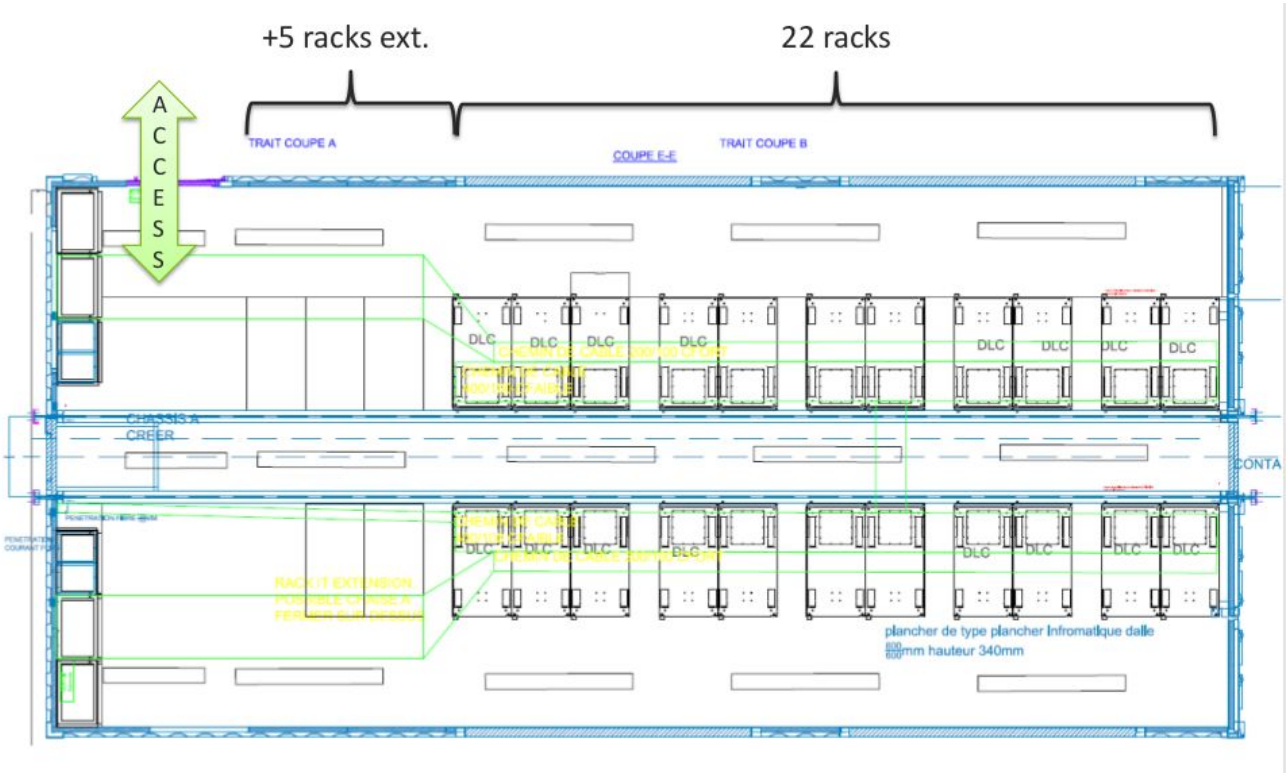
A máquina

Mobull - solução para datacenter baseada em containers.

Plug & Boot

2 containers com 22 racks 42U

A máquina


+5 racks ext.
22 racks
ACCESS
TRAIT COUPE A
COUPE E-E
TRAIT COUPE B
DLC
CHASSIS A CREER
plancher de type plancher Infromatique dalle
mm hauteur 340mm

Arquitetura

Cluster de propósito geral

SDumont Base (2015 - **descontinuado em 09/2025**). 3 tipos de nodes:

