How to MPI

1. Compile the application using the MPI libraries via the mpiicc command (for CC).

2. Launch using mpirun

1) Compile with mpi

Compile the application using the mpiicc

mpiicc --help

mpiicc --V

2) Use mpirun/SRUN to launch

Run the MPI program using ‘mpirun’ vis SRUN

The default search list is

DAPL	: Infiniband OpenFabrics
TCP	: Ethernet
SHM	: If it can be done (same node)

Examples that we are going to run

1. Hello_world – (compiling it this time)
2. Calculation of PI using more cores
3. Running a ping_pong_ring between two nodes
4. Sort
5. Pallas Test Intel Benckmarks
6. Matrix Multiplication for different RANKS

Get the Examples

- cp /tmp/tests/depot.tgz .
- tar –zxvf depot.tgz

- cd depot

- ls

comms math tests

HELLO WORLD
depot/tests/hello_world.c


```
int main(int argc,char *argv[])
{
    int myid, numprocs, namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    MPI_Get_processor_name(processor_name,&namelen);

    fprintf(stdout,"Hello I am Process %d of %d on %s\n",
       myid, numprocs, processor_name);

    MPI_Finalize();
    return 0;
}
```

HELLO WORLD -depot/tests/hello_world.c

Compilation:

```
cd
cd  depot/tests
mpiicc –o hello_world hello_world.c
```

Execution:

on 5 cores :

- srun -n 5 -N 1 hello_world

on 10 cores :

- srun -n 10 -N2 hello_world

on 40 cores :

- srun -n 40 -N4 hello_world

Calculation of PI - depot/tests/cpi.c

Compilation

cd
cd depot/tests
- mpiicc –o cpi cpi.c

Execution

- srun -n 1 -N 1 cpi
- srun -n 4 -N 1 cpi
- srun -n 8 -N 1 cpi
- srun -n 8 -N 2 –ntasks-per-node=4 cpi

- srun -n 16 -N 4 –ntasks-per-node=4 cpi
- srun -n 32 -N 4 –ntasks-per-node=8 cpi

Run parallel jobs

cd

- cd depot/tests
- mpiicc –o ping_pong_ring ping_pong_ring.c

- export I_MPI_FABRICS=dapl
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring 1000000

- export I_MPI_FABRICS=tcp
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring 100000

- export I_MPI_FABRICS=shm
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring

- export I_MPI_FABRICS=shm:dapl
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring
- srun -n4 -N2 --ntasks-per-node=2 ping_pong_ring 1000000

RUNNING PING_PONG_RING

Latency: 0.40 usec

```
Host 0 -- ip 16.16.187.163 -- ranks 0 - 4

 host | 0
======|======
    0 : SHM

 Prot -  All Intra-node communication is: SHM
```

BW: 1.6 GB/s

Latency: 50 usec

```
Host 0 -- ip 16.16.187.164 -- ranks 0
Host 1 -- ip 16.16.187.165 -- ranks 1
Host 2 -- ip 16.16.187.166 -- ranks 2
Host 3 -- ip 16.16.187.167 -- ranks 3
Host 4 -- ip 16.16.187.168 -- ranks 4

 host | 0    1    2    3    4
======|=========================
    0 : SHM  TCP  TCP  TCP  TCP
    1 : TCP  SHM  TCP  TCP  TCP
    2 : TCP  TCP  SHM  TCP  TCP
    3 : TCP  TCP  TCP  SHM  TCP
    4 : TCP  TCP  TCP  TCP  SHM

 Prot -  All Intra-node communication is: SHM
 Prot -  All Inter-node communication is: TCP
```

