Utilización básica de la librería de paso de mensajes MPI

Multiprocesadores

UNIVERSIDAD DE CANTABRIA
Arquitectura y Tecnología de Computadores
http://www.atc.unican.es

Universidad de Cantabria

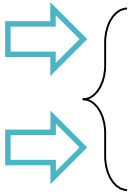
Introducción

Parallel Computing

High Performance Computing today means Parallel computing

- Parallel computing takes a problem and decomposes it into smaller parts, which can be solved concurrently and then combined for the solution to the original, larger problem.
- Parallel programming is a superset of sequential programming that requires the addition of two important capabilities:
  - communication and synchronization

- Instruction-level parallelism (ILP)
- Data-level parallelism (SIMD, GPUs, data replication)
- Thread-level parallelism (Multicores, OpenMp)
- Process-level parallelism (Distributed Applications, MPI)
- Request-level parallelism (DataBase Servers)

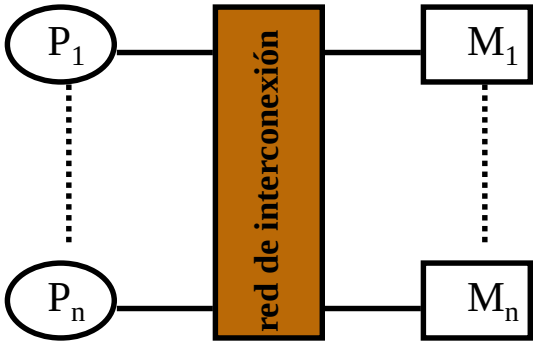


UC UNIVERSIDAD DE CANTABRIA — CANTABRIA CAMPUS INTERNACIONAL

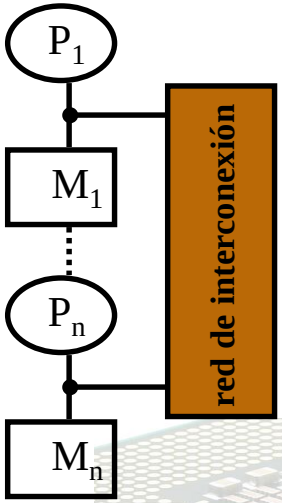
12

Introducción

- Habéis visto la forma de programar sistemas de memoria compartida:
  - Con threads explícitos de POSIX o
  - Implícitos con openMP.



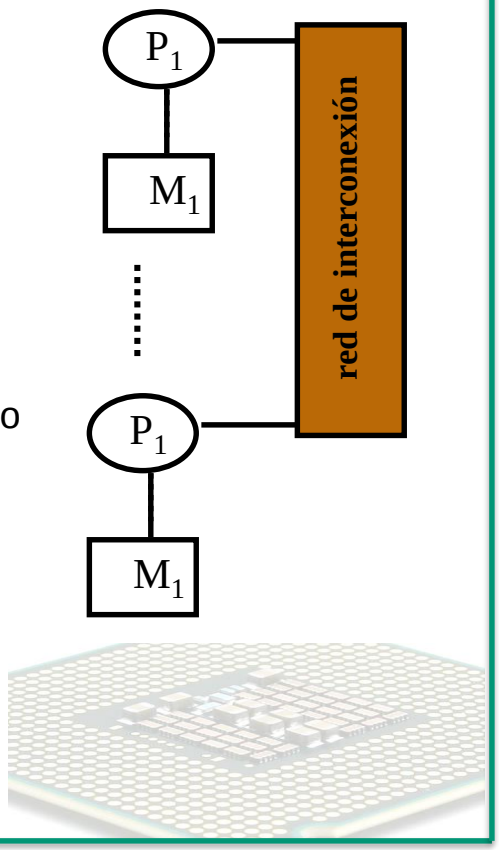
UMA
(Uniform Memory Access)
SMP



NUMA
(Non Uniform Memory Access)

