Paralelización

- Imaginemos que quermos traducir al Inglés *'El ingenioso hidalgo Don Quixote de la Mancha'*, *M. Cervantes, 1605* (approx. 1100 páginas)

Paralelización

- Imaginemos que quermos traducir al Inglés *'El ingenioso hidalgo Don Quixote de la Mancha'*, *M. Cervantes, 1605* (approx. 1100 páginas)
- Todes somos competentes en Inglés

Paralelización

- Imaginemos que quermos traducir al Inglés *'El ingenioso hidalgo Don Quixote de la Mancha'*, *M. Cervantes, 1605* (approx. 1100 páginas)
- Todes somos competentes en Inglés
- La forma más rápida ...

Paralelización

- Imaginemos que quermos traducir al Inglés *'El ingenioso hidalgo Don Quixote de la Mancha', M. Cervantes, 1605* (approx. 1100 páginas)
- Todes somos competentes en Inglés
- La forma más rápida ...
- ... sería repartirnos la misma cantidad de páginas entre nosotres

Paralelización

- Imaginemos que quermos traducir al Inglés *'El ingenioso hidalgo Don Quixote de la Mancha'*, *M. Cervantes, 1605* (approx. 1100 páginas)
- Todes somos competentes en Inglés
- La forma más rápida ...
- ... sería repartirnos la misma cantidad de páginas entre nosotres
- ... cada une trabajaría independientemente con une 'organizador'

Paralelización

- Imaginemos que quermos traducir al Inglés *'El ingenioso hidalgo Don Quixote de la Mancha'*, *M. Cervantes, 1605* (approx. 1100 páginas)
- Todes somos competentes en Inglés
- La forma más rápida ...
- ... sería repartirnos la misma cantidad de páginas entre nosotres
- ... cada une trabajaría independientemente con une 'organizador'
- esto es lo básico del **paralelismo**

Paralelización
Generalities

- El paralelismo informático mimetizar el *'divide y vencerás'*

Paralelización
Generalities

- El paralelismo informático mimetizar el *'divide y vencerás'*
- Se usan simultáneamente múltiples núcleos de cálculo (*cores*), básicamente bajo 2 paradigmas: memoria **distribuida** y memoria **compartida**

Paralelización
Generalities

- El paralelismo informático mimetizar el *'divide y vencerás'*
- Se usan simultáneamente múltiples núcleos de cálculo (*cores*), básicamente bajo 2 paradigmas: memoria **distribuida** y memoria **compartida**
- Técnicamente, los datos a procesar se dividen en distintas partes **chunk** y son enviadas a procesar por distintos núcleos de cálculo. Normalmente un núcleo raíz controlará todo el flujo de datos enviando/reciviendo/coordinando todo el proceso

Paralelización
Generalities

- El paralelismo informático mimetizar el *'divide y vencerás'*
- Se usan simultáneamente múltiples núcleos de cálculo (*cores*), básicamente bajo 2 paradigmas: memoria **distribuida** y memoria **compartida**
- Técnicamente, los datos a procesar se dividen en distintas partes **chunk** y son enviadas a procesar por distintos núcleos de cálculo. Normalmente un núcleo raíz controlará todo el flujo de datos enviando/reciviendo/coordinando todo el proceso
- Las ejecuciones son más rápidas porque la carga de trabajo se **comparte**

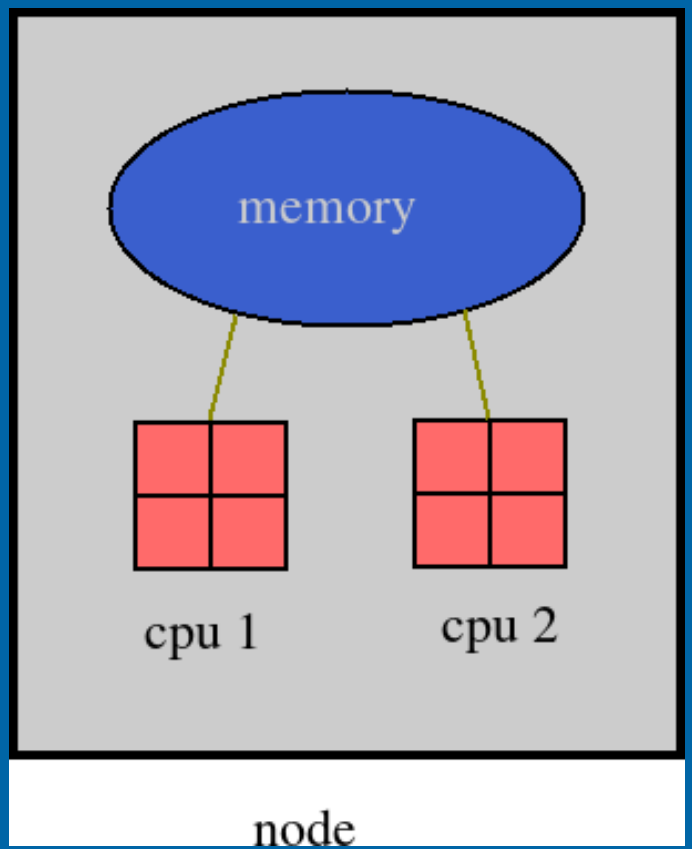
Paralelización
Clúster: la unión hace la fuerza!

