PRACE PATC Course:
Intel MIC Programming Workshop – MPI

LRZ, 27.6.- 29.6.2016
Intel Xeon Phi Programming Models: MPI
MPI on Hosts & MICs

<table>
<thead>
<tr>
<th>Pure MPI</th>
<th>Hybrid MPI/OpenMP</th>
<th>Hybrid MPI/OpenMP</th>
<th>MPI &amp; Offload</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

1 multithreaded MPI-process per core

several multithreaded MPI-processes per core

27-29 June 2016 Intel MIC Programming Workshop
• Default Module:
  – SuperMUC: mpi.ibm/1.4
  – SuperMIC: mpi.intel/5.1

• If you compile for MIC on SuperMUC login nodes use:
  – lu65fok@login07:~> module unload mpi.ibm
  – lu65fok@login07:~> module load mpi.intel
Important MPI environment variables

- Important Paths are already set by intel module, otherwise use:
  - . $ICC_BASE/bin/compilervars.sh intel64
  - . $MPI_BASE/bin64/mpivars.sh

- At LRZ the following MIC-specific environment variables are set per default on SuperMIC:
  - I_MPI_MIC=enable
  - I_MPI_HYDRA_BOOTSTRAP=ssh
  - I_MPI_FABRICS=shm:dapl
  - I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1,ofa-v2-scif0 (must be tuned)
Important MPI environment variables

- Important Paths are already set by intel module, otherwise use:
  - . $ICC_BASE/bin/compilervars.sh intel64
  - . $MPI_BASE/bin64/mpivars.sh

- Recommended environment on Salomon:

  module load intel
  export I_MPI_HYDRA_BOOTSTRAP=ssh
  export I_MPI_MIC=enable
  export I_MPI_FABRICS=shm:dapl
  export I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u,ofa-v2-scif0,ofa-v2-mcm-1
  export MIC_LD_LIBRARY_PATH =
      $MIC_LD_LIBRARY_PATH:/apps/all/impi/5.1.2.150-iccifort-2016.1.150-GCC-4.9.3-2.25/mic/lib/
## Invocation of the Intel MPI compiler

<table>
<thead>
<tr>
<th>Language</th>
<th>MPI Compiler</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>mpiicc</td>
<td>icc</td>
</tr>
<tr>
<td>C++</td>
<td>mpiicpc</td>
<td>icpc</td>
</tr>
<tr>
<td>Fortran</td>
<td>mpiifort</td>
<td>ifort</td>
</tr>
</tbody>
</table>
The following network fabrics are available for the Intel Xeon Phi coprocessor:

<table>
<thead>
<tr>
<th>Fabric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shm</td>
<td>Shared-memory</td>
</tr>
<tr>
<td>tcp</td>
<td>TCP/IP-capable network fabrics, such as Ethernet and InfiniBand (through IPoIB)</td>
</tr>
<tr>
<td>ofa</td>
<td>OFA-capable network fabric including InfiniBand (through OFED verbs)</td>
</tr>
<tr>
<td>dapl</td>
<td>DAPL–capable network fabrics, such as InfiniBand, iWarp, Dolphin, and XPMEM (through DAPL)</td>
</tr>
</tbody>
</table>
The default can be changed by setting the I_MPI_FABRICS environment variable to $I_{\text{MPI}}_{\text{FABRICS}}=\langle\text{fabric}\rangle$ or $I_{\text{MPI}}_{\text{FABRICS}}=\langle\text{intra-node fabric}\rangle:\langle\text{inter-nodes fabric}\rangle$.

- Intranode: Shared Memory, Internode: DAPL (Default on SuperMIC/MUC)
  - export $I_{\text{MPI}}_{\text{FABRICS}}=\text{shm: dapl}$

- Intranode: Shared Memory, Internode: TCP (Can be used in case of Infiniband problems)
  - export $I_{\text{MPI}}_{\text{FABRICS}}=\text{shm: tcp}$
MPI Usage in Native Mode: Best Practices

- When running MPI tasks on several hosts AND Xeon Phi coprocessors, several collective MPI functions like MPI Barriers do not return properly (cause deadlocks).
- In this case set i.e.
  - export I_MPI_DAPL_PROVIDER_LIST=ofa-v2-mlx4_0-1u
  - export I_MPI_ADJUST_BARRIER=1
  - export I_MPI_ADJUST_BCAST=1
- More details can be found under https://software.intel.com/en-us/articles/intel-mpi-library-collective-optimization-on-intel-xeon-phi
- To improve the performance of MPI_Put operations use:
  export I_MPI_SCALABLE_OPTIMIZATION=off
Sample MPI Program