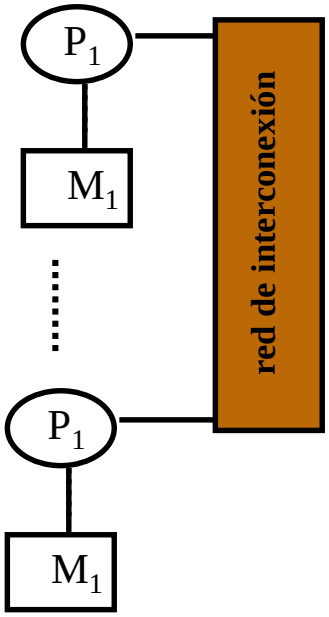
Introducción

- Ahora toca programar sistemas de memoria distribuida (no excluyente).
- Modelo de paso de mensajes.
  - Cada procesador tiene un índice.
  - Su propio espacio de direcciones. No existe memoria global. No DSM.
  - Hay una red de interconexión que los comunica.
  - Las instrucciones responden a dos **modelos**:
    - MPMD. Cada procesador ejecuta su código.
    - SPMD. Todos los procesadores ejecutan el mismo código pero éste puede estar parametrizado con el índice.
  - Y a dos **tipos**:
    - Locales sobre sus propios datos.
    - Comunicación. Intercambio de datos.
      - Pueden ser de uno a uno, P2P.
      - Involucrar varios procesos/procesadores.



Introducción

- Modelo de Comunicación:
  - Un tiempo de latencia L.
  - Un tiempo de comunicación para transferir un byte de uno a otro que depende del ancho de banda BW (throughput o goodput).
    - Así T = L + 1/BW x Tamaño del paquete.
  - Si el enlace no es directo:
    - T = d (L + 1/BW x N)
  - Hay otros modelos más complejos como LogP/LogGP/LoOgGP que tienen en cuenta el tamaño del mensaje.



Índice General

1. Introducción e instalación.
2. Comunicación punto a punto bloqueante.
3. Comunicación punto a punto no bloqueante.
4. Llamadas colectivas.
   - Barrier, Broadcast, gather, scatter.
5. Llamadas colectivas.
   - Todos con todos, reduction.
6. Tipos de datos / Comunicadores.

1. Introducción. Bibliografía

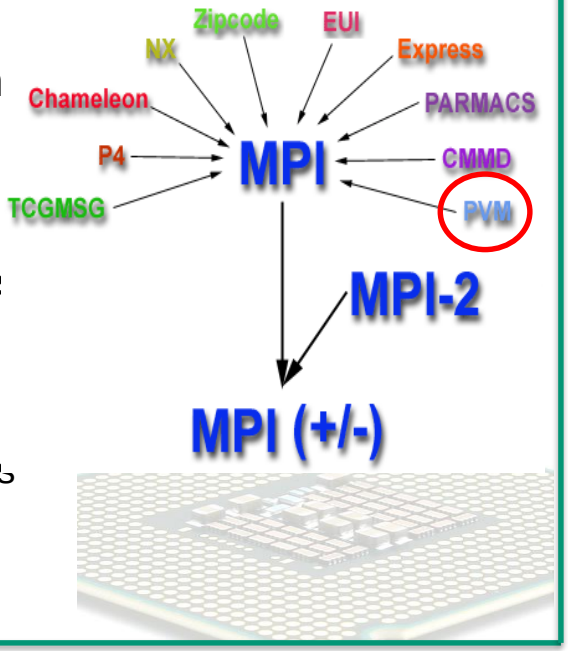
- Libros:
  - Parallel Programming with MPI. Peter S. Pacheco. Morgan Kaufmann.
  - USING MPI. W. Gropp, E. Lusk, A. Skjellum. The MIT Press.
  - MPI the complete reference 2ª Vol. 1 Core. Snir y otros. The MIT Press.
  - MPI the complete reference 2ª Vol. 2 Extensions. Gropp y otros. The MIT Press.
  - USING MPI 2. W. Gropp, E. Lusk, R. Thakur. The MIT Press.
- Web:
  - http://www.open-mpi.org/ (una implementación)
  - https://www.mpi-forum.org/ (comité de estándares)
  - https://www.open-mpi.org/doc/v4.0// (manual de llamadas).
  - https://www.mpich.org/ (otra implementación)

1. Introducción

- Ventajas.
  - Buena relación rendimiento/coste.
  - Escalabilidad.
- Inconvenientes.
  - Dificultad de programación. Pensar en paralelo.
  - Puede ser ineficaz por las comunicaciones.
  - No sirve para paralelismo fino, pero se pueden combinar.
  - El factor crítico son las comunicaciones en relación a la velocidad de cálculo.

