Troubleshooting in ambiente HPC

Tutorial Days 2024 - HPC

25 - 27 Novembre 2024

INFN – Sezione di Pisa

Antonino Formuso – INFN Pisa

Sommario

- Anatomia di un cluster HPC: OpenMPI e Infiniband
- Vita, morte e miracoli di un job parallelo
- Casi di studio:
  - Analisi job CFD con Starccm
  - Analisi job Ansys Mechanical
- Conclusioni

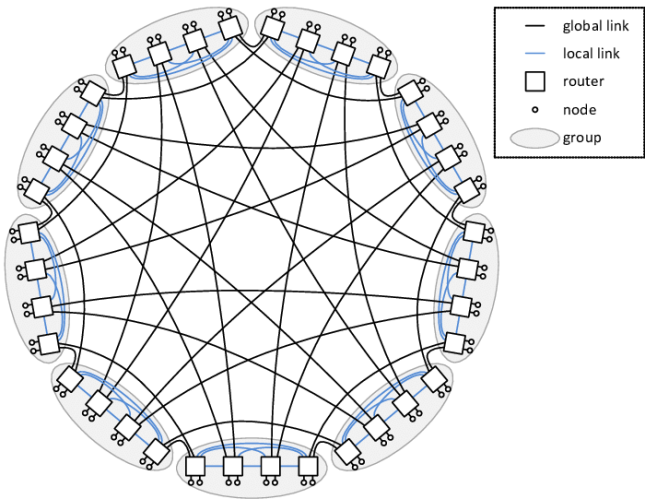
Anatomia di un cluster HPC (1)

Nodi:

- Calcolo: sono i "**muscoli**" del cluster che fanno i calcoli effettivi. Con o senza GPU.
- Login: punto di accesso per gli utenti, consentendo la gestione dei lavori e l'interazione con il sistema.
- I/O: Gestiscono l'input/output dei dati, spesso utilizzando sistemi di file distribuiti per una migliore scalabilità.

Interconnessione:

- Rete ad alta velocità: il “**sistema cardiocircolatorio**” del cluster. Interconnessione nodi con tecnologia InfiniBand.
- Topologia di rete: determina come i nodi sono collegati tra loro, influenzando le prestazioni della comunicazione.
- Topologie Cluster HPC: fat-tree, dragonfly (Leonardo), 6D Mesh/Torus (Fugaku Supercomputer)
- **La distanza (latenza) non è trascurabile !**



Anatomia di un cluster HPC (2)

Software

- Ambiente di sistema: il sistema operativo del nodo, tipicamente Linux CentOS o SUSE.
- Scheduler: il “**sistema nervoso**”. Gestisce l'allocazione delle risorse (CPU, memoria, ecc.) ai vari lavori in esecuzione. Esempi comuni includono SLURM, PBS Pro e LSF.
- Ambiente utente: ospita il run delle applicazioni HPC.
- Varietà di librerie utente : module load oppure AFS ?

Archiviazione

- Locale: usa lo spazio disco dei nodi per lo storage temporaneo dei dati.
- Remoto: tipicamente una SAN che può fornire una capacità di archiviazione scalabile.

Funzionamento di un cluster HPC

- Sottomissione del job: L'utente invia un job allo scheduler, specificando le risorse richieste (CPU, memoria, tempo di esecuzione).
- Pianificazione: lo scheduler assegna il lavoro a uno o più nodi disponibili, tenendo conto delle risorse richieste e delle politiche di pianificazione.
- Esecuzione: il job viene eseguito in parallelo sui nodi assegnati.
- Completamento: al termine del lavoro, i risultati vengono raccolti e resi disponibili all'utente.

Come si scrive codice parallelo ?

- MPI → Protocollo standard di comunicazione tra processi paralleli e distribuiti, nato nel ‘94 (primo draft presentato a SC93)
- **Diverse implementazioni di MPI → Industrie e università (OpenMPI, Intel, Cray, MPICH, MVAPICH, ecc.)**
- Un singolo programma in esecuzione su tutti i nodi → Separazione dei ruoli

OpenMPI (1)

```
if ( I am processor A ) then
    // Coordinator
 else if ( I am processor B ) then
    // Worker
end
```

1) mpicc mpi.c -o mpi

2) bsub < script-lancio.bash

```
mpirun -n 128 --hostfile zefiro.mpirun.hostfile mpi
```

```c
#include "mpi.h"
#include <stdio.h>
#include <string.h>

int main (int argc, char *argv[])
{
    int i, rank, size, namelen;
    char name[MPI_MAX_PROCESSOR_NAME];
    MPI_Status stat;

    MPI_Init (&argc, &argv);

    MPI_Comm_size (MPI_COMM_WORLD, &size);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Get_processor_name (name, &namelen);
 /*
 * MPI_SEND(variabile, quanti, tipo, destinazione, tag, comunicatore, ierr)
 * MPI_RECV(variabile, quanti, tipo, sorgente, tag, comunicatore, stato, ierr)
 *
 * -destinazione è l'intero che caratterizza il processo che deve ricevere il messaggio
 *
 * -sorgente è l'intero che caratterizza il processo che ha spedito il messaggio;
 * può essere usata la variabile MPI_ANY_SOURCE che indica al processo ricevente
 * che può accettare il messaggio da qualunque sorgente
 *
 * MPI_Send(
 *       data         =  &number,
 *       count        =  1,
 *       datatype     =  MPI_INT,
 *       destination  =  1,
 *       tag          =  0,
 *       communicator =  MPI_COMM_WORLD);
 *
 * MPI_Recv(
 *       data         =  &number,
 *       count        =  1,
 *       datatype     =  MPI_INT,
 *       source       =  0,
 *       tag          =  0,
 *       communicator =  MPI_COMM_WORLD,
 *       status       =  MPI_STATUS_IGNORE);
 *
 */
    if (rank == 0) {

        printf ("Hello world from master process %d of %d running on %s\n", rank, size, name);

        for (i = 1; i < size; i++) {
            MPI_Recv (&rank, 1, MPI_INT, i, 1, MPI_COMM_WORLD, &stat);
            MPI_Recv (&size, 1, MPI_INT, i, 1, MPI_COMM_WORLD, &stat);
            MPI_Recv (&namelen, 1, MPI_INT, i, 1, MPI_COMM_WORLD, &stat);
            MPI_Recv (name, namelen + 1, MPI_CHAR, i, 1, MPI_COMM_WORLD, &stat);
            printf (" Hello world from rank %d of %d running on %s\n", rank, size, name);
        }

    } else {

        MPI_Send (&rank, 1, MPI_INT, 0, 1, MPI_COMM_WORLD);
        MPI_Send (&size, 1, MPI_INT, 0, 1, MPI_COMM_WORLD);
        MPI_Send (&namelen, 1, MPI_INT, 0, 1, MPI_COMM_WORLD);
        MPI_Send (name, namelen + 1, MPI_CHAR, 0, 1, MPI_COMM_WORLD);

    }

    MPI_Finalize ();

    return (0);
}
```