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

int main (int argc, char* argv[]) {
    char hostname[100];
    int rank, size;

    MPI_Init (&argc, &argv);    /* starts MPI */
    MPI_Comm_rank (MPI_COMM_WORLD, &rank); /* get current process id */
    MPI_Comm_size (MPI_COMM_WORLD, &size); /* get number of processes */

    gethostname(hostname,100);
    printf( "Hello world from process %d of %d: host: %s\n", rank, size, hostname);
    MPI_Finalize();
    return 0;
}
```
MPI on hosts

- Compile for host using mpiicc / mpiifort:
  
  ```bash
  lu65fok@login12:~/tests> mpiicc testmpi.c -o testmpi-host
  ```

- Run 2 MPI tasks on host node i01r13a01

  ```bash
  lu65fok@login12:~/tests> mpiexec -n 2 -host i01r13a01 ./testmpi-host
  Hello world from process 0 of 2: host: i01r13a01
  Hello world from process 1 of 2: host: i01r13a01
  ```
MPI in native mode on 1 MIC

- Compile for MIC using mpiicc / mpiifort -mmic:
  `lu65fok@login12:~/tests> mpiicc -mmic testmpi.c -o testmpi-mic`

- Copy binary to MIC:
  `lu65fok@login12:~/tests> scp testmpi-mic i01r13a01-mic0:`

- Launch 2 MPI tasks from MIC node i01r13a01-mic0
  `lu65fok@i01r13a04:~/tests> ssh i01r13a01-mic0
  [lu65fok@i01r13a01-mic0 ~]$ mpiexec -n 2 ./testmpi-mic
  Hello world from process 1 of 2: host: i01r13a01-mic0
  Hello world from process 0 of 2: host: i01r13a01-mic0`
Do not mix up with mpicc and mpifort!!

lu65fok@login12:~/tests> mpicc -mmic testmpi.c -o testmpi-mic
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: skipping incompatible
/lrz/sys/intel/mpi_41_3_048/mic/lib/libmpigf.so when searching for -lmpigf
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: skipping incompatible
/lrz/sys/intel/mpi_41_3_048/mic/lib/libmpigf.a when searching for -lmpigf
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: cannot find -lmpigf
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: skipping incompatible
/lrz/sys/intel/mpi_41_3_048/mic/lib/libmpi.so when searching for -lmpi
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: skipping incompatible
/lrz/sys/intel/mpi_41_3_048/mic/lib/libmpi.a when searching for -lmpi
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: cannot find -lmpi
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: skipping incompatible
/lrz/sys/intel/mpi_41_3_048/mic/lib/libmpigi.a when searching for -lmpigi
/usr/lib64/gcc/x86_64-suse-linux/4.3/../../../../x86_64-suse-linux/bin/ld: cannot find -lmpigi
collect2: ld returned 1 exit status
MPI on 1 MIC

- Compile for MIC using mpiicc / mpiifort -mmic:
  lu65fok@login12:~/tests> mpiicc -mmic testmpi.c -o testmpi-mic

- Copy binary to MIC
  (not necessary if home is mounted on MICs)
  lu65fok@login12:~/tests> scp testmpi-mic i01r13a01-mic0:

- Run 2 MPI tasks on MIC node i01r13a01-mic0
  lu65fok@i01r13a04:~/tests> mpiexec -n 2 -host i01r13a01-mic0
  ./home/lu65fok/testmpi-mic
  Hello world from process 1 of 2: host: i01r13a01-mic0
  Hello world from process 0 of 2: host: i01r13a01-mic0

Full path needed!
MPI on 2 MICs

- Compile for MIC using mpiicc / mpiifort -mmic:
  
  lu65fok@login12:~/tests> mpiicc -mmic testmpi.c -o testmpi-mic

- Copy binary to MICs:
  (not necessary if home is mounted on MICs)

  lu65fok@login12:~/tests> scp testmpi-mic i01r13a01-mic0:
  lu65fok@login12:~/tests> scp testmpi-mic i01r13a01-mic1:

- Run 2 MPI tasks on MIC node i01r13a01-mic0

  lu65fok@login12:~/tests> mpiexec -n 2 -perhost 1 -host i01r13a01-mic0,i01r13a01-mic1 ./home/lu65fok/testmpi-mic

  Hello world from process 1 of 2: host: i01r13a01-mic1
  Hello world from process 0 of 2: host: i01r13a01-mic0

27-29 June 2016
Intel MIC Programming Workshop
MPI on Host and 2 MICs attached to the host