1. Introducción

- Definición de MPI: Interfaz estándar para aplicaciones paralelas de paso de mensajes.
  - Normalmente para arquitecturas distribuidas.
  - Interfaz de programación independiente del lenguaje (LIS).
- Un poco de historia:
  - Fue creada por William Gropp y Ewing Lusk entre otros.
  - Proviene de otros proyectos para la programación paralela, el más conocido **PVM**.
  - Soportado por la fundación MPI Forum (http://www.mpi-forum.org):
    - Organización formada por Universidades y Empresas
    - Entorno de programación **único** que garantice la portabilidad.
    - Ofrecer implementaciones de dominio **público**.
    - Convencer a empresas para hacer implementaciones de calidad.
    - Se han realizado varias **versiones** e **implementaciones**.



1. Introducción. Versiones

- Versiones:
  - MPI 1.1
    - Realizada para ANSI C y Fortran 77 en 1995.
  - MPI 2.0
    - Realizada para C++ (C) y Fortran 90 en 1998.
    - Es un superconjunto de la versión 1 teniendo presente que puede existir memoria compartida (cumple también con la versión 1).
      - Creación dinámica de tareas.
      - Comunicación one-side.
      - E/S paralela.
  - MPI 1.3
    - Versión final de lo que se llama MPI 1 cuya última revisión es de julio de 2008 y tiene 128 llamadas.
  - MPI 2.2
    - Versión final de lo que se llama MPI 2 cuya última revisión es de septiembre de 2009 (incluye también Fortran 95) y tiene sobre 200 llamadas.
  - MPI Forum ha sacado en 2012 el estándar de MPI 3 y en 2015 el 3.1.
    - https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf
  - Desde junio de 2021 ya está lista la versión MPI 4.

1. Introducción. Implementaciones

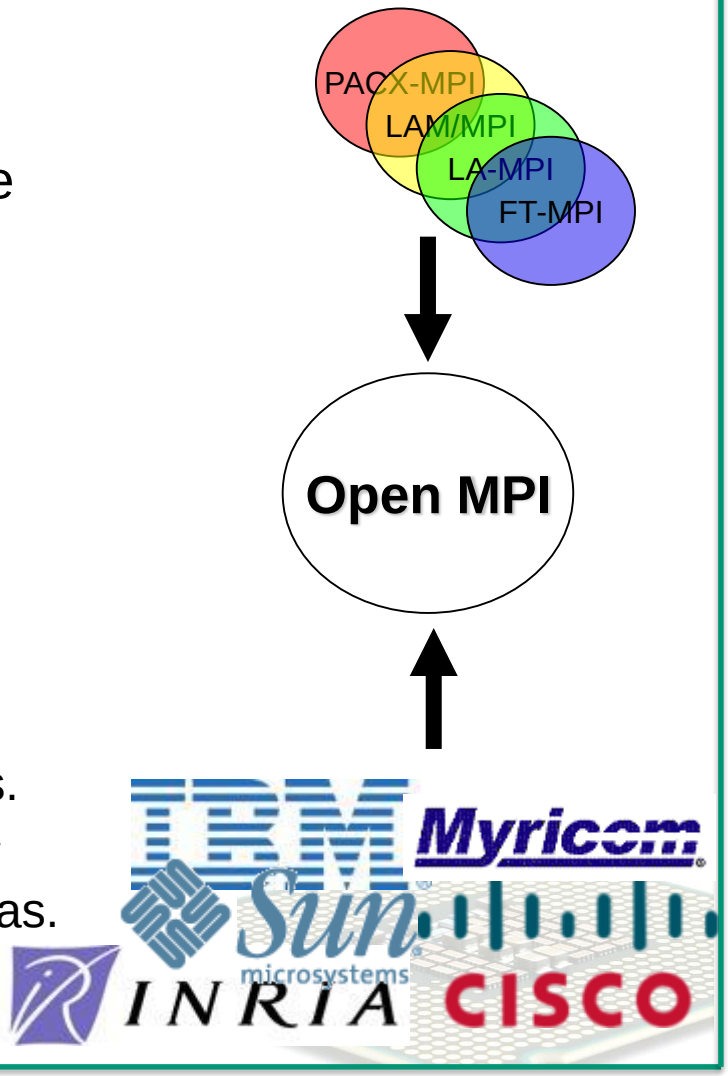
- Una implementación tendrá al menos:
  - Una biblioteca de funciones para el lenguaje C y FORTRAN.
  - Ficheros de cabecera para C y FORTRAN mpi.h y mpif.h.
  - Se pueden soportar otros lenguajes como C++.
  - Comandos para la compilación: `mpicc` y `mpif77`.
  - Comando de ejecución para aplicaciones paralelas: `mpirun`.
- Implementaciones:
  - **MPICH**. Es la más antigua.
    - Contiene todos los estándares.
    - Ha sido probada en Linux, Windows y Mac OS.
  - Intel MPI, HP MPI, SGI, Cray, Microsoft MPI, BIP, … Muchos fabricantes de computadores y redes especiales tienen su propia implementación.

