Super Computación y Cálculo Científico
Universidad Industrial de Santander

H - camp

HPC Ecosystems Interaction

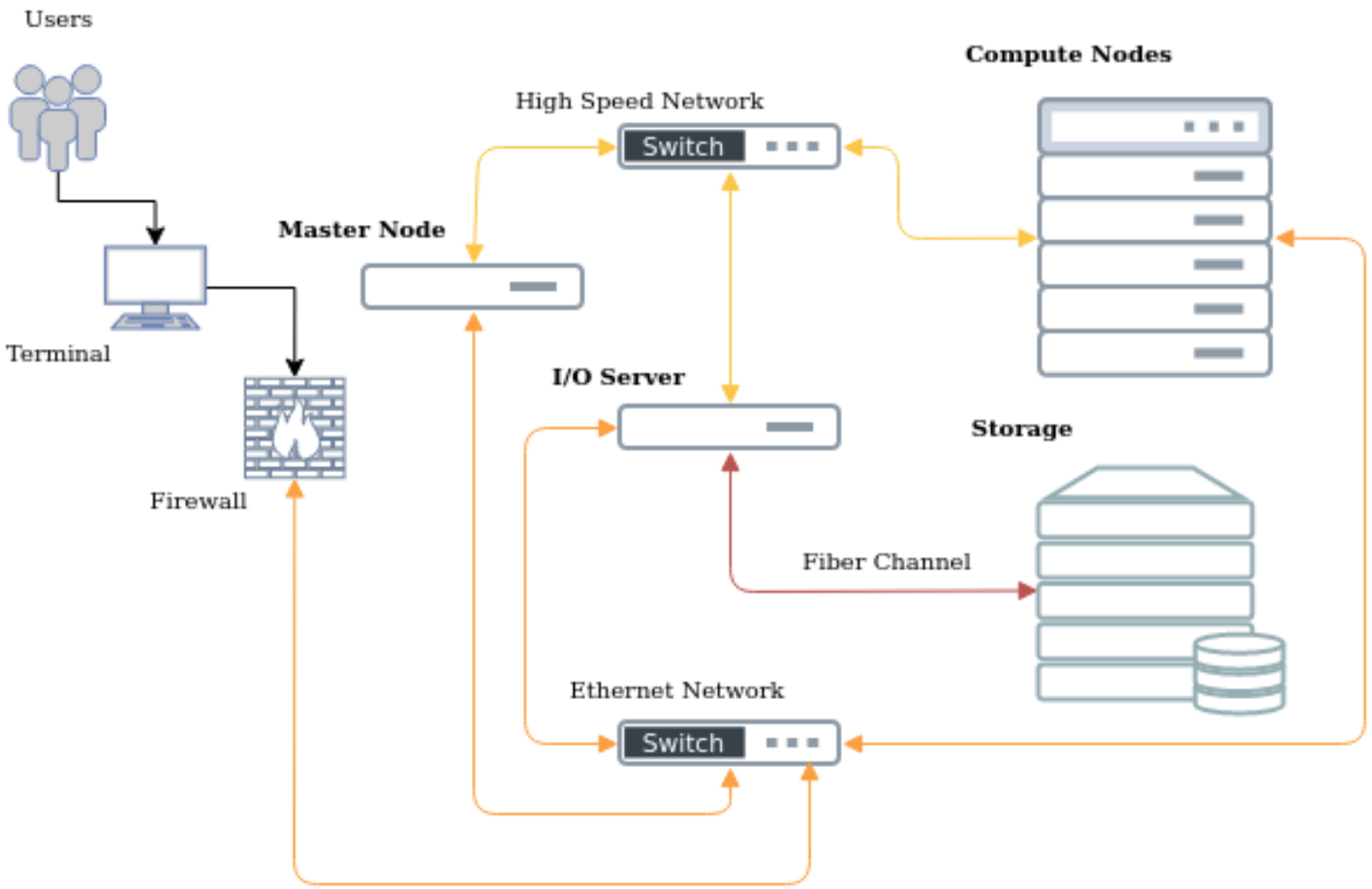
SLURM - Simple Linux Utility for Resource Management

GUANE

Universidad Industrial de Santander

HPC – Basic Architecture

H - camp



HPC - Base System

H - camp

- NTP
- NFS
- Infiniband support
- Memory usage limits
- HPC Modules – LMOD
- PowerShell
- NHC

H - camp

SLURM & MUNGE

slurm.conf

SchedulerType -> SchedulerType=sched/backfill
SelectType ->
SelectTypeParameters=CR_Core,CR_Core_Default_Dist_Block
SelectTypeParameters - >
SelectTypeParameters=CR_Core,CR_Core_Default_Dist_Block
PriorityType - > PriorityType=priority/multifactor

SLURM - NHC -> HealthCheckProgram

slurm
workload manager

LDAP - Lightweight Directory Access Protocol

H - camp

- Good practice in implementing LDAP is using dynamic groups that allow you to assign different levels of access to different storage spaces within the HPC platform.
- Storage spaces such as the user's home folder and project and research group folders must have restrictions implemented through disk quotas in conjunction with LDAP.

