Message Passing Interface
- OpenMPI

November-2011

Rasmus Elmholt

mercantec

MPI

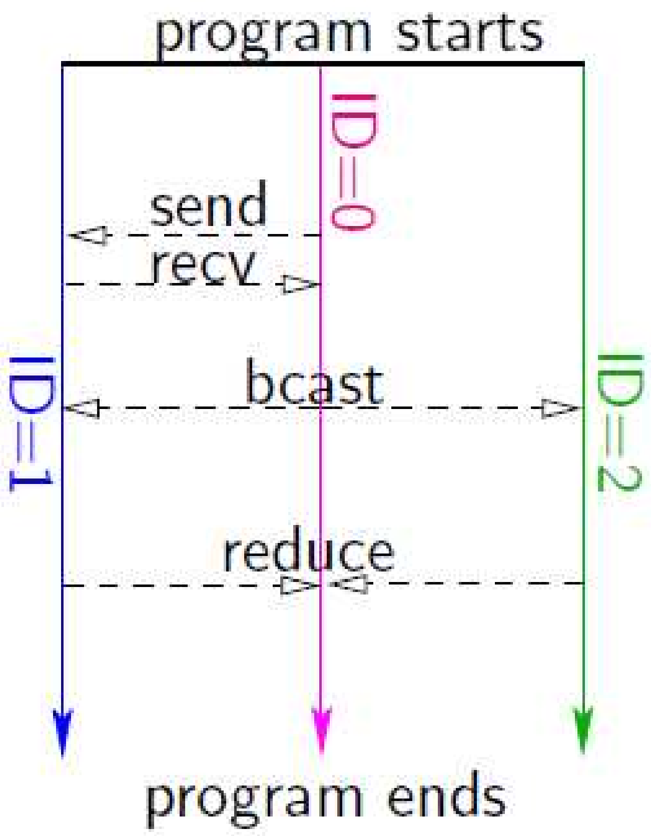
- Et API der tillader programmer at snakke med hinanden.
- Er ikke en officiel standart fra en af de store(ISO, IEEE osv.)
- Er blevet en de facto standart på alle high performance clusters(http://www.mpi-forum.org)
- Før MPI havde hver producent deres metode at lave Message Parsing.
- Er sprog og infrastruktur uafhængig

Parallel Programmerings-stile

- Message passing interface (MPI)
  - Application consists of several processes
  - Communicates by passing data to one another (send/receive) (broadcast/gather)
  - Synchronization is still required of the programmer, however, locking is not since nothing is (directly) shared.
  - Common approach is domain decomposition where each task is assigned a subset and communicates it edge values to neighbouring subdomains

MPI programmeringsmodel

- Alle processer kører det samme program.
  - Hver process får unikt ID (kaldet rank).
  - Master processen har rank 0 (ID=0).
- Betingede operationer gør at programmerne afvikler forskellige dele af koden (baseret på rank).
- Alle processer terminerer ved afslutningen af programmet.