bsub < script-lancio.bash

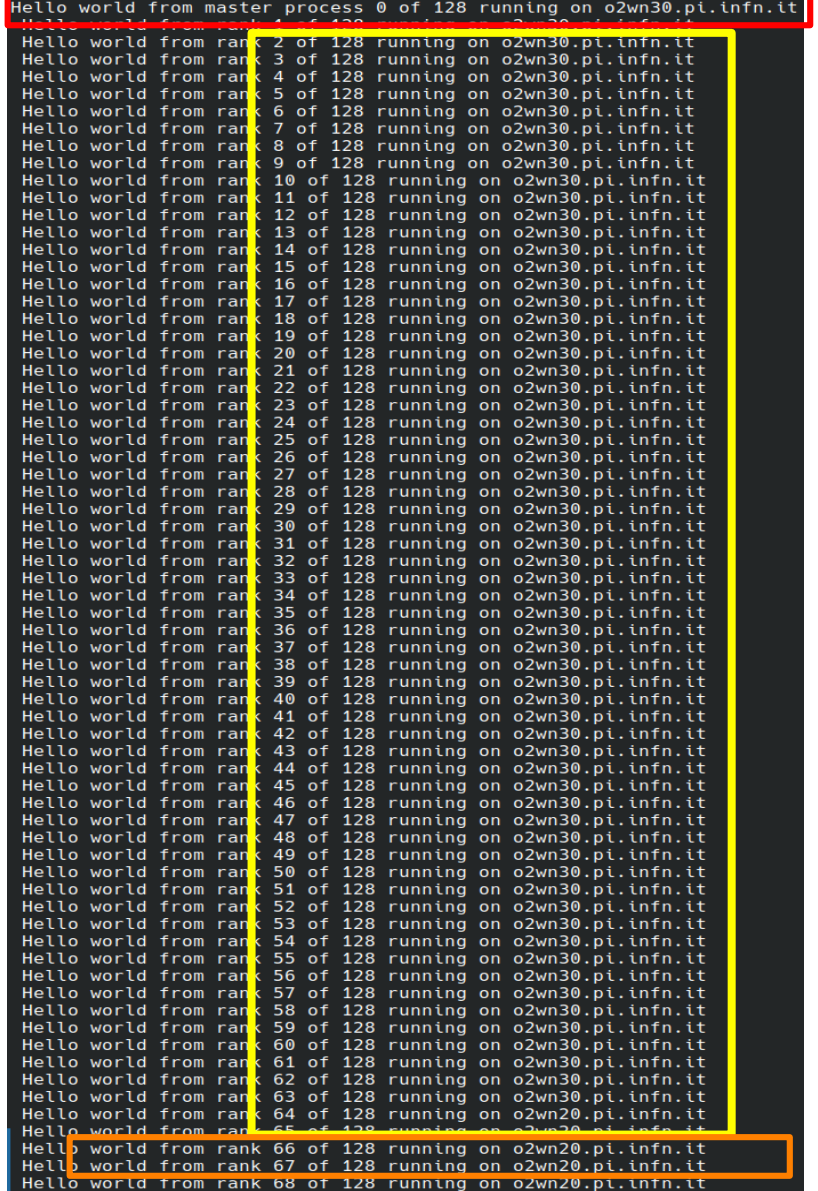
```
#!/bin/bash
#BSUB -q parallel
#BSUB -J TonyStark
#BSUB -n 128
##BSUB -R "span[ptile=32]"
#BSUB -e $PWD/zefiro.err
#BSUB -o $PWD/zefiro.out

cd $HOME
echo "LSB_MCPU_HOSTS:" > zefiro.hosts-list-info
echo $LSB_MCPU_HOSTS >> zefiro.hosts-list-info

echo "LSB_HOSTS:" >> zefiro.hosts-list-info
echo $LSB_HOSTS >> zefiro.hosts-list-info


nodes=(`echo $LSB_MCPU_HOSTS | awk '{s="";for (i=1;i<=NF;i+=2) {s=s?s FS $i:$i} print s}'`)
echo "nodes:" >> zefiro.hosts-list-info
echo "${nodes[*]}" >> zefiro.hosts-list-info

cores=(`echo $LSB_MCPU_HOSTS | awk '{s="";for (i=2;i<=NF;i+=2) {s=s?s FS $i:$i} print s}'`)
echo "cores:" >> zefiro.hosts-list-info
echo "${cores[*]}" >> zefiro.hosts-list-info

echo "for:" >> zefiro.hosts-list-info
j=0; for nodo in ${nodes[*]}; do echo "$nodo slots=${cores[$j]}"; j=$[$j+1]; done > zefiro.mpirun.hostfile

echo $LSB_MCPU_HOSTS | tr " " ":" > $PWD/machinesfile

mpirun -n 128 --hostfile zefiro.mpirun.hostfile mpi
```

bsub < script-lancio.bash

```
[formuso@gridui1 ~]$ cat zefiro.hosts-list-info
LSB_MCPU_HOSTS:
o2wn30 64 o2wn20 64
LSB_HOSTS:
o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn
 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn30 o2wn20 o2w
0 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2
20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20 o2wn20
nodes:
o2wn30 o2wn20
cores:
64 64
for:
[formuso@gridui1 ~]$ cat zefiro.mpirun.hostfile
o2wn30 slots=64
o2wn20 slots=64
```