GUANE - GpUs Advanced computiNg Environment

H - camp

HPC – SC3UIS

Technical specifications - GUANE

16 nodes ProLiant SL390s G7

- 8 nodes:
  - 2 Intel(R) Xeon(R) CPU E5645 @ 2.40GHz.
  - 104 GB RAM
  - 1 disk SAS de 200GB
  - 8 GPU Tesla M2075
- 3 nodes:
  - 2 Intel(R) Xeon(R) CPU E5645 @ 2.40GHz.
  - 104 GB RAM
  - 1 disk SAS de 200GB
  - 8 GPU Tesla S2050
- 5 nodes:
  - 2 Intel(R) Xeon(R) CPU E5640 @ 2.67GHz
  - 104 GB RAM
  - 1 disk SAS de 200GB
  - 8 GPU Tesla S2050

Network

- 10Gbit/s Ethernet – Administration
- 40 Gb/sec Infiniband

GUANE - GpUs Advanced computiNg Environment

H - camp

HPC – SC3UIS

OTHER NODES

THOR
Technical Specifications

- ProLiant DL580 Gen9
- 4 Intel(R) Xeon(R) CPU E7-8867 v3 @ 2.50GHz – 128 Cores
- 1320732708 kB – 1.2TB RAM

YAJÉ
Technical Specifications

- ProLiant ML350 Gen9
- 1 Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz – 6 Cores
- 49031292 kB – 48GB RAM
- 1 NVIDIA GeForce GTX Titan X 12 GB

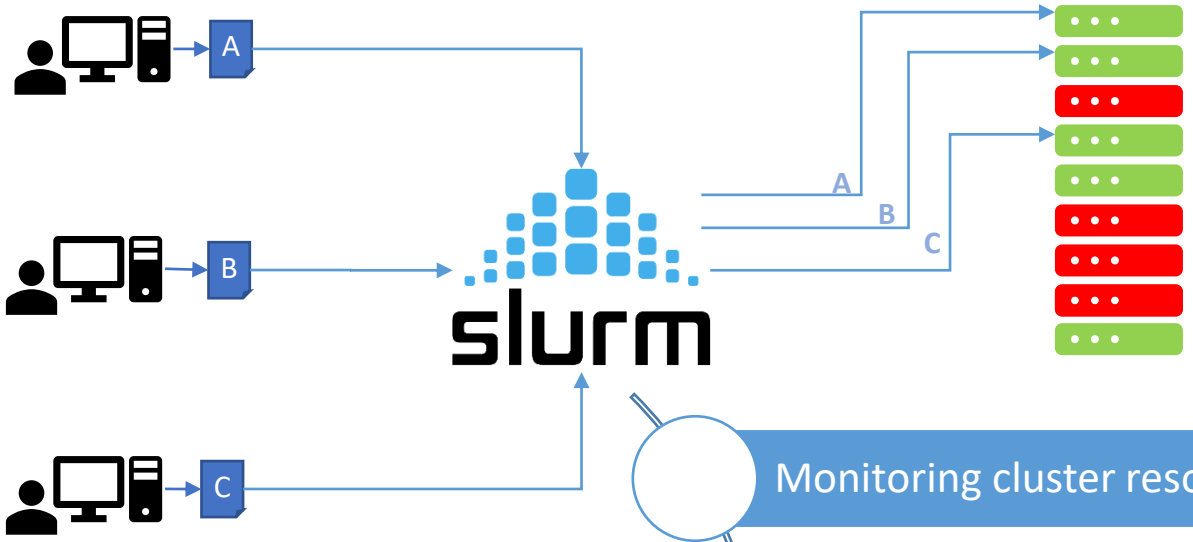
FELIX (Framework to Enhance artificiaL Intelligence applications eXecution)
Technical Specifications

- ProLiant DL580 G7
- 4 Intel(R) Xeon(R) CPU X7560 @ 2.27GHz – 64 Cores
- 131844368 kB – 128GB RAM
- 2 NVIDIA GeForce GTX Titan X 12 GB

What is SLURM?

H - camp

SLURM is open-source Linux cluster management and job management software.

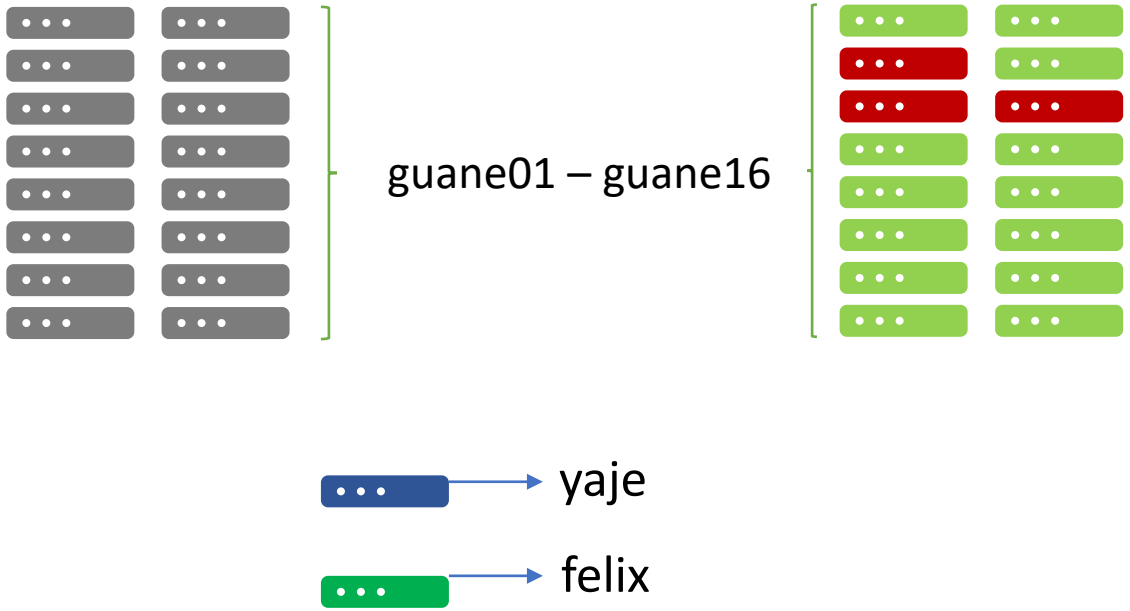


- Monitoring cluster resources
- Collects the resources requested by users.
- Assign priority to jobs.
- Runs the jobs on the assigned compute nodes.