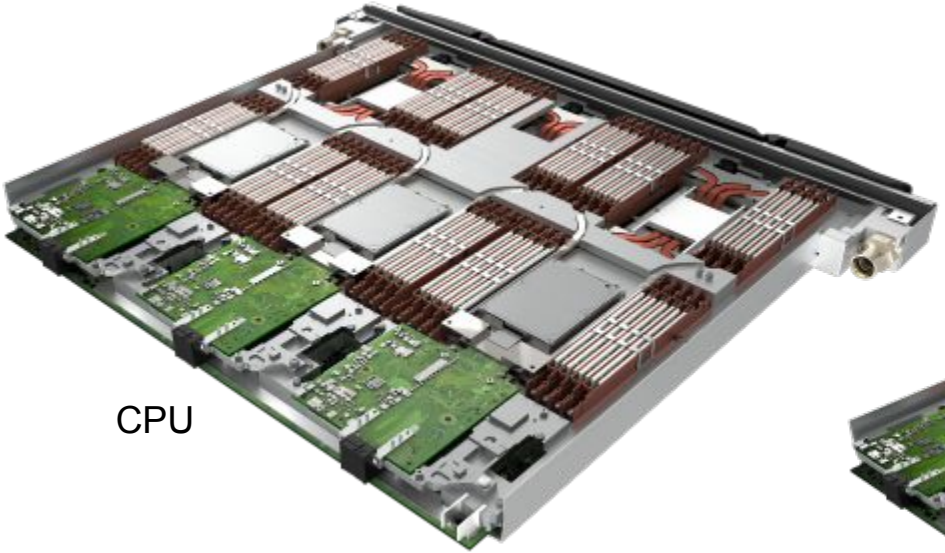
- thin nodes
  - 504 nós computacionais CPU
- hybrid nodes
  - 198 nós computacionais e 396 nVidia K40
  - 54 nós computacionais e 108 Intel Phi 7120P
- fat-node
  - 240 cores e 7TB de memória RAM

Expansão SDumont - 2018/2019 - Atual

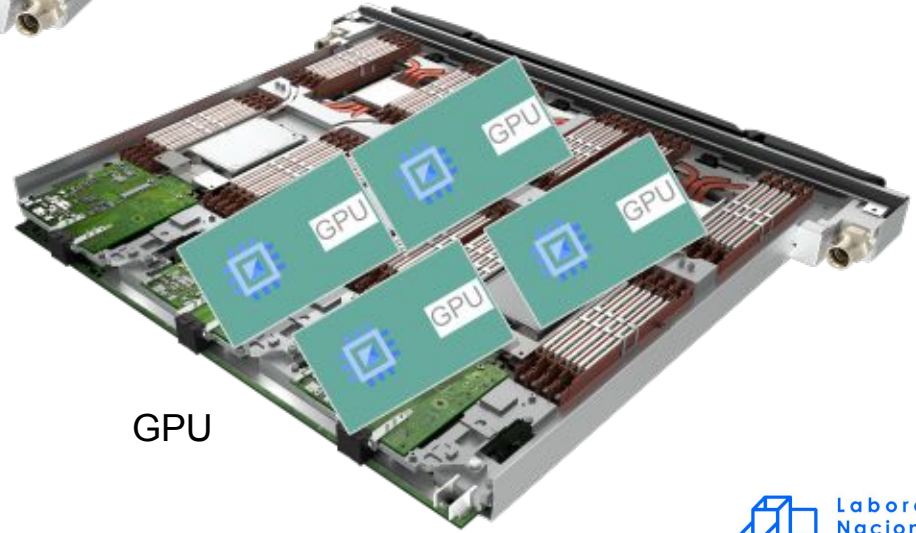
- 1 Célula Sequana X1000 CPU
  - 82 Blades X1120 - 384 GB
  - 12 Blades X1120 - 768 GB
  - **282 nós computacionais**
  - 2x Intel CascadeLake Gold 6252 (24c)