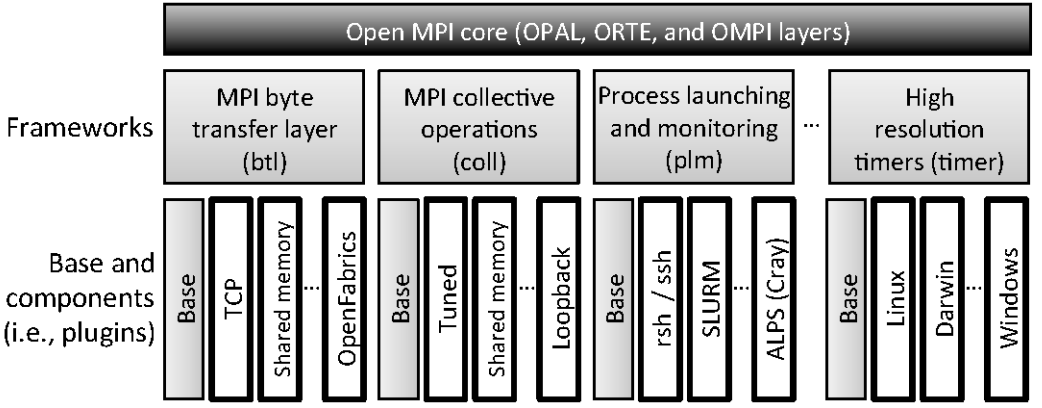
- Interazione tra OpenMPI e lo scheduler (LSF);
- Ultime release di OpenMPI recuperano info dei job in automatico;
- I software commerciali usano implementazioni interne di mpi e necessitano di un file host machine o di conoscere lo scheduler
- ...

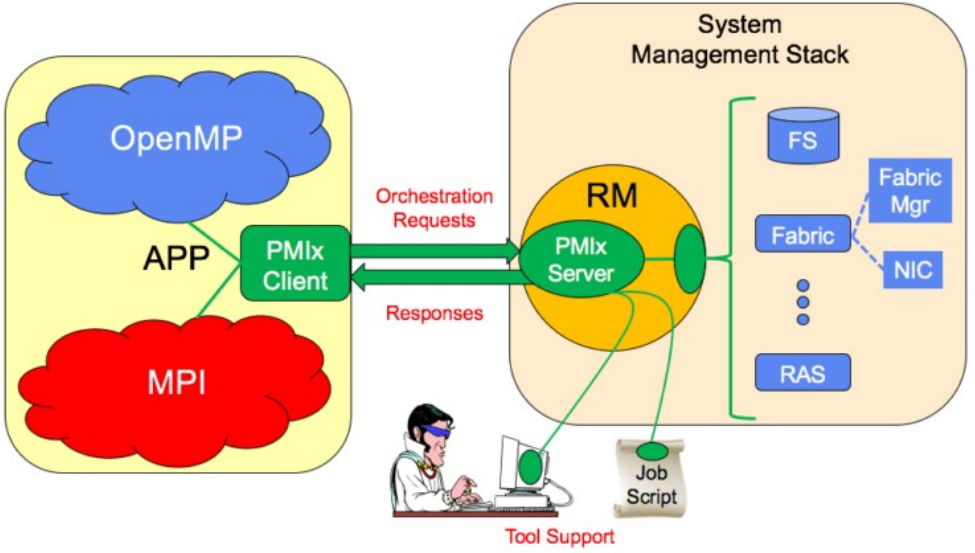
Output:

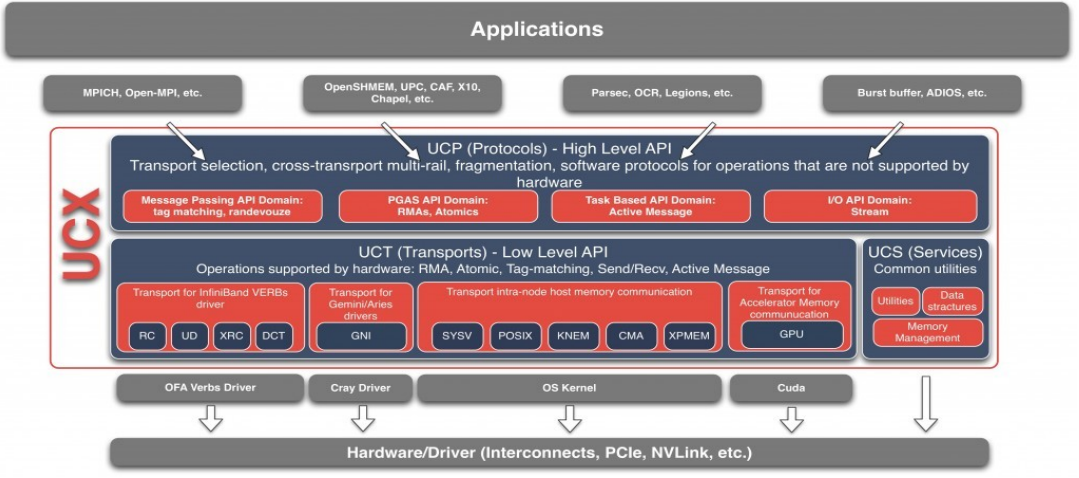

Hello world from master process 0 of 128 running on o2wn30.pi.infn.it
Hello world from rank 2 of 128 running on o2wn30.pi.infn.it
Hello world from rank 3 of 128 running on o2wn30.pi.infn.it
Hello world from rank 4 of 128 running on o2wn30.pi.infn.it
Hello world from rank 5 of 128 running on o2wn30.pi.infn.it
Hello world from rank 6 of 128 running on o2wn30.pi.infn.it
Hello world from rank 7 of 128 running on o2wn30.pi.infn.it
Hello world from rank 8 of 128 running on o2wn30.pi.infn.it
Hello world from rank 9 of 128 running on o2wn30.pi.infn.it
Hello world from rank 10 of 128 running on o2wn30.pi.infn.it
Hello world from rank 11 of 128 running on o2wn30.pi.infn.it
Hello world from rank 12 of 128 running on o2wn30.pi.infn.it
Hello world from rank 13 of 128 running on o2wn30.pi.infn.it
Hello world from rank 14 of 128 running on o2wn30.pi.infn.it
Hello world from rank 15 of 128 running on o2wn30.pi.infn.it
Hello world from rank 16 of 128 running on o2wn30.pi.infn.it
Hello world from rank 17 of 128 running on o2wn30.pi.infn.it
Hello world from rank 18 of 128 running on o2wn30.pi.infn.it
Hello world from rank 19 of 128 running on o2wn30.pi.infn.it
Hello world from rank 20 of 128 running on o2wn30.pi.infn.it
Hello world from rank 21 of 128 running on o2wn30.pi.infn.it
Hello world from rank 22 of 128 running on o2wn30.pi.infn.it
Hello world from rank 23 of 128 running on o2wn30.pi.infn.it
Hello world from rank 24 of 128 running on o2wn30.pi.infn.it
Hello world from rank 25 of 128 running on o2wn30.pi.infn.it
Hello world from rank 26 of 128 running on o2wn30.pi.infn.it
Hello world from rank 27 of 128 running on o2wn30.pi.infn.it
Hello world from rank 28 of 128 running on o2wn30.pi.infn.it
Hello world from rank 29 of 128 running on o2wn30.pi.infn.it
Hello world from rank 30 of 128 running on o2wn30.pi.infn.it
Hello world from rank 31 of 128 running on o2wn30.pi.infn.it
Hello world from rank 32 of 128 running on o2wn30.pi.infn.it
Hello world from rank 33 of 128 running on o2wn30.pi.infn.it
Hello world from rank 34 of 128 running on o2wn30.pi.infn.it
Hello world from rank 35 of 128 running on o2wn30.pi.infn.it
Hello world from rank 36 of 128 running on o2wn30.pi.infn.it
Hello world from rank 37 of 128 running on o2wn30.pi.infn.it
Hello world from rank 38 of 128 running on o2wn30.pi.infn.it
Hello world from rank 39 of 128 running on o2wn30.pi.infn.it
Hello world from rank 40 of 128 running on o2wn30.pi.infn.it
Hello world from rank 41 of 128 running on o2wn30.pi.infn.it
Hello world from rank 42 of 128 running on o2wn30.pi.infn.it
Hello world from rank 43 of 128 running on o2wn30.pi.infn.it
Hello world from rank 44 of 128 running on o2wn30.pi.infn.it
Hello world from rank 45 of 128 running on o2wn30.pi.infn.it
Hello world from rank 46 of 128 running on o2wn30.pi.infn.it
Hello world from rank 47 of 128 running on o2wn30.pi.infn.it
Hello world from rank 48 of 128 running on o2wn30.pi.infn.it
Hello world from rank 49 of 128 running on o2wn30.pi.infn.it
Hello world from rank 50 of 128 running on o2wn30.pi.infn.it
Hello world from rank 51 of 128 running on o2wn30.pi.infn.it
Hello world from rank 52 of 128 running on o2wn30.pi.infn.it
Hello world from rank 53 of 128 running on o2wn30.pi.infn.it
Hello world from rank 54 of 128 running on o2wn30.pi.infn.it
Hello world from rank 55 of 128 running on o2wn30.pi.infn.it
Hello world from rank 56 of 128 running on o2wn30.pi.infn.it
Hello world from rank 57 of 128 running on o2wn30.pi.infn.it
Hello world from rank 58 of 128 running on o2wn30.pi.infn.it
Hello world from rank 59 of 128 running on o2wn30.pi.infn.it
Hello world from rank 60 of 128 running on o2wn30.pi.infn.it
Hello world from rank 61 of 128 running on o2wn30.pi.infn.it
Hello world from rank 62 of 128 running on o2wn30.pi.infn.it
Hello world from rank 63 of 128 running on o2wn30.pi.infn.it
Hello world from rank 64 of 128 running on o2wn20.pi.infn.it
Hello world from rank 66 of 128 running on o2wn20.pi.infn.it
Hello world from rank 67 of 128 running on o2wn20.pi.infn.it

OpenMPI, architettura ed uso

- Modular Component Architecture (MCA), Framework (public interfaces), Component and Module
- Module: an MCA **module is an instance of a component** (in the C++ sense of the word "instance": an MCA component is analogous to a C++ class). For example, **if a node running an Open MPI application has multiple ethernet NICs, the Open MPI application will contain one TCP btl component, but two TCP btl modules.** This difference between components and modules is important because modules have private state; components do not.
- Parola chiave: **astrazione**
- Configurazione via MCA parameters (config file/cli)
- Open Run-Time Environment **(ORTE)** →Reference Implementation of the Process Management Interface Exascale standard **(OpenPMIx)** → PMIx Reference RunTime Environment **(PRRTE)**





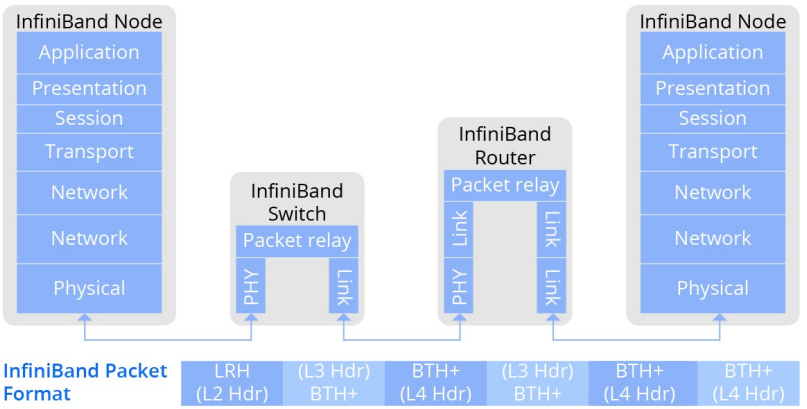


Quindi ?