1. Introducción: Implementaciones

- Implementaciones: Cerrada desde 9/15.
  - **LAM/MPI**. Es junto con MPICH la más usada.
    - Implementación de MPI pensada originalmente para correr sobre redes de computadores, da soporte a muchas arquitecturas y redes avanzadas.
    - Contiene los estándares 1 y 2:
      - Creación de procesos.
      - Comunicación one-sided.
      - E/S de MPI.
      - Soporte a threads.
      - C++
    - Tiene complementos con otras herramientas como checkpoint o colas de trabajos.
    - Ha dado lugar a Open MPI. De hecho **recomiendan** irse a openMPI

1. Introducción: Implementaciones

- **Open MPI**.
  - Versión **abierta** desde cero de Universidades y fabricantes.
  - Da soporte a todo el estándar MPI.
    - Concurrencia de threads segura.
    - Creación dinámica de tareas.
    - Tolerancia a fallos de red.
    - Se puede instrumentalizar en ejecución.
    - Soporte a redes heterogéneas.
    - Soporte a muchos SO de 32 y 64bits y planificadores de tareas.
    - Portable y tuneable…



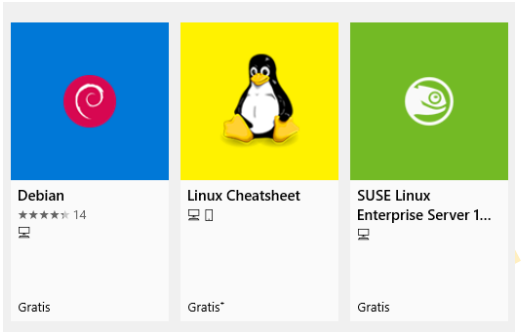
1. Introducción. Implementaciones

- **Open MPI**.
  - Está desarrollado en Linux, OS X, Windows y Solaris (32 y 64).
  - Básicamente corre en cualquier **sistema** POSIX.
  - Están soportados todas las **redes** que tengan comunicación punto a punto: TCP / ethernet, Shared memory, Loopback (send-to-self), Myrinet / GM, Myrinet / MX, Infiniband / OpenIB, Infiniband / mVAPI, Portals.
  - Corre sobre un **middelware** llamado ORTE que da soporte a Open MPI y a su vez está construido sobre los RTE mas populares como:
    - Sun Grid Engine, PBS Pro, Torque, and Open PBS (the TM system), LoadLeveler, LSF, POE, **rsh / ssh**, SLURM, XGrid, y Yod (Red Storm).
  - Está siendo sustituido por PMIx.

1. Instalación

- Para hacer las prácticas este curso peculiar:
  - Vuestro portátil.
    - Doble arranque.
    - Máquina virtual.
    - Subsistema de Linux en Windows. WSL 2

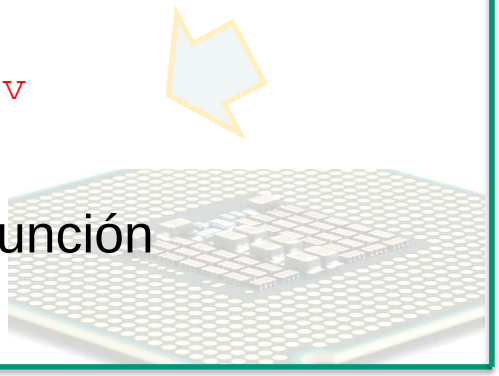
    https://docs.microsoft.com/es-es/windows/wsl/about
    https://docs.microsoft.com/es-es/windows/wsl/install-win10



  - Ordenadores del laboratorio (todos).
  - Conectándonos a Altamira.