- 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

Arquitetura

Bull Sequana - Machine Learning/Deep Learning

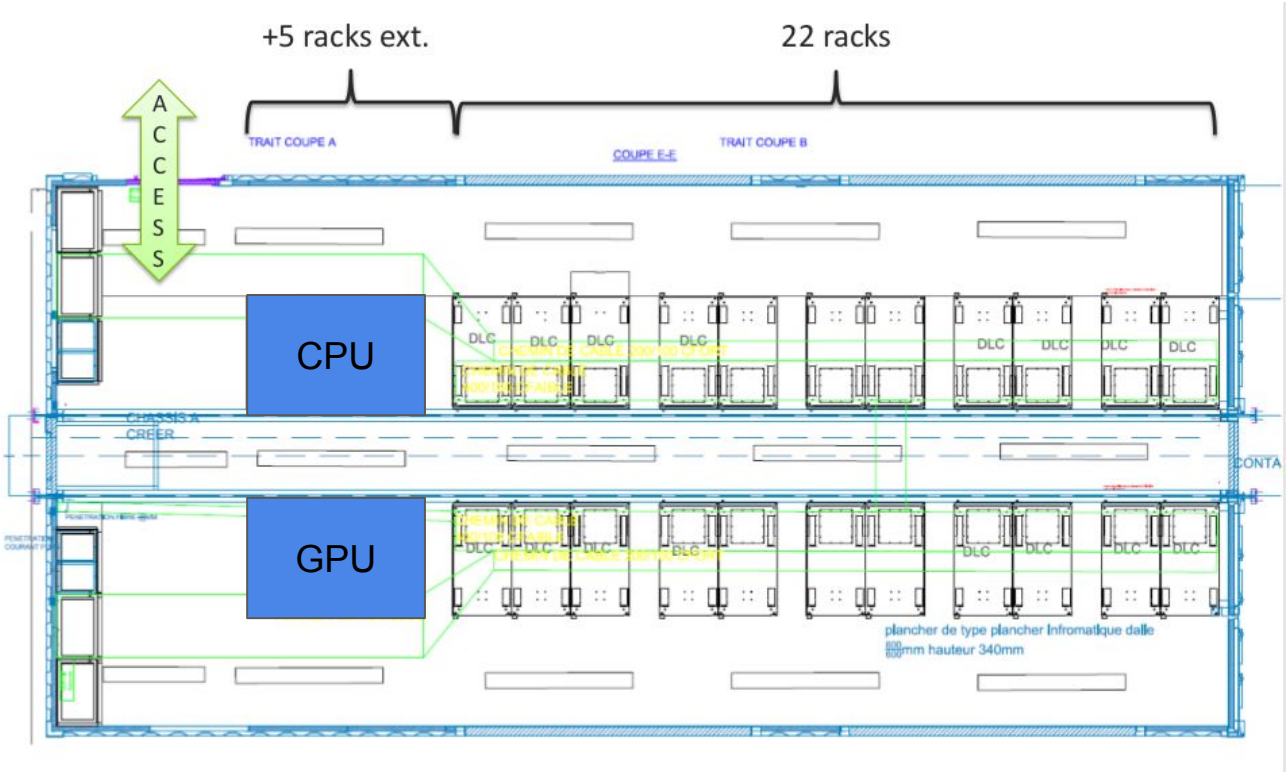
- 1 nó computacional
- Configuração
  - 2x Intel Skylake GOLD 6148, 2,4Ghz (20c)
  - 384 GB DDR4
  - 4x Infiniband EDR 100Gbps
  - 8x NVidia V100 com NVLink

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



Arquitetura

Armazenamento

- Lustre - CRAY/HPE ClusterStor 9000 - /scratch_old
  - Total 1,7 Petabytes
  - **Processo de descomissionamento**
- Lustre - CRAY/HPE ClusterStor L300 - /scratch
  - Total 1,1 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 - Base

- 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

Desempenho - Expansão

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 - Expansão

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	

Desempenho - SDumont II

Novembro 2025

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
89	**Santos Dumont** - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, Red Hat Enterprise Linux, EVIDEN Laboratório Nacional de Computação Científica Brazil	68,064	14.29	20.26	312

Filas

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
- Intel OneAPI
  - Versões: 2022
- PGI
  - Versão 2016.5 e 2019.10 (Community) - **Expirado**! Possível instalar novas versões, caso necessário.