BW: 100 MB/s

Latency: 1.60 usec

```
Host 0 -- ip 16.16.187.164 -- ranks 0
Host 1 -- ip 16.16.187.165 -- ranks 1
Host 2 -- ip 16.16.187.166 -- ranks 2
Host 3 -- ip 16.16.187.167 -- ranks 3
Host 4 -- ip 16.16.187.168 -- ranks 4

 host | 0    1    2    3    4
======|=========================
    0 : SHM  IBV  IBV  IBV  IBV
    1 : IBV  SHM  IBV  IBV  IBV
    2 : IBV  IBV  SHM  IBV  IBV
    3 : IBV  IBV  IBV  SHM  IBV
    4 : IBV  IBV  IBV  IBV  SHM

 Prot -  All Intra-node communication is: SHM
 Prot -  All Inter-node communication is: IBV
```

BW: 1.8 GB/s

SORT NUMBERS - depot/math

We are going to sort 50.000.000 elements

cd

cd depot/math

mpiicc –o sort1 sort1.c

echo 50000000 > number

- srun -n 1 sort1 < number
- srun -n 2 sort1 < number
- srun -n 4 sort1 < number
- srun -n 8 sort1 < number
- srun -n 16 sort1 < number

- srun -n 8 -N2 sort1 --ntasks-per-node=4 < number
- srun -n 16 -N2 sort1 --ntasks-per-node=8 < number

- export I_MPI_FABRICS=tcp

- srun -n16 -N2 --ntasks-per-node=8 sort1 < number

srun -n 8 -N2 sort1 --ntasks-per-node=2 < number

Intel Benchmarks

- On the cluster located at /v/appl/opt/cluster_studio_xe/imb
- cd
- cp -r /v/appl/opt/cluster_studio_xe/imb .
- cd imb/3.2.3/src
- make

Intel Benchmarks – the tests

IMB-MPI1		
Single Transfer	Parallel Transfer	Collective
PingPong	Sendrecv	Bcast
PingPing	Exchange	Allgather
		Allgatherv
	Multi-PingPong	Alltoall
	Multi-PingPing	Alltoallv
	Multi-Sendrecv	Reduce
	Multi-Exchange	Reduce_scatter
		Allreduce
		Barrier
		Multi-versions of these

Intel Benchmarks

- export I_MPI_FABRICS=dapl
- srun –n2 –N2 IMB-MPI1
- srun –n2 –N2 IMB-MPI1 PingPong PingPing
- srun –n2 –N2 IMB-MPI1 SendRecv
- export I_MPI_FABRICS=tcp

- srun –n2 –N2 IMB-MPI1 PingPong PingPing
- srun –n2 –N2 IMB-MPI1 SendRecv

MATRIX MULTIPLICATION – 10x10

Compilation

- cd
- cd depot/math
- mpiicc–o matrix2 matrix2.c

- Execution
- srun -n 4 matrix2

Usage : ./matrix2 <matrix filename> <matrix filename> <output filename>

- srun -n 4 matrix2 matA matB output

MATRIX MULTIPLICATION – 1000x1000

- Execution

```
srun -n  1  -N1  matrix2  matA_1k matB_1k output

srun -n  4  -N1  matrix2  matA_1k matB_1k output

srun   -n 8  -N1  matrix2  matA_1k matB_1k output

srun   -n  8  -N 2  --ntasks-per-node-=4 matrix2  matA_1k matB_1k output

srun   -n  16  -N 2 –ntasks-per-node=8 matrix2  matA_1k matB_1k output

srun   -n  24  -N 4 –ntasks-per-node=6 matrix2  matA_1k matB_1k output

srun   -n  32  -N 4 –ntasks-per-node=8 matrix2  matA_1k matB_1k output
```