- bsub -q parallel -G staff
- -e /home/users/formuso/error
- -o /home/users/formuso/output
- -n 128
- /usr/lib64/openmpi/bin/mpirun -n 128 --hostfile hostfile **--mca btl openib,self,sm,^tcp, ...**
- /home/users/formuso/mpi ...
- OpenMPI è già super ottimizzato per ottenere comunicazioni estremamente perfermanti (latenze ridotte)
- Difficilmente si mette mano a questo livello

InfiniBand

- InfiniBand è uno standard di comunicazione ad **alta velocità** e **bassa latenza** (micro secondi) pensato per i sistemi HPC
- Nasce nel ‘1999, fondazione di Mellanox (N.B. oggi di proprietà di Nvidia)
- Network stack riprogettato per ovviare alle inefficienze di TCP/IP (ad esempio: uso del kernel per trasmettere I messaggi): Host Channel Adapter (HCA)
- Plug and Play (fino ad un certo punto)
- **Subnet Mananger (SM)** per ogni sottorete → In esecuzione con licenza sullo switch ib oppure su uno o piu nodi di calcolo
- **Accesso diretto alla memoria (RDMA),** riduzione overhead..
- Introdotto di recente **SHARP** (Scalable Hierarchical Aggregation and Reduction Protocol) a bordo degli switch



InfiniBand unidirectional data rates

	Year[20]	Line code		Signaling rate (Gbit/s)	Throughput (Gbit/s)[21] 1x	4x	8x	12x	Adapter latency (µs)[22]
SDR	2001, 2003	NRZ	8b/10b[23]	2.5	2	**8**	16	24	5
DDR	2005			5	4	**16**	32	48	2.5
QDR	2007			10	8	**32**	64	96	1.3
FDR10	2011		64b/66b	10.3125[24]	10	**40**	80	120	0.7
FDR	2011			14.0625[25][19]	13.64	**54.54**	109.08	163.64	0.7
EDR	2014[26]			25.78125	25	**100**	200	300	0.5
HDR	2018[26]	PAM4		53.125[27]	50	**200**	400	600	<0.6[28]
NDR	2022[26]		256b/257b[i]	106.25[29]	100	**400**	800	1200	?
XDR	2024[30]		[to be determined]	200	200	**800**	1600	2400	[to be determined]
GDR	TBA	[to be determined]		400	400	**1600**	3200	4800	

Cosa è un job parallelo ?

- https://www.youtube.com/watch?v=8T7vZIlxpqI

Il Ciclo di Vita di un Job Parallelo

1) Creazione del programma parallelo
2) Sottomissione
3) Esecuzione: il nodo master del job fa da direttore d'orchestra. Ciascun processo va in esecuzione sul core che gli è stato assegnato.
4) Sincronizzazioni
   - Debole (es. Job di tipo Map & Reduce)
   - Forte (es. Job di CFD con frequente (ogni quanto?) scambio di dati tra processi)
5) Terminazione: Il job viene considerato completato quando tutti i sotto-task sono terminati. I risultati vengono depositati nella home utente.
6) Se siamo arrivati fin qui è un **miracolo !**

Morte di un job parallelo: al via le indagini !

Troubleshooting:
la nostra best practice !

1) Analisi del file di output/error del job fallito
2) Verifica che l’ambiente utente (container) sia up&running
3) Verifica che il file system sia montato e accessibile a tutti i nodi implicati nel job
4) Verifica che i demoni dello schedulel siano in esecuzione sui nodi
5) Verifica di autenticazione: uid utente esiste in tutti i nodi del job ?
6) Verifica raggiungibilità nodi
7) Verifica lo stato di salute delle rete InfiniBand: SM è in esecuzione ?
8) Analisi del job utente (un job non è mai uguale a se stesso)

Caso 1: Errore critico su MPI

- Da ieri molte simulazioni stanno incorrendo in un errore apparentemente completamente random che causa il crash della simulazione. Allego 4 run di esempio, due dei quali hanno avuto questo problema e due dei quali, invece, non l'hanno avuto. Le 4 simulazioni sono completamente identiche e lanciate nelle stesse condizioni con lo stesso numero di core, addirittura dalla stessa cartella.