Partitions

Computing nodes are grouped into logical sets called partitions that depend on their hardware characteristics or function:

	PARTITION
	normal*
	guane_16_cores
	guane_24_cores
	Viz
	deepL



CONNECTION TO GUANE

H - camp

```
SSH username@ip-address or hostname
```

ssh user_name@toctoc.sc3.uis.edu.co

ssh guane

Use of SLURM

H - camp

sinfo

- Shows the information of the nodes and partitions.
- An asterisk (*) after the partition name indicates that it is the default partition.
- An asterisk (*) after the node status indicates that it is not responding.

```
[user_name@guane ~]# sinfo
```

```
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal*           up   infinite      4    mix guane[03,05,09,16]
normal*           up   infinite      8  alloc guane[01-02,04,10,12-15]
normal*           up   infinite      2   idle guane[06,11]
guane_16_cores    up   infinite      2    mix guane[03,05]
guane_16_cores    up   infinite      1   idle guane06
guane_24_cores    up   infinite      2    mix guane[09,16]
guane_24_cores    up   infinite      8  alloc guane[01-02,04,10,12-15]
guane_24_cores    up   infinite      1   idle guane11
Viz               up   infinite      1   idle yaje
deepL             up   infinite      1  alloc felix
```

Use of SLURM

H - camp

```
squeue -u student_30
```

- Displays the job queue for user student_30

```
JOBID       PARTITION         NAME       USER  ST         TIME NODE NODELIST(REASON)
18276           deepL mafft_09_mpi    druedap   R      7:46:29    1 felix
18277          normal    gisaid_04    druedap   R      7:39:41    1 guane02
18282  guane_24_cores    gisaid_03    druedap   R      2:33:47    1 guane04
```

```
[user_name@guane ~]# squeue
```

```
JOBID       PARTITION         NAME       USER  ST         TIME NODE NODELIST(REASON)
17772  guane_24_cores        boinc  latorresn   R  23-08:13:22    1 guane10
18014  guane_24_cores         sim1  ccbernalc   R  10-22:24:19    1 guane15
18015          normal      orcaLNi geramirezc   R  10-21:48:41    1 guane01
18046  guane_24_cores         sim1  ccbernalc   R   9-12:22:39    1 guane13
18252          normal    cubes3.sh jmpachecoa   R     22:22:32    1 guane03
18275  guane_24_cores         sim1  arromerob   R      8:28:06    1 guane14
18276           deepL mafft_09_mpi    druedap   R      7:47:35    1 felix
18277          normal    gisaid_04    druedap   R      7:40:46    1 guane02
18279  guane_24_cores         sim1  arromerob   R      6:09:15    1 guane12
18281          normal         bash  emvargasd   R      4:40:48    1 guane16
18282  guane_24_cores    gisaid_03    druedap   R      2:34:52    1 guane04
18283  guane_16_cores    cubes1.sh crcarvajal   R      1:28:47    1 guane05
18284  guane_24_cores    cubes2.sh crcarvajal   R      1:26:58    1 guane09
```

STATUS

R = Running

PD = Pending

CA = Cancelled

Use of SLURM

H - camp

srun *options*

- Allows you to run an application directly with options specified by the user in options parameters.

```
[user_name@guane ~]# srun --ntasks=4 --partition=normal --label /bin/hostname
```

```
[user_name@guane ~]# srun -n 4 -p normal -l /bin/hostname
```

```
2:  guane01.uis.edu.co
1:  guane01.uis.edu.co
0:  guane01.uis.edu.co
3:  guane01.uis.edu.co
```

Use of SLURM

H - camp

salloc *options*

- Gets the assignment of a job with console access.
- The resources reserved for the job are those specified in options.
- Allows you to make an interactive reservation.

Interactive Reservation

```
[user_name@guane ~]# salloc --nodes=1 --partition=normal --exclusive srun --pty /bin/bash
```

```
[user_name@guane ~]# srun --nodes=1 --partition=normal --exclusive  --pty /bin/bash
```

H - camp

Environment Modules– Software in GUANE

- Modules are a packaging of environment variables within a script.
- One module is defined per application, which defines an appropriate environment for its execution.
- Command list:
  - module available
  - module load MODULE_NAME
  - module unload
  - module list
  - module purge

H - camp

Environment Modules

module avail

- Shows all the modules available on the platform.

```
------------------------------------------------------------------------------ /opt/ohpc/pub/modulefiles ----------------
   Analytics/Anaconda/python3              Chemistry/gamess/2019R2                     containers/docker/19.03.9
   Analytics/Darknet/1.0                   Chemistry/gromacs/2018.8_GPU                devtools/cmake/3.14.3
   Analytics/Julia/1.0.5                   Chemistry/gromacs/2019.3              (D)   devtools/cuda/7.5
   Analytics/Julia/1.2.0             (D)   Chemistry/nwchem/6.8                        devtools/cuda/8.0
   Analytics/Octave/5.1.0                  Chemistry/orca/4.0.1.2                      devtools/cuda/9.1
   Bioinformatics/Bioconda/python3         Chemistry/orca/4.2                    (D)   devtools/cuda/10.1          (D)
   Bioinformatics/Geneious/9.1.8           EasyBuild/3.9.4                             devtools/gcc/5.3.0
   Bioinformatics/NGSEP/4.0.1              Matlab/R2020a                               devtools/gcc/6.2.0
   Bioinformatics/SpreaD3/0.9.6            QuantumATK/2018.06-SP1-1/2018.06-SP1-1      devtools/gcc/7.4.0
   Bioinformatics/TempEst/1.5.3            QuantumATK/2019.03-SP1/2019.03-SP1          devtools/gcc/8.3.0
   Bioinformatics/clustalOmega/1.2.4       QuantumExpresso/6.5                         devtools/gcc/9.2.0          (D)
   Bioinformatics/jmodeltest/2.1.10        autotools                                   devtools/globalarrays/5.6.1
   Bioinformatics/megaCC/10.1.8            boinc/7.14.2                                devtools/intel/2016.4
   CAE/ansys/2020r1                        clustershell/1.8.2                          devtools/intel/2017.8
   CFD/OpenFOAM/2.4.0                      cmake/3.15.4                                devtools/intel/2019.4
   CFD/OpenFOAM/1906                 (D)   comsol/5.3a                                 devtools/intel/2020.1       (D)

---------------------------------------------------------- /opt/ohpc/admin/spack/0.12.1/share/spack/modules/linux-centos7-
   ncurses-6.1-gcc-8.3.0-fazhf5h    openblas-0.3.3-gcc-8.3.0-byhg6e2    pcre-8.42-gcc-8.3.0-4rago5n    pkgconf-1.4.2-gcc-8.3

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

H - camp

Environment Modules