Implementações MPI

- OpenMPI
  - Versões: 2.1.x, 4.0.x, 4.1.x e 5.0.x
- Intel MPI
  - Versões: 2016, 2017, 2018, 2019 e 2020
- Intel OneAPI
  - Versões: 2022
- MPICH2
  - Versões: 1.4.1 e 4.2.1

Slurm

Versão 23.11.1

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”.

SDumont II

- Nova versão do SDumont (BullSequana XH3000), adquirido em 2025.
- Em fase de testes e homologação.
- Previsão para entrar em operação em Fev/2025
- 60 nós computacionais de CPU:
  - 2 x AMD Genoa 9684X com 96 cores
- 62 nós computacionais de GPU:
  - 2 x Intel SHR M9468 HBM2 com 48 cores e 4 x Nvidia Hopper HGX H100
- 36 nós computacionais NVIDIA Grace Hopper:
  - 4 x NVIDIA GH200 Grace Hopper Superchip (CPU ARM + GPU)
- 4 nós computacionais ARM:
  - o 1 x NVIDIA Grace CPU Superchip.
- 36 nós computacionais APU AMD nodes:
  - 4 x AMD MI300A

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_dev       1        4      192         1    00:20:00
      xpto                sequana_cpu_dev       1        4      192         1    00:20:00
```

*¹Lista resumida: sacctmgr list user $USER -s format=partition%30
*²A variável de ambiente "**$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

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
sequana_cpu        up   infinite      174/6/7/187 sdumont[6068-6164,6192-6251,6255-6275,6279-6287]
sequana_cpu_dev    up      20:00       99/6/7/112 sdumont[6068-6084,6165-6169,6192-6251,6255-6275,6279-6287]
sequana_cpu_long   up   infinite      174/6/7/187 sdumont[6068-6164,6192-6251,6255-6275,6279-6287]
sequana_cpu_bigmem up   infinite        11/5/2/18 sdumont[6018-6035]
sequana_cpu_bigmem_long up   infinite        11/5/2/18 sdumont[6018-6035]
sequana_gpu        up   infinite       34/17/0/51 sdumont[8029-8045,8047-8050,8064-8083,8085-8091,8093-8095]
sequana_gpu_dev    up   infinite       34/27/0/61 sdumont[8029-8055,8060-8083,8085-8091,8093-8095]
sequana_gpu_long   up   infinite       34/17/0/51 sdumont[8029-8045,8047-8050,8064-8083,8085-8091,8093-8095]
sequana_all     drain   infinite     321/46/9/376 sdumont[6000-6287,8000-8095]
gdl                up   infinite          0/1/0/1 sdumont4000
```

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 sequana_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 sequana_cpu_dev     sleep    rpsouto   R        0:40      3 sdumont6089

$ scontrol --details show job 10756932

   NumNodes=1 NumCPUs=6 NumTasks=6 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   Nodes=sdumont6089 CPU_IDs=0-5 Mem=64000
```

srun

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

$ squeue -u $USER
      JOBID PARTITION           NAME     USER      ST       TIME  NODES NODELIST(REASON)
   10756971 sequana_cpu_dev     sleep    rpsouto   R        0:40      3 sdumont6089

$ scontrol --details show job 10756932

   NumNodes=1 NumCPUs=6 NumTasks=6 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   Nodes=sdumont6089 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=sequana_cpu_dev AllocNode:Sid=sdumont11:6575
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=sdumont6089
   BatchHost=sdumont6089
   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
```

`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 AllocNode:Sid=sdumont18:19952
   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=sdumont18:19952
   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