1. Instalación

- Práctica: Instalación y prueba de una implementación.
  - MPICH http://www.mcs.anl.gov/research/projects/mpi/mpich1/download.html
    http://www.mcs.anl.gov/research/projects/mpich2/downloads/index.php?s=downloads
    - Windows (nosotros NO) y UNIX.
  - LAM http://www.lam-mpi.org/7.1/download.php Cerrado
    - Linux en tgz y rpm.
  - OpenMPI http://www.open-mpi.org/software/ompi/v1.4/
    - Linux en tgz y rpm.
    - Una macro autogen lo instala, necesario autoconf, automake y libtool.
    - También se puede hacer
      - `$ ./configure –prefix=/usr/local (en /usr/local)`
      - `# make all install`
  - Ubuntu: Intenta ejecutar mpicc, te dirá el paquete que te tienes que bajar:
    - `Antes: sudo apt-get install openmpi-dev`
    - `Ahora: sudo apt update`
      `sudo apt install openmpi-bin`
      `sudo apt install libopenmpi-dev`
    - O desde Synaptic (ver posterior).
- Copia el programa C de ejemplo y ejecútalo.
- Realiza un programa “hello world” usando la función MPI_Get_processor_name.



1. Instalación

openmpi-bin: Programa de ejecución de códigos paralelos (mpirun).
libopenmpi-dbg: Generador de información de depuración para MPI.
libopenmpi-dev: Necesario para el desarrollo de programas basados en MPI (mpicc, etc).

```
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $ ls -l
total 28
-rwxr-xr-x 1 rafa rafa 13311 abr 17 12:52 a.out
-rwxr-xr-x 1 rafa rafa   521 feb 27 17:22 hello.c
-rwxr-xr-x 1 rafa rafa   410 feb 27 17:22 inicio.c
-rwxr-xr-x 1 rafa rafa  1635 feb 27 17:22 saludos.c
rafa@rafa-64 ~/mpi/sesion1 $ ll
total 28
-rwxr-xr-x 1 rafa rafa 13311 abr 17 12:52 a.out
-rwxr-xr-x 1 rafa rafa   521 feb 27 17:22 hello.c
-rwxr-xr-x 1 rafa rafa   410 feb 27 17:22 inicio.c
-rwxr-xr-x 1 rafa rafa  1635 feb 27 17:22 saludos.c
rafa@rafa-64 ~/mpi/sesion1 $ ll
total 28
-rwxr-xr-x 1 rafa rafa 13311 abr 17 12:52 a.out
-rwxr-xr-x 1 rafa rafa   521 feb 27 17:22 hello.c
-rwxr-xr-x 1 rafa rafa   410 feb 27 17:22 inicio.c
-rwxr-xr-x 1 rafa rafa  1635 feb 27 17:22 saludos.c
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $ ll
total 28
-rwxr-xr-x 1 rafa rafa 13311 abr 17 12:52 a.out
-rwxr-xr-x 1 rafa rafa   521 feb 27 17:22 hello.c
-rwxr-xr-x 1 rafa rafa   410 feb 27 17:22 inicio.c
-rwxr-xr-x 1 rafa rafa  1635 feb 27 17:22 saludos.c
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $
rafa@rafa-64 ~/mpi/sesion1 $ ll
total 28
-rwxr-xr-x 1 rafa rafa 13311 abr 17 12:52 a.out
-rwxr-xr-x 1 rafa rafa   521 feb 27 17:22 hello.c
-rwxr-xr-x 1 rafa rafa   410 feb 27 17:22 inicio.c
-rwxr-xr-x 1 rafa rafa  1635 feb 27 17:22 saludos.c
rafa@rafa-64 ~/mpi/sesion1 $ llll
llll: command not found
```