```
module load module_name
```

- Loads the environment variables corresponding to the selected module (*module_name*)

```
[user_name@guane ~]# module load CFD/OpenFOAM/1906
```

```
module list
```

- List all modules that have been loaded with the **module load** command. You should keep in mind that you can load one or more modules simultaneously.

H - camp

Environment Modules

module unload *module_name*

- Removes all environment variables corresponding to the selected module(*module_name*)

```
[user_name@guane ~]# module unload CFD/OpenFOAM/1906
```

module purge

- Removes all environment variables from all modules that are loaded in the current session

H - camp

BATCH JOB SCRIPT

myjob.slurm

```
#!/bin/bash

# Resource request
#SBATCH --partition=guane_16_cores
#SBATCH --nodes=1
#SBATCH --ntask=1
#SBATCH --ntasks-per-node=1
#SBATCH --mem=1G

# Job Execution Time
#SBATCH --time=1-12:30:00

# Job name and output files
#SBATCH --job-name=myjob
#SBATCH --output=myjob.out
#SBATCH --error=myjob.err

# Loading of the environment module
module load CFD/OpenFOAM/1906
# Execution
blockMesh
```

Preliminaries

- Specify the command interpreter (Bash).
- It should always be the first line.

SLURM Directives

- They should always start with #SBATCH
  - They are ignored by bash but interpreted by SLURM.
- Comments can be made before, between, or after directives.
- They must be placed before loading the modules and executing the job.

Script commands

- Loading the modules required for the execution of the work
- Commands that you want to execute in the computing nodes
  - Executable of the loaded application.
  - Programming commands can be written in bash.

Use of SLURM

H - camp

sbatch *batch_file*

- Sends the batch_file to SLURM for execution.
- If the submission is successful, SLURM returns the job ID

```
[user_name@guane ~]# sbatch myjob.slurm
```

```
[user_name@guane ~]# squeue -u druedap
```

```
JOBID       PARTITION         NAME       USER  ST         TIME NODE NODELIST(REASON)
18276           deepL mafft_09_mpi    druedap   R      7:46:29    1 felix
18277          normal     gisaid_04    druedap   R      7:39:41    1 guane02
18282  guane_24_cores     gisaid_03    druedap   R      2:33:47    1 guane04
```

Use of SLURM

H - camp

scancel *jobid*

- Sends a signal to the job and/or its threads.
- By default, the signal sent is SIGKILL for the termination of the job.
- The job that is canceled is the one that corresponds to jobid.
- The jobid is obtained by executing the **sinfo** command.

```
[user_name@guane ~]# scancel 12345
```

- Filters can be used for job cancellation

```
[user_name@guane ~]# scancel --user=sutedent_30
```

H - camp

EXAMPLE – Use of OpenFOAM in GUANE

- Download the examples *
  wget http://www.hpc.lsu.edu/training/weekly-materials/Downloads/intro_of.tar.gz
- Unzip the sample file
  tar zxf intro_of.tar.gz
- Enter the folder of the first example to test
  cd intro_of/cavity
- Create the batch job script
  nano cavity.slurm
- Launch of work in SLURM
  sbatch cavity.slurm

```
#!/bin/bash

# Resource request
#SBATCH --partition=normal
#SBATCH --nodes=1
#SBATCH --ntasks=16
#SBATCH --ntasks-per-node=16

# Job Execution Time
#SBATCH --time=0-00:10:00

# Job name and output files
#SBATCH --job-name=cavity
#SBATCH --output=cavity_%j.out
#SBATCH --error=cavity_%j.err

# Loading of the environment module
module load CFD/OpenFOAM/2.4.0

# Execution
blockMesh    # (generate mesh information)
iconFoam     # (running the PISO solver)
foamToVTK    #(convert to VTK format, optional)
```

*** Example URL:**
http://www.hpc.lsu.edu/training/weekly-materials/2016-Spring/intro_of_20160224.pdf

H - camp

Recommendations and others

Take account of

H - camp

- SSH login
- Multi-factor authentication
- How Many login nodes are there?
- Connection troubleshooting guide.
- Live Status from services.
  - Nodes alive
  - Filesystems
  - Mass Storage systems
  - Planned Outages
- System Ticket platform.

- Wiki Documentation
  - Software
  - Interactive Nodes
  - Data sharing approach
  - FAQs.

General recommendations

H - camp

- Don't run jobs in login nodes.
- Don't run may jobs at time.
- Don't use all user space on Scratch partition.
- Clean up your $HOME directory frecuently

These general recommendations may change over time and may need to be adjusted for your HPC workloads

Remember that HPCs are shared systems and try avoid allocating resources which you don't use

Job Generator

H - camp

demo.jobgenerator.local

Super Computación y Cálculo Científico UIS

SLURM JOB SCRIPT GENERATOR

Enter your cluster and job specs to create a template job script.

Job Name: grom.job

grom

Email Address:

user@mail.com

Job events:

Begin End Abort

Cluster:

guane

Partition:

manycores16

Aditional Features

Especific features for: manycores16

CPU_WTM_E5-640

GPU_F_S2050

Gpu

gres: 1

Exclusive

Requeue on node fail

Number of Nodes: 2

Tasks (processes): 4

Tasks per node: 2

Memory per core:

50000 MB

Walltime:

00 D 5 H 10 M 00 S

Output File: grom.%j.out

grom

Error File: grom.%j.err

grom

Program here:

```
module load gcc/5.2.0  openmpi/1.10.2 gromacs/2016.3-CUDA

srun --mpi=pmi2 --accel-bind=g --ntasks=$SLURM_NTASKS gmx mdrun -v
```

grom.job

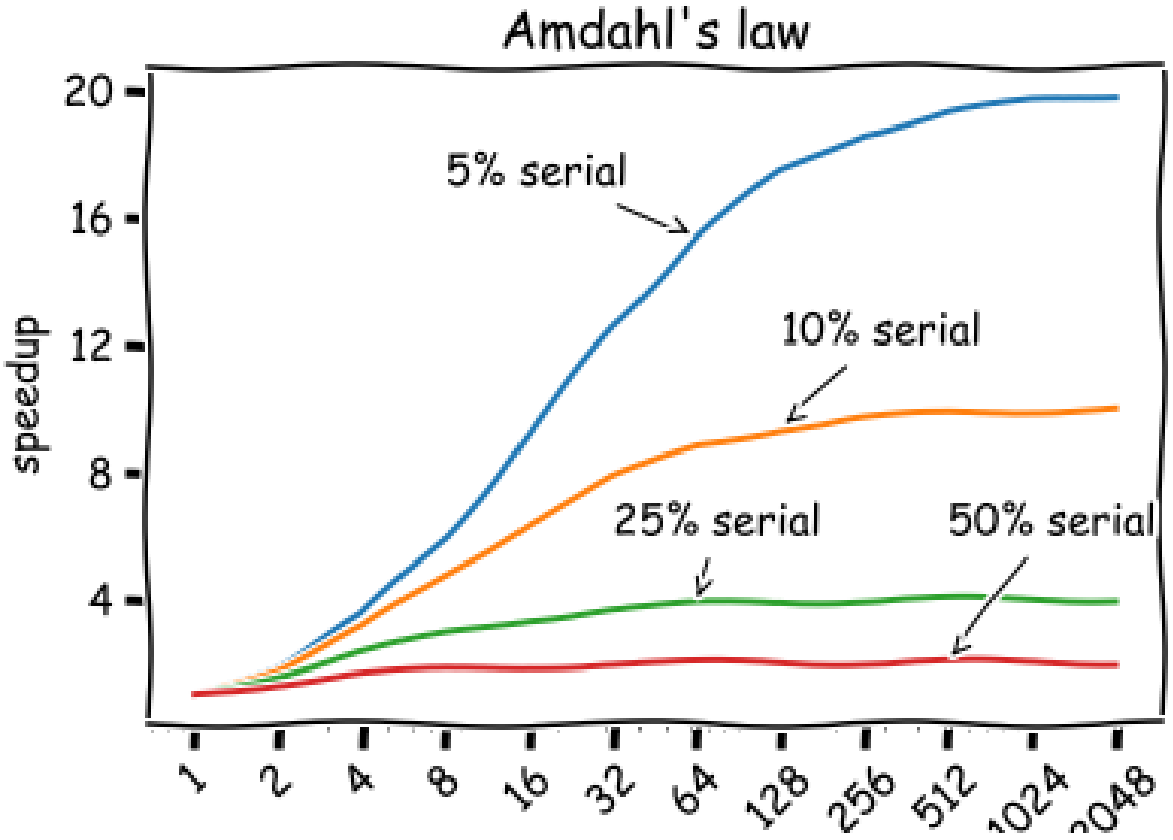
Copy