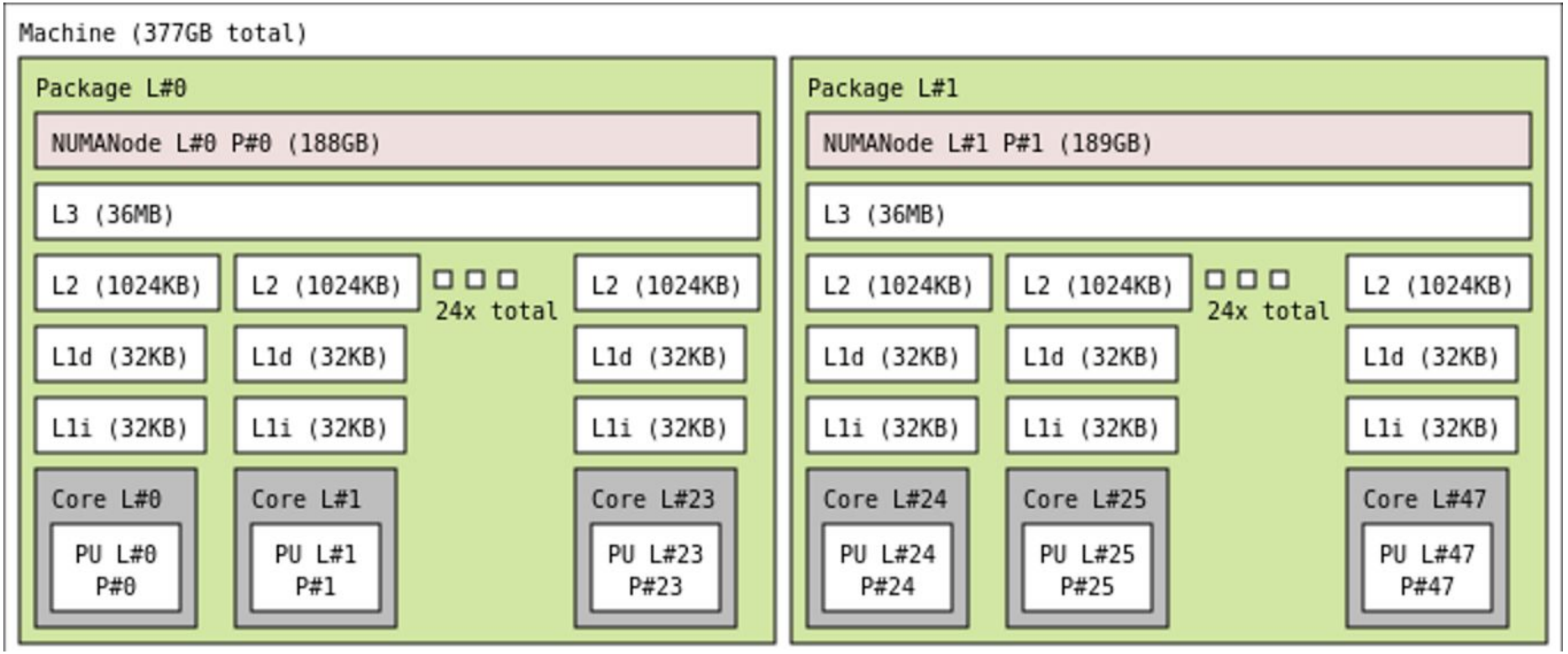
```
--dependency (-d): adia o início do job até que a dependência especificada seja satisfeita

$ 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 sdumont6000
```

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*)