- bsub
- -q fluent4
- -G fluent
- -J $jobname -o $PWD\${jobname}_run.txt -e $PWD\${jobname}_err.txt
- -n $cores
- /afs/pi.infn.it/pisw/starcd/STAR-CCM+${starversion}/star/bin/starccm+ -jvmargs '-DXmx2g' -power -rsh /usr/bin/ssh -batchsystem lsf -batch $macroname $simname;

```
Job <TEST9> was submitted from host <localui4> by user <toni> in cluster <INFN-HPC>.
Job was executed on host(s) <32*n3wn113>, in queue <fluent4>, as user <toni> in cluster <INFN-HPC>.
                            <32*n3wn96>
</home/users/toni> was used as the home directory.
</home/users/toni/cl_fluent/Scripts/testcases/MRF/pamela_MRF> was used as the working directory.
Started at Fri Jul  8 14:38:33 2022
Results reported at Fri Jul  8 14:40:46 2022

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
/afs/pi.infn.it/pisw/starcd/STAR-CCM+17.02.007-R8/star/bin/starccm+ -jvmargs -DXmx2g -power -rsh /usr/bin/ssh -batchsystem lsf -batch /gpfs/ddn/flue
------------------------------------------------------------

Exited with exit code 1.

Resource usage summary:

    CPU time :                   3624.59 sec.
    Max Memory :                 1407 MB
    Average Memory :             1316.00 MB
    Total Requested Memory : -
    Delta Memory :               -
    (Delta: the difference between total requested memory and actual max usage.)
    Max Swap :                   17 MB

    Max Processes :              169
    Max Threads :                222

The output (if any) follows:

        Supported Java version : 11.0.12+7
Starting local server: /afs/pi.infn.it/pisw/starcd/STAR-CCM+17.02.007-R8/star/bin/starccm+ -power -batchsystem lsf -server -rsh /usr/bin/ssh -xencod
Starting parallel server
--------------------------------------------------------------------------
It appears as if there is not enough space for /dev/shm/vader_segment.n3wn96.47940001.3 (the shared-memory backing
file). It is likely that your MPI job will now either abort or experience
performance degradation.

  Local host:  n3wn96.pi.infn.it
  Space Requested: 4194312 B
  Space Available: 3551232 B
--------------------------------------------------------------------------
[n3wn96.pi.infn.it:18559] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18543] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18565] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18557] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18569] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18563] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18581] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18585] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18567] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18577] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18587] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18583] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18591] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18576] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18589] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18579] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn96.pi.infn.it:18575] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:26580] 14 more processes have sent help message help-opal-shmem-mmap.txt / target full
[n3wn113.pi.infn.it:26580] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
[n3wn113.pi.infn.it:02170] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:02182] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:02178] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:26580] 1 more process has sent help message help-opal-shmem-mmap.txt / target full
[n3wn113.pi.infn.it:02162] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:02174] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:02204] pml_ucx.c:273  Error: Failed to create UCP worker
```

1

```
[n3wn113.pi.infn.it:02176] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:02200] pml_ucx.c:273  Error: Failed to create UCP worker
[n3wn113.pi.infn.it:02146] [[18324,1],12] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02152] [[18324,1],8] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02160] [[18324,1],9] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02166] [[18324,1],19] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02168] [[18324,1],1] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
--------------------------------------------------------------------------
MPI_INIT has failed because at least one MPI process is unreachable
from another.  This *usually* means that an underlying communication
plugin -- such as a BTL or an MTL -- has either not loaded or not
allowed itself to be used.  Your MPI job will now abort.

You may wish to try to narrow down the problem;

 * Check the output of ompi_info to see which BTL/MTL plugins are
   available.
 * Run your application with MPI_THREAD_SINGLE.
 * Set the MCA parameter btl_base_verbose to 100 (or mtl_base_verbose,
   if using MTL-based communications) to see exactly which
   communication plugins were considered and/or discarded.
--------------------------------------------------------------------------
[n3wn113.pi.infn.it:02164] [[18324,1],2] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02144] [[18324,1],4] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02154] [[18324,1],18] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02156] [[18324,1],10] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02172] [[18324,1],27] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02158] [[18324,1],21] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113.pi.infn.it:02142] [[18324,1],3] selected pml ucx, but peer [[18324,1],0] on n3wn113 selected pml cm
[n3wn113:02168] *** An error occurred in MPI_Init
[n3wn113:02168] *** reported by process [1200881665,1]
[n3wn113:02168] *** on a NULL communicator
[n3wn113:02168] *** Unknown error
[n3wn113:02168] *** MPI_ERRORS_ARE_FATAL (processes in this communicator will now abort,
[n3wn113:02168] ***    and potentially your MPI job)
[n3wn96.pi.infn.it:18553] pml_ucx.c:175  Error: Failed to receive UCX worker address: Not found (-13)
[n3wn96.pi.infn.it:18561] pml_ucx.c:175  Error: Failed to receive UCX worker address: Not found (-13)
[n3wn96.pi.infn.it:18533] pml_ucx.c:175  Error: Failed to receive UCX worker address: Not found (-13)
--------------------------------------------------------------------------
It looks like MPI_INIT failed for some reason; your parallel process is
likely to abort.  There are many reasons that a parallel process can
fail during MPI_INIT; some of which are due to configuration or environment
problems.  This failure appears to be an internal failure; here's some
additional information (which may only be relevant to an Open MPI
developer):
```

2

```
[formuso@gridui1 job-starccm]$ cat /afs/pi.infn.it/project/hpc/scripts/start-dk.bash
#!/bin/bash

# Modifica 10/10/2023  da Formuso per usare UCX: --cap-add CAP_SYS_PTRACE --shm-size="8g"
# Prima di questa modifica: --shm-size="2g"

IMG=$1
NET=$2
OPDEV=""
docker pull ${IMG}
for i in `ls /dev/${NET}`;
do
  OPDEV="${OPDEV} --device=/dev/${NET}/${i}"
done
docker run --name=wn-hpc --net=host --rm=true --ulimit memlock=-1 --cap-add=CAP_SYS_PTRACE --shm-size="8g" \
           ${OPDEV} \
           -e container=docker --privileged=true --security-opt seccomp:unconfined --cap-add=SYS_ADMIN \
           -v /afs:/afs -v /cvmfs:/cvmfs -v /gpfs/ddn:/gpfs/ddn -v /chrootlfs/home:/home/grid \
           -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
           -d -t ${IMG} /etc/sysconfig/docker-pi/start
[formuso@gridui1 job-starccm]$
```

3

Caso 2: IO error writing array data

Buongiorno,

Da ieri mi sta capitando molto spesso che lanciando una simulazione non venga salvato il file .sim girato (quello che normalmente ha un @ seguito da un numero) per un errore durante il salvataggio di tipo IO error writing array data.

Abbiamo avuto questo problema in passato cercando di salvare grandi simulazioni con poco spazio disponibile su disco, ma abbiamo 1.7T a disposizione al momento e sto cercando di salvare un file di cui non so la dimensione esatta, ma sono certo non sia superiore alla decina di GB.

A cosa può essere dovuto?

Da dove partiamo ?