```
### Number of nodes requested.
#SBATCH --nodes=2

### Number of tasks(processes).
#SBATCH --ntasks=4

### Amount memory per core in MB.
#SBATCH --mem-per-cpu=50000M

### Wall clock time limit in Days-HH:MM:SS
#SBATCH --time=00-05:10:00

### Send an email when the job status changes.
#SBATCH --mail-user=user@mail.com
#SBATCH --mail-type=begin,fail

### File in which to store job output.
#SBATCH --output=grom_%j.out

### File in which to store job error messages.
#SBATCH --error=grom_%j.err

## Insert code, and run your programs here ##

module load gcc/5.2.0  openmpi/1.10.2 gromacs/2016.3-CUDA

srun --mpi=pmi2 --accel-bind=g --ntasks=$SLURM_NTASKS gmx mdrun

## Submit script example: ##
## sbatch  grom.job
```

Generate

Computational Perfomance

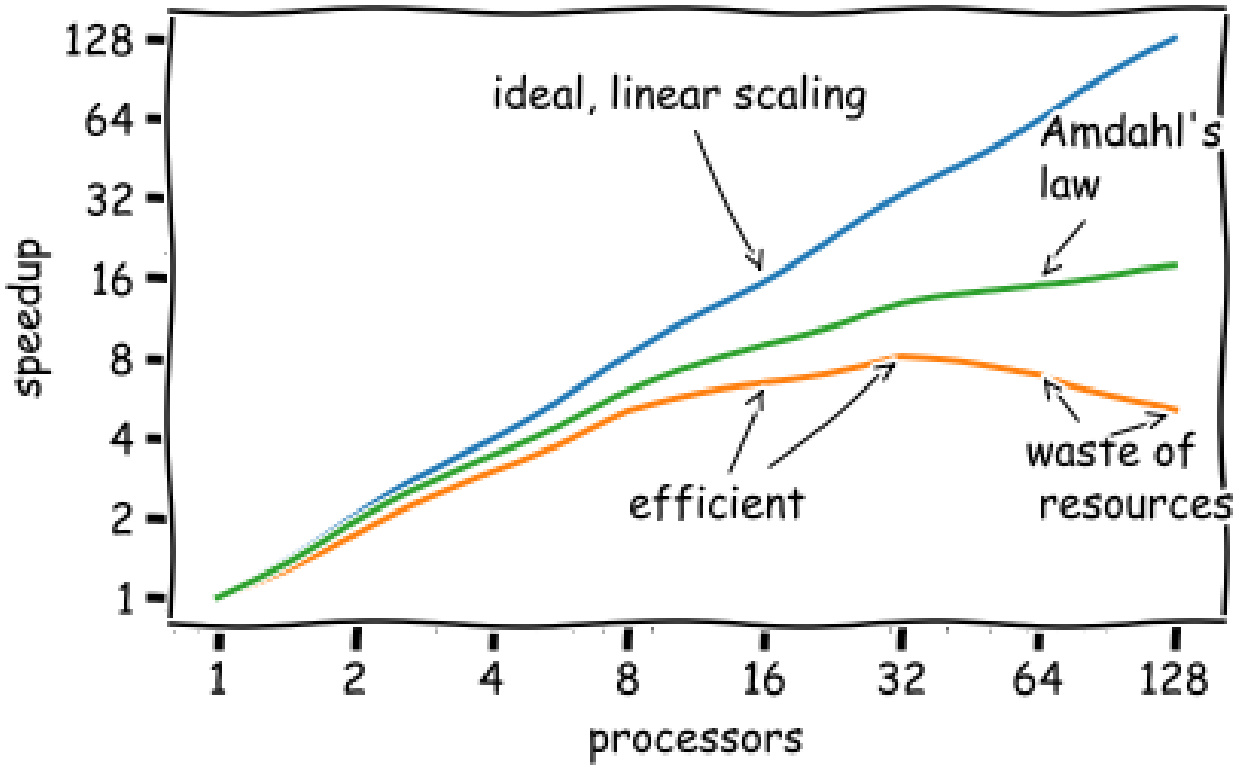
H - camp



- Performance improvements from parallel execution do not scale linearly.
- Parallel programming allows application to take advantage of parallel hardware, serial code will not 'just work'.
- Common case of distributed memory parallelism, MPI (Message passing Interface)

Computational Perfomance

H - camp



Computational Perfomance

H - camp

Scope:

- Calculate de expected computational efficiency of the job.
- Test the behavior of a program when executed in parallel.
- Scalability from **1...n** processors.

Tools:

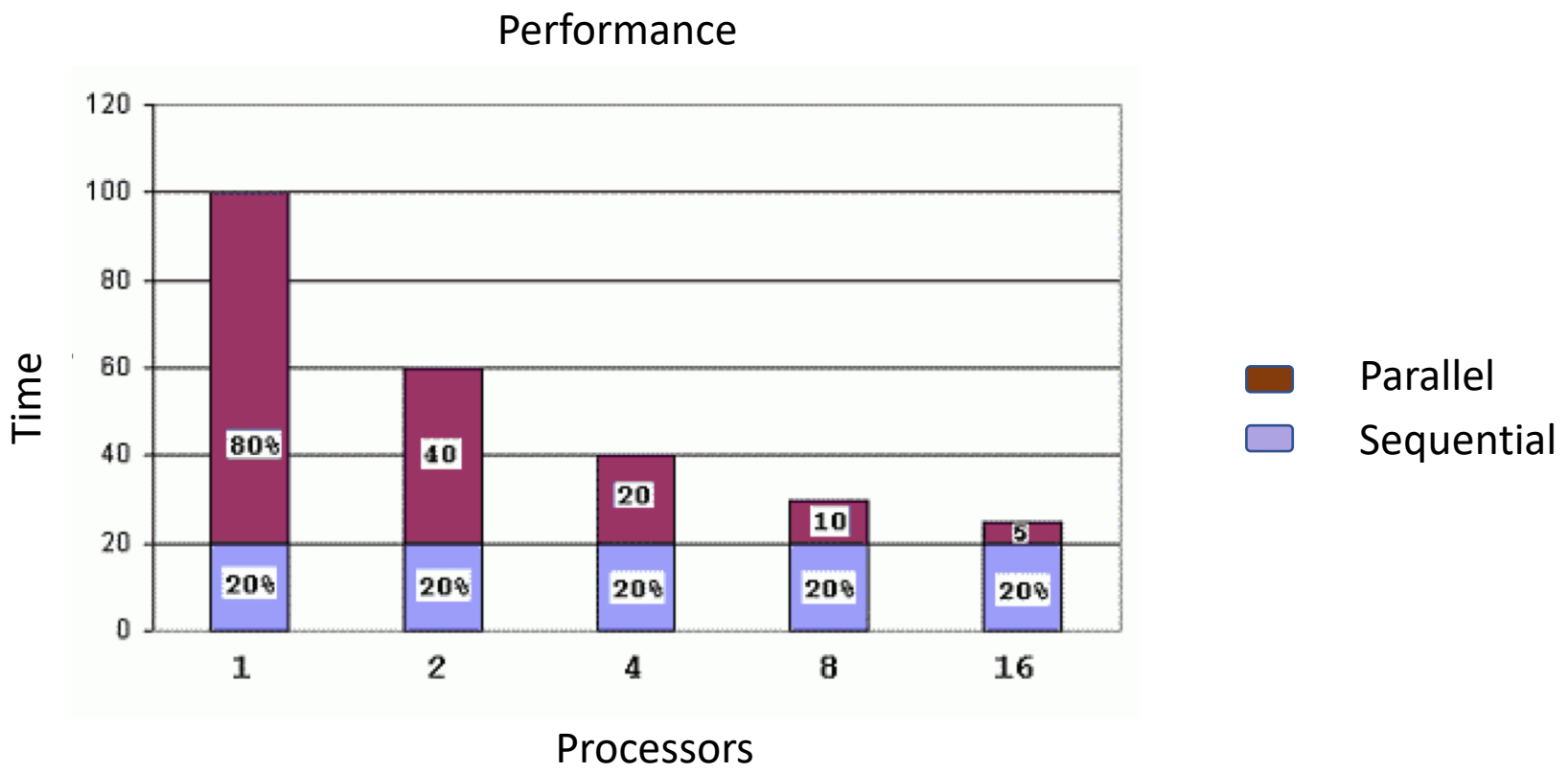
- **Speed Up**: factor that indicates the gain through parallelization. → $S(n) = T(1)/T(n)$