MATRIX MULTIPLICATION – 1000x1000 SHM vs TCP vs IB

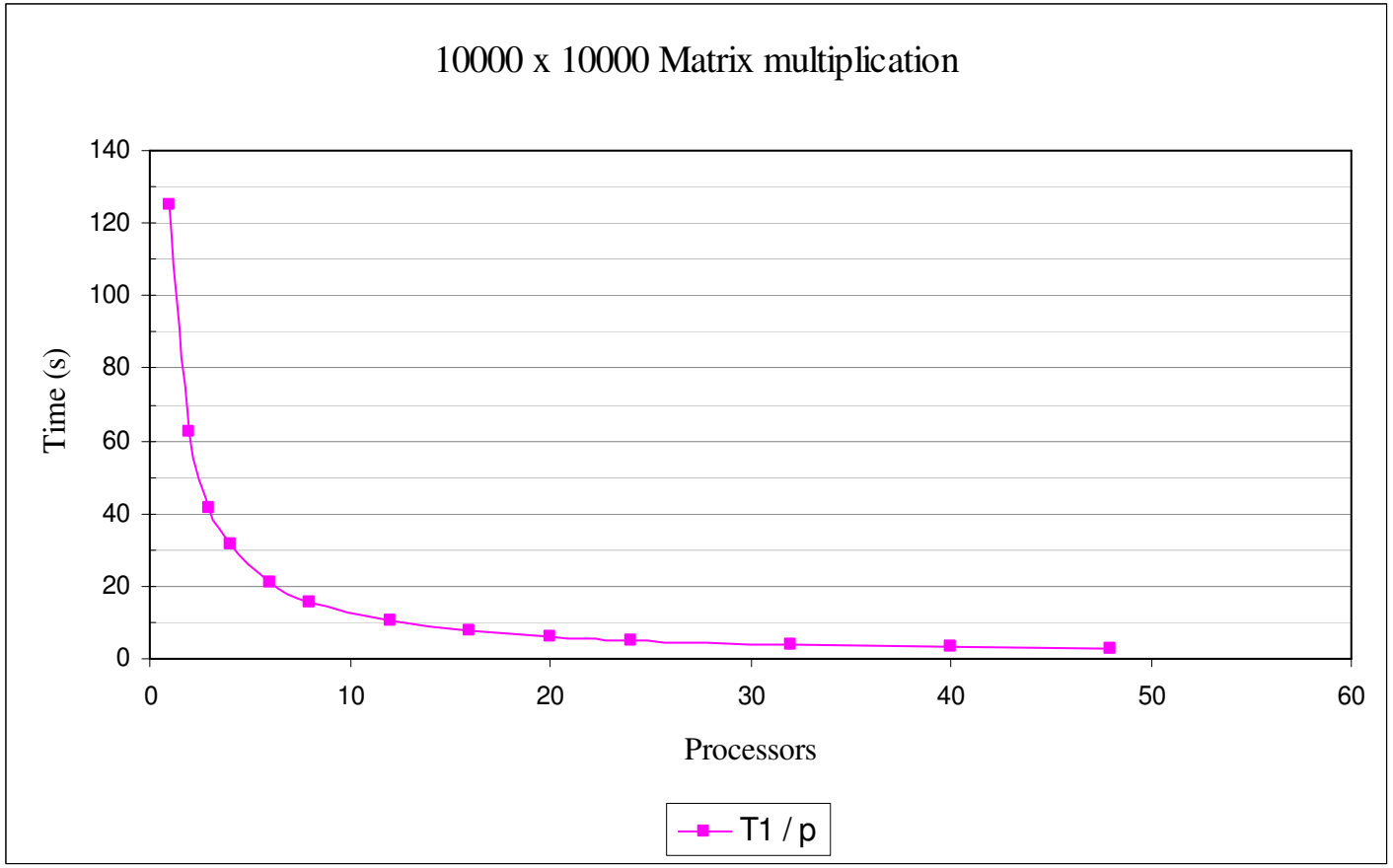
- Execution

```
export I_MPI_FABRICS=shm
srun -n 8 -N 1 matrix2 matA_1k matB_1k output
```

```
export I_MPI_FABRICS=tcp
srun -n 8 -N 4 –ntasks-per-node=2 matrix2 matA_1k matB_1k output
```

```
export I_MPI_FABRICS=dapl
srun -n 8 -N 4 –ntasks-per-node=2 matrix2 matA_1k matB_1k output
```

Multiplication 10000 x 10000




hp
invent