- Ovviamente dalla best practice !
- E se alla fine non trovo nessun indizio ? Che direzione prendere ?
- Alcuni indizi (i ticket di “denuncia”) possono sembrare scorrelati all’apparenza… e invece potrebbe esserci un legame.
- Abbiamo a che fare con un **“serial killer”** ? Si, purtroppo avevamo finito gli i-node allocabili sul fileset !
- Se un utente cancellava qualche file si liberavano i-node e il job lanciato dopo funziona fino ad successvo esauriumento i-node

Caso 3: il job è più lento

- Variante 1: Buongiorno, ho lanciato questo job su questo cluster ma risulta essere più lento rispetto a quello lanciato due giorni fa..
- Variante 2: lo stesso job lanciato sul cluster piu piccolo era più veloce..
- Conclusione lato utente: il cluster non funziona !
- Moltissime variabili entrano in gioco…

Caso 3: il job lento...

```
[root@ne2wn30 ~]# ibdiagnet
Loading IBDIAGNET from: /usr/lib64/ibdiagnet1.5.7
-W- Topology file is not specified.
    Reports regarding cluster links will use direct routes.
Loading IBDM from: /usr/lib64/ibdm1.5.7
-I- Using port 1 as the local port.
-I- Discovering ... 37 nodes (1 Switches & 36 CA-s) discovered.


-I---------------------------------------------------
-I- Bad Guids/LIDs Info
-I---------------------------------------------------
-I- No bad Guids were found

-I---------------------------------------------------
-I- Links With Logical State = INIT
-I---------------------------------------------------
-I- No bad Links (with logical state = INIT) were found

-I---------------------------------------------------
-I- General Device Info
-I---------------------------------------------------

-I---------------------------------------------------
-I- PM Counters Info
-I---------------------------------------------------
-W- lid=0x000e guid=0x0002c9020048edb8 dev=48438 Port=17
      Performance Monitor counter      : Value
      port_xmit_discard                : 0xffff (overflow)

-I---------------------------------------------------
-I- Fabric Partitions Report (see ibdiagnet.pkey for a full hosts list)
-I---------------------------------------------------
-I-    PKey:0x7fff Hosts:36 full:36 limited:0

-I---------------------------------------------------
-I- IPoIB Subnets Check
-I---------------------------------------------------
-I- Subnet: IPv4 PKey:0x7fff QKey:0x00000b1b MTU:2048Byte rate:10Gbps SL:0x00
-W- Suboptimal rate for group. Lowest member rate:40Gbps > group-rate:10Gbps

-I---------------------------------------------------
-I- Bad Links Info
-I- No bad link were found
-I---------------------------------------------------
----------------------------------------------------------------
-I- Stages Status Report:
    STAGE                                Errors Warnings
    Bad GUIDs/LIDs Check                 0      0
    Link State Active Check              0      0
    General Devices Info Report          0      0
    Performance Counters Report          0      1
    Partitions Check                     0      0
    IPoIB Subnets Check                  0      1

Please see /var/cache/ibutils/ibdiagnet.log for complete log
----------------------------------------------------------------

-I- Done. Run time was 1 seconds.
```

```
[root@ne2wn30 ~]# ibstat
CA 'mlx4_0'
        CA type: MT4099
        Number of ports: 1
        Firmware version: 2.30.3000
        Hardware version: 1
        Node GUID: 0xf4521403003e34e0
        System image GUID: 0xf4521403003e34e3
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 40
                Base lid: 34
                LMC: 0
                SM lid: 1
                Capability mask: 0x02514868
                Port GUID: 0xf4521403003e34e1
                Link layer: InfiniBand
[root@ne2wn30 ~]# ibstatus
Infiniband device 'mlx4_0' port 1 status:
        default gid:     fe80:0000:0000:0000:f452:1403:003e:34e1
        base lid:        0x22
        sm lid:          0x1
        state:           4: ACTIVE
        phys state:      5: LinkUp
        rate:            40 Gb/sec (4X QDR)
        link_layer:      InfiniBand
```

- Può succedere di trovare un HCA a 10 Gb/sec ?
- Si, si erano rotte 3 coppie all’interno del cavo.

Caso 4:
Ansys Mechanical 2024 R1 – job distribuito parallelo

bash-4.1$ bpeek 15150921 << output from stdout >> NOTE: The -usessh command line option is no longer necessary as SSH is the default protocol for Distributed ANSYS. The -usessh option is ignored.

<< output from stderr >>

[ne2wn31][[7964,1],80]

[**/nfs/lebhomes07/dev2/fluentuser/alex_test/slurm_openmpi/OpenMPI/openmpi-4.0.5/opal/mca/btl/tcp/**
btl_tcp_endpoint.c:626:mca_btl_tcp_endpoint_recv_connect_ack] received unexpected process identifier [[7964,1],104]
[ne2wn31][[7964,1],72][/nfs/lebhomes07/dev2/fluentuser/alex_test/slurm_openmpi/OpenMPI/openmpi-4.0.5/opal/mca/btl/tcp/
btl_tcp_endpoint.c:626:mca_btl_tcp_endpoint_recv_connect_ack] received unexpected process identifier [[7964,1],104]
[ne2wn30][[7964,1],40][/nfs/lebhomes07/dev2/fluentuser/alex_test/slurm_openmpi/OpenMPI/openmpi-4.0.5/opal/mca/btl/tcp/
btl_tcp_endpoint.c:626:mca_btl_tcp_endpoint_recv_connect_ack] received unexpected process identifier [[7964,1],48]
[ne2wn31][[7964,1],96][/nfs/lebhomes07/dev2/fluentuser/alex_test/slurm_openmpi/OpenMPI/openmpi-4.0.5/opal/mca/btl/tcp/
btl_tcp_endpoint.c:626:mca_btl_tcp_endpoint_recv_connect_ack] received unexpected process identifier [[7964,1],104]
[ne2wn30][[7964,1],36][/nfs/lebhomes07/dev2/fluentuser/alex_test/slurm_openmpi/OpenMPI/openmpi-4.0.5/opal/mca/btl/tcp/
btl_tcp_endpoint.c:626:mca_btl_tcp_endpoint_recv_connect_ack] received unexpected process identifier [[7964,1],48]

