CPA – Computació Paral·lela

Grau en Enginyeria Informàtica

Cluster de pràctiques: KAHAN

J. M. Alonso, P. Alonso, F. Alvarruiz, I. Blanquer,
J. Ibáñez, E. Ramos, J. E. Román

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València

Curs 2024/25

etsinf

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Contingut

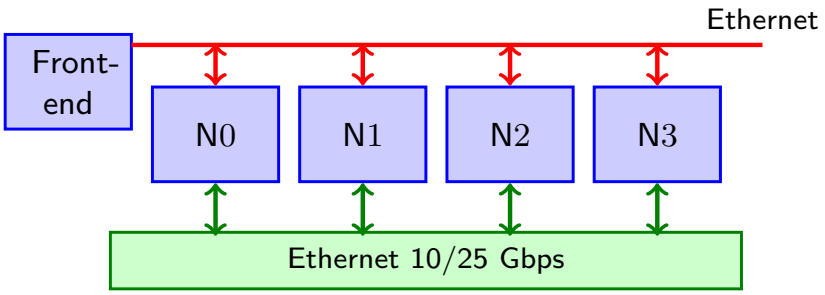
Apartat 1

Cluster de pràctiques: KAHAN

- Cluster de Pràctiques
- Execució de Programes Paral·lels

Cluster de Pràctiques

Configuració hardware: 4 nodes



Cada node:

- 1 processador AMD EPYC 7551P 32 nuclis (64 virtuals)
- 64GB de memòria RAM
- Disc SSD 240GB
- Ethernet 10/25 Gbps 2-port 622FLR -SFP28

Agregat: 4 processadors, 128 nuclis (256 virtuals), 256 GB

Cluster de Pràctiques: Front-End

El node capçalera (*front-end*) permet als usuaris interactuar amb el cluster

Connexió:

```
$ ssh -Y -l login@alumno.upv.es kahan.dsic.upv.es
```

Per a tasques rutinàries (no llançar execucions costoses)

- Edició i compilació dels programes
- Execucions curtes per a comprovar

Comandos útils:

- Fitxers/directoris: `cd`, `pwd`, `ls`, `cp`, `mkdir`, `rm`, `mv`, `scp`, `less`, `cat`, `chmod`, `find`
- Processos: `w`, `kill`, `ps`, `top`
- Editors i altres: `vim`, `emacs`, `pico`, `man`

Cluster de Pràctiques: Xarxa

Gigabit Ethernet

- Xarxa auxiliar, només per a tràfic del S.O. (`ssh`, NFS)

Ethernet 10/25 Gbps

- Xarxa ràpida de baixa latència, ideal per a clusters
- Tarjeta Ethernet 10/25 Gbps 2-port 622FLR -SFP28
- Suporta RDMA, RoCE, iWarp
- Pot funcionar a 25 Gbps

Execució de Programes Paral·lels

OpenMP: executar directament

Sol ser necessari indicar el nombre de fils

```
$ OMP_NUM_THREADS=4 ./prgomp
```

Una altra opció és exportar les variables

```
$ OMP_NUM_THREADS=4; OMP_SCHEDULE=dynamic
$ export OMP_NUM_THREADS OMP_SCHEDULE
$ ./prgomp
```

MPI: usar el comando `mpiexec` (o `mpirun`)

Opcions: seleccionar el host, l'arquitectura

```
$ mpiexec -n 4 prgmpi <args>
$ mpiexec -n 6 -host node1,node2,node5 prgmpi
```


Sistemes de Cues

El sistema de cues (o planificador de treballs o gestor de recursos) és un software que permet usar un cluster de forma compartida entre molts usuaris

- L'usuari pot llançar "treballs" normalment en mode *batch* (no en interactiu) utilitzant un o més nodes
- Un treball (*job*) és una execució particular, amb una sèrie d'atributs (nodes, temps màxim d'execució, etc.)
- Es definixen polítiques de planificació de treballs
- El sistema comptabilitza els recursos utilitzats (hores)
- Objectiu: maximitzar utilització, minimitzar espera

Forma de treballar:

1. Es defineix el treball i es llança a la cua (dóna un identif.)
2. Després d'un temps d'espera, el treball s'executa
3. Al finalitzar es recupera la eixida produïda

Cluster de Pràctiques: Cues (1)

Usarem el sistema de cues SLURM

Exemple de treball `jobopenmp.sh`

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=5:00
#SBATCH --partition=cpa
OMP_NUM_THREADS=3 ./pintegral 1
```

- `--nodes`: nombre de nodes que es demanen
- `--time`: temps d'execució requerit
- `--partition`: partició emprada en el sistema de cues

Per a MPI usar `mpiexec` (no fa falta `-n`)

Cluster de Pràctiques: Cues (2)

Per a llançar:

```
$ sbatch jobopenmp.sh
Submitted batch job 728
```

Al finalitzar es crea en el directori actual un fitxer: `slurm-728.out`

Per a vore l'estat:

```
$ squeue
JOBID PARTITION NAME         USER ST TIME NODES NODELIST
728   cpa       jobopenmp.sh ramos R 0:01  1    kahan01
```

Possibles estats: en cua (`PD`), executant (`R`), acabant (`CD`)

Cancel·lació d'un treball: `scancel`