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

`srun`: distribuição das tarefas

```
$ srun -p sequana_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 - sdumont6027, task  2  2 [2318]: mask 0x4 set
1: cpu-bind=MASK - sdumont6027, task  1  1 [2317]: mask 0x2 set
3: cpu-bind=MASK - sdumont6027, task  3  3 [2319]: mask 0x8 set
4: cpu-bind=MASK - sdumont6027, task  4  4 [2320]: mask 0x10 set
0: cpu-bind=MASK - sdumont6027, task  0  0 [2316]: mask 0x1 set
5: cpu-bind=MASK - sdumont6027, 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 sequana_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 - sdumont6027, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont6027, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont6028, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont6028, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont6028, 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 sequana_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 - sdumont6027, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont6027, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont6028, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont6028, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont6028, 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 sequana_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 - sdumont6027, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont6027, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont6028, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont6028, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont6028, 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 sequana_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 - sdumont6027, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont6027, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont6028, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont6028, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont6028, 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 sequana_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 - sdumont6027, task  0  0 [21551]: mask 0x1 set
1: cpu-bind=MASK - sdumont6027, task  1  1 [21552]: mask 0x2 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [21553]: mask 0x4 set
3: cpu-bind=MASK - sdumont6028, task  3  0 [25463]: mask 0x1 set
4: cpu-bind=MASK - sdumont6028, task  4  1 [25464]: mask 0x2 set
5: cpu-bind=MASK - sdumont6028, 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 sequana_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 - sdumont6027, task  0  0 [25051]: mask 0x3f set
1: cpu-bind=MASK - sdumont6027, task  1  1 [25052]: mask 0xfc0 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [25053]: mask 0x3f000 set
3: cpu-bind=MASK - sdumont6027, task  3  3 [25054]: mask 0xfc0000 set
4: cpu-bind=MASK - sdumont6028, task  4  0 [28964]: mask 0x3f set
5: cpu-bind=MASK - sdumont6028, task  5  1 [28965]: mask 0xfc0 set
6: cpu-bind=MASK - sdumont6028, task  6  2 [28966]: mask 0x3f000 set
7: cpu-bind=MASK - sdumont6028, 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 sequana_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 - sdumont6027, task  0  0 [25051]: mask 0x3f set
1: cpu-bind=MASK - sdumont6027, task  1  1 [25052]: mask 0xfc0 set
2: cpu-bind=MASK - sdumont6027, task  2  2 [25053]: mask 0x3f000 set
3: cpu-bind=MASK - sdumont6027, task  3  3 [25054]: mask 0xfc0000 set
4: cpu-bind=MASK - sdumont6028, task  4  0 [28964]: mask 0x3f set
5: cpu-bind=MASK - sdumont6028, task  5  1 [28965]: mask 0xfc0 set
6: cpu-bind=MASK - sdumont6028, task  6  2 [28966]: mask 0x3f000 set
7: cpu-bind=MASK - sdumont6028, 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_sequana
$ git clone https://github.com/robertopsouto/ESD2025.git
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
ESD2025/
├── README.md
└── sequana
    ├── env_openmpi
    └── NPB3.4.2-MZ
```

Exemplo: NAS Parallel Benchmark (BT-MZ)

```
ESD2025/
├── README.md
└── 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 ESD2025/sequana/
$ ls -A1
env_openmpi
NPB3.4.2-MZ

$ cat env_openmpi
module load openmpi/gnu/4.1.4_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
  ../ESD2025/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 openmpi/gnu/4.1.4_sequana

DIR=$PWD

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

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 sdumont6000

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

$ sbatch -N1 -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
```