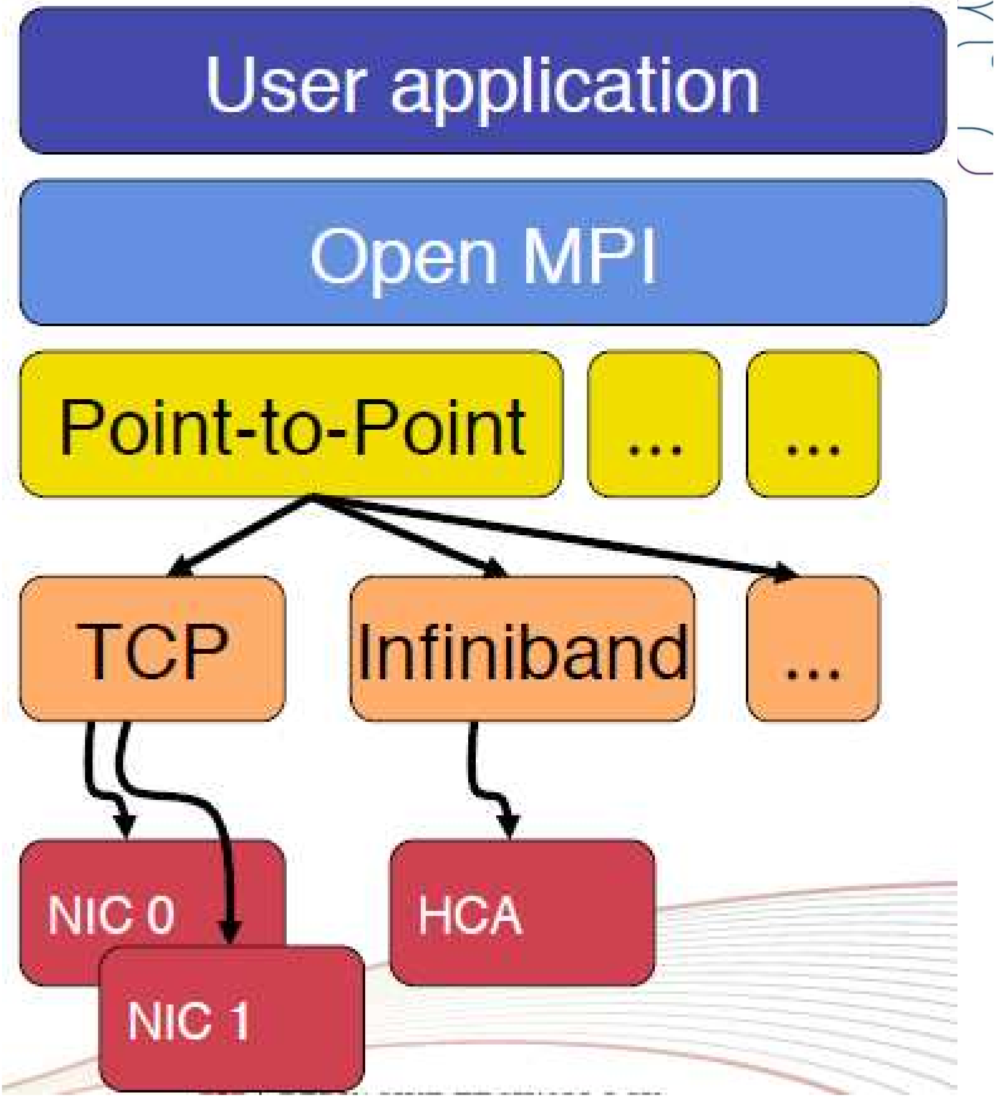


Open MPI er baseret på plug-ins

- Mange forskellige typer plug-in typer
  - Back-end network (ethernet, infiniband, myrinet, …)
  - Resource manager support (ssh, rsh, torque/PBS …)
  - Operating system support (Linux, Windows, Mac OS)
- Alt kan loades ved runtime.
  - Valg af netværk er et runtime valg.

Modular Component Architecture (MCA)

- Framework: (P-to-P)
  - API targeted to a specific task
- Component: (TCP/Inf..)
  - An implementation of a framework’s API
- Module: (NIC0, HCA1)
  - An instance of a component



Kompilering af MPI programmer på Linux

Open MPI team anbefaler at anvende wrapper-compiler, mpicc, istedet for native gcc, det vil sige undgå:

```
gcc -g -o hello hello.c
```

anvend istedet:

```
mpicc -g -o hello hello.c
```

```
# Nedenstående viser nødvendige compile-flag:
mpicc --showme:compile
```

```
# Nedenstående viser nødvendige linker-flag:
mpicc --showme:link
```

Running MPI on Linux

- To distribute applications call "mpirun", "mpiexec" or "orterun" - they are the same binary (in Open MPI)

- **General form**
- mpirun [-np X] ditprogram arg1 …

- If **not using a scheduler**, need a hostfile
- mpirun [-np X] --hostfile hostfile ditprogram arg1 …

- **If using a scheduler (torque, PBS, …)**, no need for hostfile nor -np
- mpirun ditprogram arg1 …

MPI License

- Open MPI is licensed under the BSD
- All contributed code must be compatible with BSD
- GPL is not compatible with BSD
- One, top-level LICENSE file
- Always include all relevant notices when importing external source code

MPI kommunikation

- Specificer BTL: Byte Transfer Layer
  - Framework for MPI point-to-point communications
  - mpirun --mca btl tcp,self -np 4 ring_c
  - angiver hvilke komponenter der skal anvendes / loades.
  - tcp: TCP sockets
  - self: Loopback (send-to-self)

MPI kommunikation

- Angiv openIB BTL:
  - mpirun --mca btl openib,self -np 4 ring_c
  - openib: OpenFabrics verbs (InfiniBand)
  - self: Loopback (send-to-self)

- Angiv: Specify sm+openib BTLs
  - mpirun --mca btl sm,openib,self -np 4 ring_c
  - sm: Shared memory (on-host communication)
  - openib: OpenFabrics verbs (InfiniBand)
  - self: Loopback (send-to-self)

OpenMPI på Ubuntu

- Aptitude install libopenmpi-dev openmpi-bin openmpi-doc
- Krav: Passwordløs SSH skal virke

- Skriv og compiler et program
  - http://mars.tekkom.dk/mediawiki/index.php/Weekend_Projekt_-_Test_Cluster#Hello_World_eksempel
- Kør det på alle maskiner
  - http://mars.tekkom.dk/mediawiki/index.php/Weekend_Projekt_-_Test_Cluster#Installer_OpenMPI

Litteratur

- https://computing.llnl.gov/tutorials/mpi/