- **Efficiency**: resource use efficient metric. → $E(n) = \frac{S(n)}{n} = \frac{T(1)}{n * T(n)}$

- **Efficiency is archived when the measurement is kept at factor 0.5.**

Computational Perfomance

H - camp



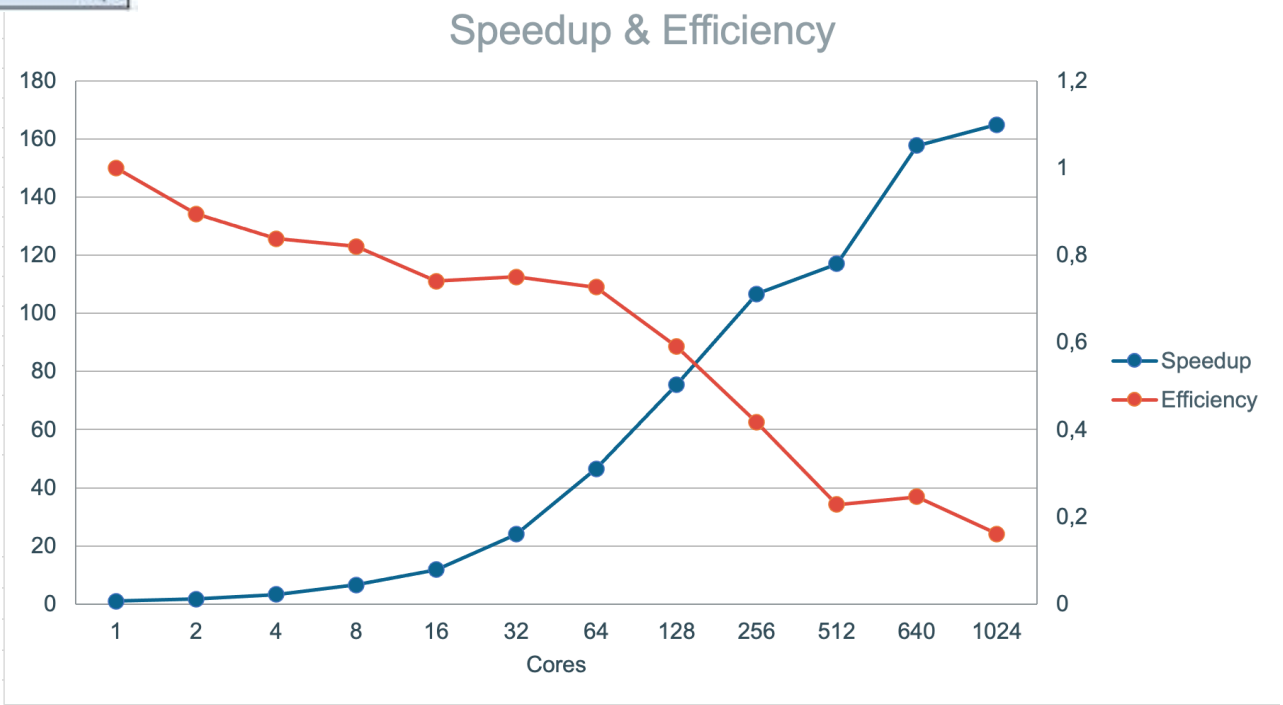
$$S(n) = T(1)/T(n) \qquad E(n) = \frac{S(n)}{n} = \frac{T(1)}{nT(n)}$$

N =	**2**	**4**	**8**	**16**
S =	100/(40+20) = 1.666	2.5	3.333	4
E =	100/(2*60) = 0.83 = 1.666/2	0.625	0.416	0.25

Computational Perfomance

H - camp

Task	Task per Node	Time	Speedup	Efficiency
1	1	1:00:27	1,0	1,0
2	2	0:33:47	1,8	0,9
4	4	0:18:02	3,4	0,8
8	8	0:09:13	6,6	0,8
16	16	0:05:06	11,9	0,7
32	16	0:02:31	24,0	0,8
64	16	0:01:18	46,5	0,7
128	16	0:00:48	75,6	0,6
256	16	0:00:34	106,7	0,4
512	16	0:00:31	117,0	0,2
640	16	0:00:23	157,7	0,2
1024	32	0:00:22	164,9	0,2



Compilation

FAMIILIES

- GNU
- INTEL
- NVIDIA

Compilers

- Gcc
- G++
- Fortran
- Nvcc
- Icc
- Ifort
- Openmp
- Mpi
- ...

gcc -Olevel [options] [source files] [object files] [-o output file]

option	optimization level	execution time	code size	memory usage	compile time
-O0	optimization for compilation time (default)	+	+	-	-
-O1 or -O	optimization for code size and execution time	-	-	+	+
-O2	optimization more for code size and execution time	--		+	++
-O3	optimization more for code size and execution time	---		+	+++
-Os	optimization for code size		--		++
-Ofast	O3 with fast none accurate math calculations	---		+	+++

Compilation Automation Tools

H - camp

main.c