Un **clúster** está compuesto por una série de computadoras connecatadas entre sí para compartir trabajo

Nodo: unidad básica

Unidad física básica de computación: unidad de cálculo, memoria

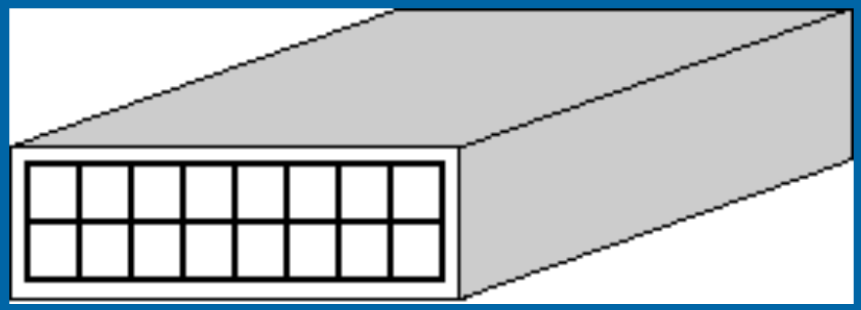
Unidad de cálculo puede estar compuesta de varias *CPU*s y cada una compuesta por múltiples núcleos de cálculo (*core*).



Ejemplo de nodo con 2 cpus y cada uno con 4 cores

Switch: connectividad

Los nodos están conectados a través de un switch que tranfiere la información entre ellos



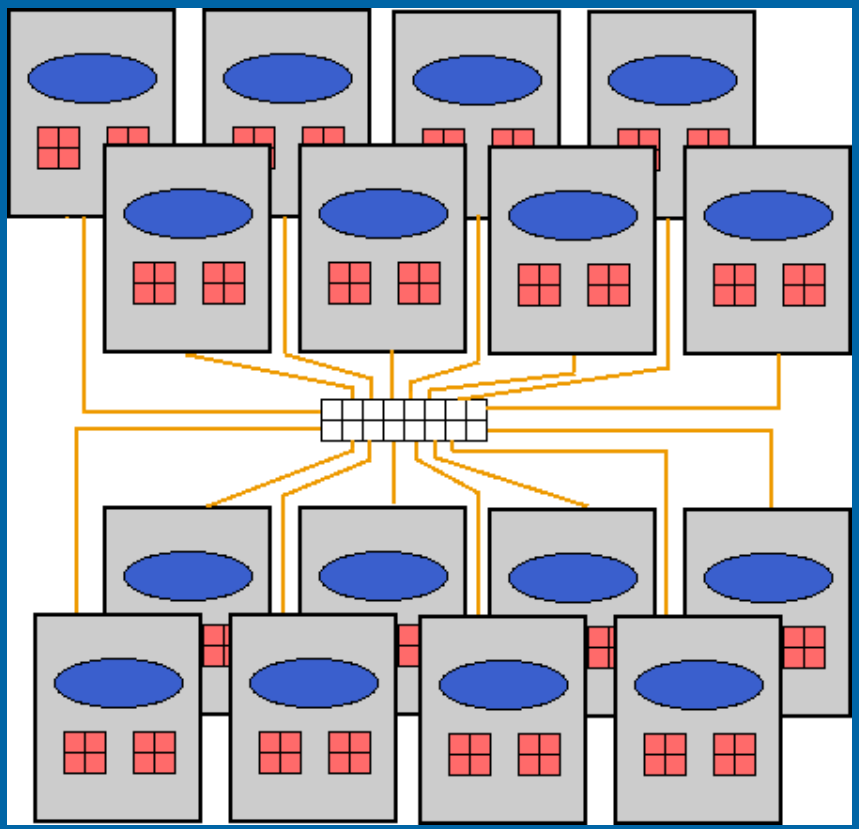
Ejemplo de switch con conexiones para 16 nodos.

Cada nodo está identificado dentro del clúster.

Con el switch podemos especificar el nodo destinatario de la información

Paralelización
Clúster: la unión hace la fuerza!