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     sdumont6000
```

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

sequana_gpu | sequana_gpu_dev | sequana_gpu_long

As filas sequana_gpu* permitem o compartilhamento do nó por múltiplos jobs (até 4). Para submeter jobs para essas filas, é necessário fazer utilização do **GRES do SLURM.**

- Para garantir o melhor uso dos recursos, o parâmetro --exclusive **não deve ser utilizado**!
- Cada nó computacional do SDumont expansão (Sequana X1120) possui 4 aceleradores NVIDIA V100, 48 núcleos CPU e 384GB de memória RAM.
- Para alocar uma ou mais GPUs, é obrigatório especificar a quantidade de aceleradores (parâmetros --gpus, --gpus-per-node, --gpus-per-socket, --gpus-per-task).
- Caso não seja informada a quantidade de GPU's, o job não entrará em execução e ficará pendente em fila com a REASON **QOSMinGRES.**
- Quando não for informado a quantidade de cores desejados (parâmetros --cpus-per-gpu, -n, --ntasks, --ntasks-per-node, --ntasks-per-socket, --tasks-per-node), o SLURM reservará 12 cores por GPU solicitada.
- Quando o usuário não especificar a quantidade de memória a ser utilizada (parâmetros --mem-per-gpu, --mem, --mem-per-cpu), o SLURM reservará 8GB por core solicitado.
- Por padrão, se não for especificada a quantidade de cores e memória, para cada GPU solicitada serão alocados 12 cores e 96GB de memória RAM.
- A GPU (unidade) não é compartilhável, portanto ao ser alocada para um job, nenhum outro job conseguirá utilizá-la.

Uso das filas com GPU

sequana_gpu | sequana_gpu_dev | sequana_gpu_long

As filas sequana_gpu* permitem o compartilhamento do nó por múltiplos jobs (até 4). Para submeter jobs para essas filas, é necessário fazer utilização do **GRES do SLURM.**

- Parâmetros do slurm para alocações referentes a GPU
  - --gpus=n: Número total de GPUs solicitadas para o job
  - --gpus-per-node=n: Número de GPUs solicitadas para cada nó alocado
  - --gres=gpu:n: Número de GPUs solicitadas por nó (o mesmo que a combinação de --gpus=n --gpus-per-node=n)
  - --gpus-per-socket=n: Número de GPUs solicitadas para cada socket alocada
  - --gpus-per-task=n: Número de GPUs solicitadas para cada tarefa alocada
  - --cpus-per-gpu=n: Número de CPUs solicitadas para cada GPU alocada (Valor padrão: 12 CPUs para cada GPU alocada)
  - --mem-per-gpu=n: Memória RAM solicitada para cada GPU alocada (Valor padrão: 8GB para cada CPU alocada)

Uso das filas com GPU - SEQUANA

- Carregar o ambiente com o NVIDIA CUDA Toolkit v12.6
  ```
  $ module load cuda/12.6_sequana
  ```

- Verificar a configuração de GPU do nó
  ```
  $ srun -p sequana_gpu_dev -n 1 --gpus=2 nvidia-smi
  ```

```
srun: job 11258811 queued and waiting for resources
srun: job 11258811 has been allocated resources
Thu Jan 23 13:34:26 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 560.35.03              Driver Version: 560.35.03      CUDA Version: 12.6     |
|-----------------------------------------+------------------------+----------------------+
| 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 V100-SXM2-32GB           On  |   00000000:60:00.0 Off |                    0 |
| N/A   45C    P0             46W /  300W |       1MiB /  32768MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   1  Tesla V100-SXM2-32GB           On  |   00000000:61:00.0 Off |                    0 |
| N/A   47C    P0             46W /  300W |       1MiB /  32768MiB |      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 - SEQUANA

- Baixar exemplos (samples) do repositório Github da NVIDIA

```
$ cd $SCRATCH
$ git clone https://github.com/NVIDIA/cuda-samples.git
Cloning into 'cuda-samples'...
remote: Enumerating objects: 19507, done.
remote: Counting objects: 100% (4227/4227), done.
remote: Compressing objects: 100% (655/655), done.
remote: Total 19507 (delta 3932), reused 3572 (delta 3572),
pack-reused 15280 (from 2)
Receiving objects: 100% (19507/19507), 133.82 MiB | 31.31
MiB/s, done.
Resolving deltas: 100% (17105/17105), done.
Updating files: 100% (4026/4026), done.
```