```c
#include "print.h"

int main() {
  printInt(42);
}
```

print.h

```c
void printInt(int x);
```

print.c

```c
#include<stdio.h>

void printInt(int x) {
  printf("%d\n", x);
}
```

```
→  demo ls
Makefile main.c   print.c  print.h
→  demo
→  demo
→  demo gcc -o main print.c main.c -I .
→  demo ls
Makefile main     main.c   print.c  print.h
→  demo ./main
42
→  demo
```

Target

```makefile
print.o: print.c print.h
        gcc -c print.c
```

prerequisites

commads

```makefile

.PONNY: clean

CC = gcc

app: main.o print.o
    $(CC) main.o print.o -o main

main.o: main.c
    $(CC) -c main.c

print.o: print.c print.h
    $(CC) -c print.c

clean:
    rm -f a.out *.o

"Makefile" 19L, 181B
```

GNU Make

```
→  demo ls
Makefile main.c   print.c  print.h
→  demo make
gcc -c main.c
gcc -c print.c
gcc main.o print.o -o main
→  demo ls
Makefile main     main.c   main.o   print.c  print.h  print.o
→  demo ./main
42
```

Compilation take account of

H - camp

- **Make**
  - Carefully review of phase options
    ./configure --help
  - Verbose mode **VERBOSE=1**
  - Use compile parallelism. ***-j***

- **Guide when an error comes out**
  - Compile with single thread ***–j1***
  - Always look for the first error in the list
  - Compile again
  - Google
  - Compile without optimizations
  - Patience.

Storage Perfomance

H - camp

- Principal
  - $HOME
  - $SCRATCH
- Others shared file system

- **/dev/shm**

*You can use **/dev/shm** to improve the performance of application software in parallel tasks or overall Linux system performance. On heavily loaded system, it can make tons of difference.*

```
[cbernal@guane03 ~]$ df -h
Filesystem                      Size  Used Avail Use% Mounted on
devtmpfs                         52G     0   52G   0% /dev
tmpfs                            52G  6.2M   52G   1% /dev/shm
tmpfs                            52G   50M   52G   1% /run
tmpfs                            52G     0   52G   0% /sys/fs/cgroup
/dev/mapper/centos_guane03-root  60G  5.4G   55G   9% /
/dev/sda1                      1014M  361M  654M  36% /boot
/dev/mapper/centos_guane03-var   30G  1.4G   29G   5% /var
/dev/mapper/centos_guane03-opt   30G   52M   30G   1% /opt
/dev/mapper/centos_guane03-tmp   40G  293M   40G   1% /tmp
192.168.66.49:/datasets         7.3T  5.2T  2.2T  71% /datasets
192.168.38.10:/courses          1.4T  104M  1.4T   1% /courses
192.168.38.10:/home             3.0T  771G  2.3T  26% /home
192.168.38.10:/covid            1.1T  232G  885G  21% /covid
192.168.38.10:/scratch          2.5T  2.0T  495G  81% /scratch
192.168.38.50:/opt/ohpc/admin   500G  461G   40G  93% /opt/ohpc/admin
192.168.38.50:/usr/local/src    300G  184G  117G  62% /usr/local/src
192.168.38.50:/opt/ohpc/pub     500G  461G   40G  93% /opt/ohpc/pub
192.168.66.43:/girg             5.9T  1.7T  4.2T  29% /girg
[cbernal@guane03 ~]$
```

- dd if=/dev/zero of=/dev/shm/test1.img bs=1G count=1
- dd if=/dev/zero of=$HOME/test1.img bs=1G count=1
- dd if=/dev/zero of=/tmp/test1.img bs=1G count=1

Disk Quota

H - camp

```
Last login: Sun Nov 20 21:13:05 2022 from toctoc.sc3.uis.edu.co

Centro de Supercomputacion y Calculo Cientifico
Universidad Industrial de Santander

==========================================================================
Hostname: guane.uis.edu.co
Distribucion: CentOS Linux release 7.9.2009 (Core)
Kernel: 3.10.0-1160.76.1.el7.x86_64
==========================================================================

System Load:    0.00, 0.01, 0.05        System Uptime:          55 days 21 hours 28 min 18 sec
Memory Usage:   0.0%                    Swap Usage:             1.3%
Local Users:    2                       Whoami:                 cbernal

 Disk Quota Usage:
Filesystem      space           quota           limit
/home           2636M           9216M           10240M
==========================================================================

Para mas informacion de como utilizar la plataforma:    http://wiki.sc3.uis.edu.co/

Para cambiar la clave de acceso asignada, digite el comando:    passwd
```

Ticket in a good way

H - camp

In a HPC SysAmin team, Consultants handle thousands of support requests per year. In order to ensure efficient timely resolution of issues include as much of the following as possible when making a request.

* Error messages
* Job Ids
* Location of relevant files
* Input/output
* Job scripts
* Source code
* Executables
* Output of module list
* Any steps you have tried
* **Steps to reproduce**

...

H - camp


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
sukriya
kop khun krap
go raibh maith agat
dakujem
terima kasih
grazie
arigatō
takk
merci
спасибо
рахмат
Баярлалаа
kiitos
dhanyavad
hvala
mauruuru
köszönöm
enkosi
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
tapadh leat
хвала
asante
manana
obrigada
djiere dieuf
mochchakkeram
chnorakaloutioun
gratias ago
gràcies
sulpáy
akun
dankon
acitu
sobodi
dèkuji
mèsi
didi madoba
sagolun
najis tuke
kam sah hamnida
rahmat
rahmet
tanemirt
ありがとう
dioich
dhanyavadagalu
shukriya
merca
мерси
trugarez
chokrane
murakoze
tenki
mamnun
дякую
감사합니다
xièxie
ευχαριστώ
welalin
tack
barka
mersi
spas
kia ora
mahalo
paldies
nanni
nandri
dankie
gràcie
bayarlalaa