- 3 differenti indizi saltano all’occhio !

```
formuso@nbformuso:/afs/pi.infn.it/pisw/ansys/structures/v241/commonfiles/MPI/OpenMPI/4.0.5/linx64/bin> readelf -d mpirun

Dynamic section at offset 0x1d88 contains 34 entries:
  Tag        Type                         Name/Value
 0x0000000000000001 (NEEDED)             Shared library: [libopen-rte.so.40]
 0x0000000000000001 (NEEDED)             Shared library: [libopen-pal.so.40]
 0x0000000000000001 (NEEDED)             Shared library: [libdl.so.2]
 0x0000000000000001 (NEEDED)             Shared library: [libudev.so.1]
 0x0000000000000001 (NEEDED)             Shared library: [librt.so.1]
 0x0000000000000001 (NEEDED)             Shared library: [libm.so.6]
 0x0000000000000001 (NEEDED)             Shared library: [libutil.so.1]
 0x0000000000000001 (NEEDED)             Shared library: [libz.so.1]
 0x0000000000000001 (NEEDED)             Shared library: [libpthread.so.0]
 0x0000000000000001 (NEEDED)             Shared library: [libc.so.6]
 0x000000000000000f (RPATH)              Library rpath: [$ORIGIN:/nfs/lebhomes07/dev2/fluentuser/alex_test/slurm_openmpi/OpenMPI/openmpi_4.0.5_linx64-Release/linx64/lib]
 0x000000000000000c (INIT)               0x400bf0
 0x000000000000000d (FINI)               0x4010e4
 0x0000000000000019 (INIT_ARRAY)         0x601d78
 0x000000000000001b (INIT_ARRAYSZ)       8 (bytes)
 0x000000000000001a (FINI_ARRAY)         0x601d80
 0x000000000000001c (FINI_ARRAYSZ)       8 (bytes)
 0x0000000000000004 (HASH)               0x400278
 0x0000000000000005 (STRTAB)             0x400648
 0x0000000000000006 (SYMTAB)             0x400348
 0x000000000000000a (STRSZ)              721 (bytes)
 0x000000000000000b (SYMENT)             24 (bytes)
 0x0000000000000015 (DEBUG)              0x0
 0x0000000000000003 (PLTGOT)             0x602000
 0x0000000000000002 (PLTRELSZ)           336 (bytes)
 0x0000000000000014 (PLTREL)             RELA
 0x0000000000000017 (JMPREL)             0x400aa0
 0x0000000000000007 (RELA)               0x400980
 0x0000000000000008 (RELASZ)             288 (bytes)
 0x0000000000000009 (RELAENT)            24 (bytes)
 0x000000006ffffffe (VERNEED)            0x400960
 0x000000006fffffff (VERNEEDNUM)         1
 0x000000006ffffff0 (VERSYM)             0x40091a
 0x0000000000000000 (NULL)               0x0
```

- Alibi di ferro…
- Test v242 → tandrus
- Escalation
- Grazie a Francesco Laruina

Conclusioni

- Intuito e metodo
- Verificare sempre le ipotesi e le affermazioni
- Le indagini, un mestiere appassionante...
- Quel che non c’è non si può rompere ma ...

```c
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>

int main(int argc, char* argv[])
{
    MPI_Init(&argc, &argv);

    // Get my rank in the communicator
    int my_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    // Determine the rank of the broadcast e
    int sender = 0;
    // Message to broadcast
    char greeting[] = "Grazie per l'attenzio

    if(my_rank == sender)
    {
        printf("[MPI process sender %d] Mess
    }

    MPI_Bcast(&greeting, 1, MPI_CHAR, sender, MPI_COMM_WORLD);

    if(my_rank != sender)
    {
        printf("[MPI process receiver %d] Message received: %s\n", my_rank, greeting);
    }

    MPI_Finalize();

    return EXIT_SUCCESS;
}
```

```
[formuso@o2wn101 ~]$ mpicc mpi_bcast.c -o mpi_bcast
[formuso@o2wn101 ~]$ mpirun -n 8 mpi_bcast
[MPI process sender 0] Message sent: Grazie per l'attenzione. Domande ?
[MPI process receiver 5] Message received: Grazie per l'attenzione. Domande ?
[MPI process receiver 6] Message received: Grazie per l'attenzione. Domande ?
[MPI process receiver 7] Message received: Grazie per l'attenzione. Domande ?
[MPI process receiver 1] Message received: Grazie per l'attenzione. Domande ?
[MPI process receiver 2] Message received: Grazie per l'attenzione. Domande ?
[MPI process receiver 3] Message received: Grazie per l'attenzione. Domande ?
[MPI process receiver 4] Message received: Grazie per l'attenzione. Domande ?
```

Riferimenti

- https://www.open-mpi.org/
- https://openucx.org/
- https://aosabook.org/
- https://openpmix.github.io/
- https://pmix.github.io/features
- https://community.fs.com/
- https://www.r-ccs.riken.jp/en/fugaku/
- https://www.datacentermap.com/italy/
- https://community.fs.com/article/exploring-the-significance-of-infiniband-networking-and-hdr-in-supercomputing.html
- https://www.fujitsu.com/global/imagesgig5/interconnect-of-k-computer.pdf
- https://www.fujitsu.com/dk/imagesgig5/tofu-6d-mesh-torus-interconnect-sc11.pdf
- https://www.fujitsu.com/global/documents/about/resources/publications/technicalreview/topics/article005.pdf
- https://leonardo-supercomputer.cineca.eu/it/leonardo-hpc-system/
- https://en.wikipedia.org/wiki/Frontier_(supercomputer)
- https://www.naddod.com/blog/top-10-advantages-of-infiniband
- https://dl.acm.org/doi/10.1145/3316480.3325517
- https://www.researchgate.net/publication/261313973_On-the-Fly_Adaptive_Routing_in_High-Radix_Hierarchical_Networks
- https://www.laetusinpraesens.org/docs10s/toroidx.php