Uso das filas com GPU - SEQUANA

- Verificar exemplo de multiplicação de matrizes

```
$ cd $SCRATCH/cuda-samples/Samples/0_Introduction/matrixMul
$ ls
Makefile      matrixMul_vs2017.sln      matrixMul_vs2019.sln
matrixMul_vs2022.sln      NsightEclipse.xml
matrixMul.cu  matrixMul_vs2017.vcxproj
matrixMul_vs2019.vcxproj  matrixMul_vs2022.vcxproj  README.md
```

Uso das filas com GPU - SEQUANA

- Compilar exemplo de multiplicação de matrizes

```
$ make
/petrobr/app_sequana/cuda/cuda-12.6/bin/nvcc -ccbin g++ -I../../../Common  -m64    --threads 0
--std=c++11 -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_89,code=sm_89 -gencode arch=compute_90,code=sm_90
-gencode arch=compute_90,code=compute_90 -o matrixMul.o -c matrixMul.cu
/petrobr/app_sequana/cuda/cuda-12.6/bin/nvcc -ccbin g++   -m64      -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_89,code=sm_89 -gencode arch=compute_90,code=sm_90 -gencode
arch=compute_90,code=compute_90 -o matrixMul matrixMul.o
mkdir -p ../../../bin/x86_64/linux/release
cp matrixMul ../../../bin/x86_64/linux/release


$ ls
Makefile   matrixMul.cu  matrixMul_vs2017.sln      matrixMul_vs2019.sln      matrixMul_vs2022.sln
NsightEclipse.xml matrixMul  matrixMul.o   matrixMul_vs2017.vcxproj  matrixMul_vs2019.vcxproj
matrixMul_vs2022.vcxproj  README.md
```

Uso das filas com GPU - SEQUANA

- Executar exemplo de multiplicação de matrizes

```
$ srun -p sequana_gpu_dev --gpus=1 ./matrixMul -wA=1024 -hA=256 -wB=256
-hB=1024
srun: job 11258798 queued and waiting for resources
srun: job 11258798 has been allocated resources
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Volta" with compute capability 7.0

MatrixA(1024,256), MatrixB(256,1024)
Computing result using CUDA Kernel...
done
Performance= 2405.58 GFlop/s, Time= 0.056 msec, Size= 134217728 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

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

Uso das filas com GPU - SEQUANA

- Executar exemplo de multiplicação de matrizes

```
$ srun -p sequana_gpu_dev --gpus=1 ./matrixMul -wA=2048 -hA=256 -wB=256
-hB=2048
srun: job 11258802 queued and waiting for resources
srun: job 11258802 has been allocated resources
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Volta" with compute capability 7.0

MatrixA(2048,256), MatrixB(256,2048)
Computing result using CUDA Kernel...
done
Performance= 2472.28 GFlop/s, Time= 0.109 msec, Size= 268435456 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

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

Uso das filas com GPU - SEQUANA

- Executar exemplo de multiplicação de matrizes

```
$ srun -p sequana_gpu_dev --gpus=1 ./matrixMul -wA=4096 -hA=256 -wB=256
-hB=4096
srun: job 11258805 queued and waiting for resources
srun: job 11258805 has been allocated resources
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Volta" with compute capability 7.0

MatrixA(4096,256), MatrixB(256,4096)
Computing result using CUDA Kernel...
done
Performance= 2212.56 GFlop/s, Time= 0.243 msec, Size= 536870912 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

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

Uso das filas com GPU - SEQUANA

- Executar exemplo de multiplicação de matrizes

```
$ srun -p sequana_gpu_dev --gpus=1 ./matrixMul -wA=8192 -hA=256 -wB=256
-hB=8192
srun: job 11258807 queued and waiting for resources
srun: job 11258807 has been allocated resources
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Volta" with compute capability 7.0

MatrixA(8192,256), MatrixB(256,8192)
Computing result using CUDA Kernel...
done
Performance= 2210.97 GFlop/s, Time= 0.486 msec, Size= 1073741824 Ops,
WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

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