Clúster: es una série de nodos interconectados mediante un switch con acceso compartido a discos y otros recursos



Ejemplo de clúster de 16 nodos con 32 cpus y 128 cores

Otras configuraciones: vectorial (Cray)

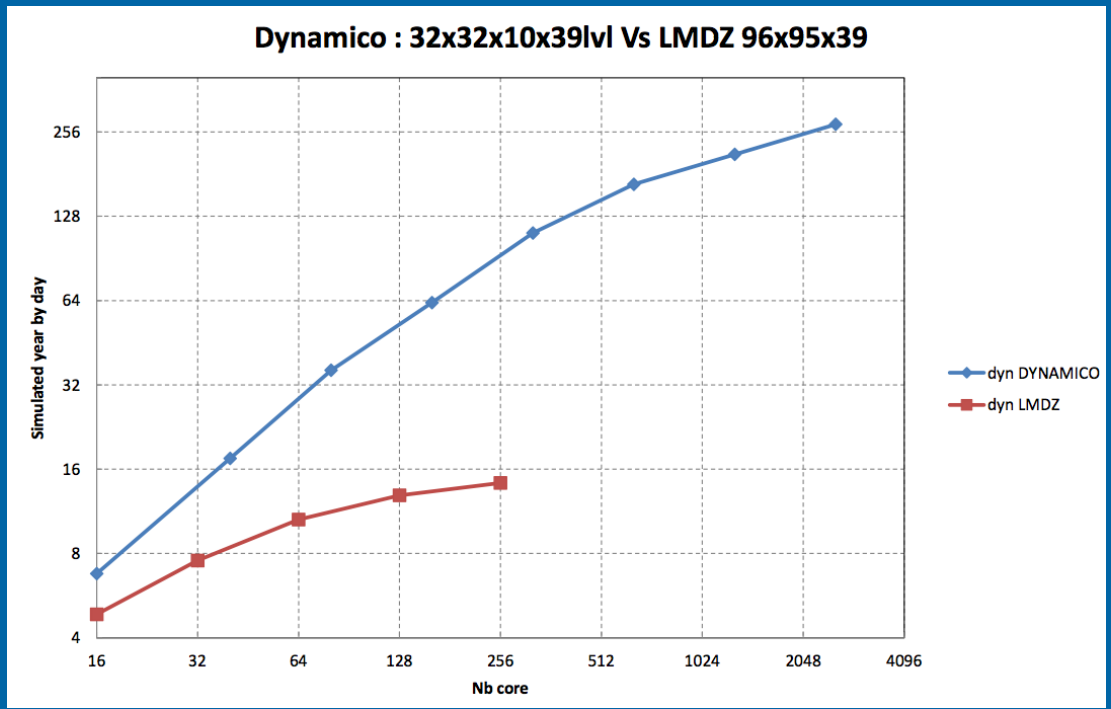
Primero (Jun 2023, Top500): Frontier (EEUU): 1,194.00 petaflops (primero **hexascale**!), 8,699,904 cores

América del Sur: (Brazil, 35°): (Petróleo Brasileiro S.A), 233,856 cores

Paralelización
Clúster: la unión hace la fuerza!

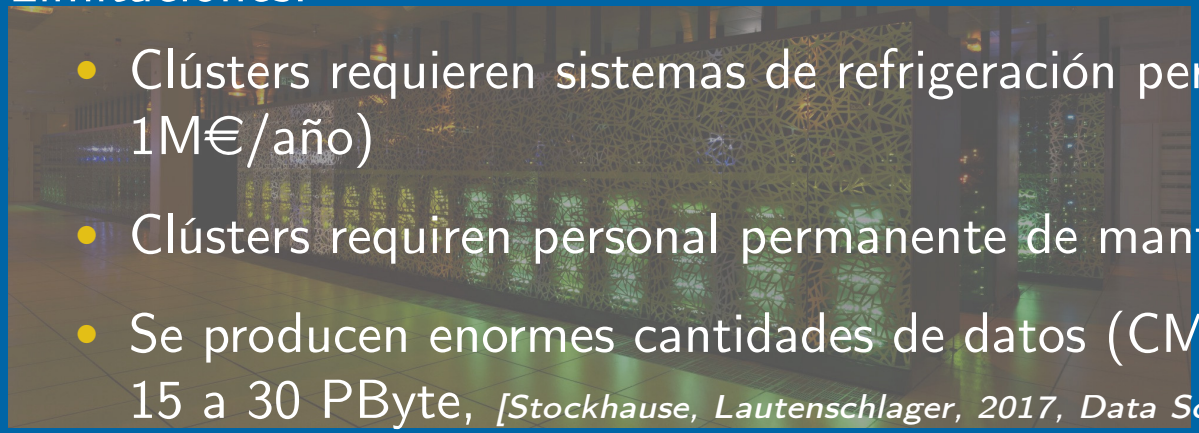
Limitaciones:

- El código tiene que estar preparado para repartir el trabajo de manera eficiente; memoria distribuida (MPI, normalemte entre nodos), compartiendo memoria (openMP, normalmente entre cores dentro del nodo)
- Cuello de botella Switch: a más transmisión, menor eficiencia
- Escalabilidad el código tiene que 'escalar': más cores $\rightarrow$ más rápido



Paralelización
Clúster: la unión hace la fuerza!

Limitaciones:



- Clústers requieren sistemas de refrigeración permanentes (más de 1M€/año)
- Clústers requiren personal permanente de mantenimiento
- Se producen enormes cantidades de datos (CMIP6, estimación de 15 a 30 PByte, *[Stockhause, Lautenschlager, 2017, Data Science Journal, doi: 10.5334/dsj-2017-030]*)