lu65fok@login12:~/tests> mpirun -n 1 -host i01r13a01 ./testmpi-host : -n 1 -host i01r13a01-mic0 /home/lu65fok/testmpi-mic : -n 1 -host i01r13a01-mic1 /home/lu65fok/testmpi-mic

Hello world from process 0 of 3: host: i01r13a01
Hello world from process 2 of 3: host: i01r13a01-mic1
Hello world from process 1 of 3: host: i01r13a01-mic0
Hello world from process 3 of 6: host: i01r13a02
Hello world from process 0 of 6: host: i01r13a01
Hello world from process 2 of 6: host: i01r13a01-mic1
Hello world from process 5 of 6: host: i01r13a02-mic1
Hello world from process 1 of 6: host: i01r13a01-mic0
Hello world from process 4 of 6: host: i01r13a02-mic0
lu65fok@login12:~/tests> cat machinefile.txt
i01r13a01-mic0
i01r13a01-mic1
i01r13a02-mic0
i01r13a02-mic1

lu65fok@login12:~/tests> mpirun -n 4 -machinefile machinefile.txt /home/lu65fok/testmpi-mic
Hello world from process 3 of 4: host: i01r13a02-mic1
Hello world from process 2 of 4: host: i01r13a02-mic0
Hello world from process 1 of 4: host: i01r13a01-mic1
Hello world from process 0 of 4: host: i01r13a01-mic0
lu65fok@login12:~/tests> cat machinefile.txt
i01r13a01-mic0:2
i01r13a01-mic1
i01r13a02-mic0
i01r13a02-mic1

lu65fok@login12:~/tests> mpirun -n 4 -machinefile machinefile.txt /home/lu65fok/testmpi-mic
Hello world from process 3 of 4: host: i01r13a02-mic0
Hello world from process 0 of 4: host: i01r13a01-mic0
Hello world from process 2 of 4: host: i01r13a01-mic1
Hello world from process 1 of 4: host: i01r13a01-mic0
#include <unistd.h>
#include <stdio.h>
#include <mpi.h>

int main (int argc, char* argv[]) {
    char hostname[100];
    int rank, size;
    MPI_Init (&argc, &argv);  /* starts MPI */
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);  /* get current process id */
    MPI_Comm_size (MPI_COMM_WORLD, &size);  /* get number of processes */

    gethostname(hostname,100);

    #pragma offload target(mic)
    {
        char michostname[50];
        gethostname(michostname, 50);
        printf("MIC: I am %s and I have %ld logical cores. I was called by process %d of %d: host: %s \n", michostname, 
            sysconf(_SC_NPROCESSORS_ONLN), rank, size, hostname);
    }

    printf( "Hello world from process %d of %d: host: %s\n", rank, size, hostname);
    MPI_Finalize();
    return 0;
}
Offload from MPI Tasks using 1 host

lu65fok@login12:~/tests> mpiicc testmpioffload.c -o testmpioffload
lu65fok@login12:~/tests> mpirun -n 4 -host i01r13a01 ./testmpioffload
Hello world from process 0 of 4: host: i01r13a01
Hello world from process 1 of 4: host: i01r13a01
Hello world from process 2 of 4: host: i01r13a01
Hello world from process 3 of 4: host: i01r13a01

MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by
process 0 of 4: host: i01r13a01
MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by
process 1 of 4: host: i01r13a01
MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by
process 2 of 4: host: i01r13a01
MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by
process 3 of 4: host: i01r13a01
lu65fok@login12:~/tests> mpirun -n 4 -perhost 2 -host i01r13a01,i01r13a02 ./testmpioffload
Hello world from process 2 of 4: host: i01r13a02
Hello world from process 0 of 4: host: i01r13a01
Hello world from process 3 of 4: host: i01r13a02
Hello world from process 1 of 4: host: i01r13a01
MIC: I am i01r13a02-mic0 and I have 240 logical cores. I was called by process 2 of 4: host: i01r13a02
MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by process 1 of 4: host: i01r13a01
MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by process 0 of 4: host: i01r13a01
MIC: I am i01r13a02-mic0 and I have 240 logical cores. I was called by process 3 of 4: host: i01r13a02
#pragma offload target(mic:rank%2)
{
    char michostname[50];
    gethostname(michostname, sizeof(michostname));
    printf("MIC: I am %s and I have %ld logical cores. I was called by \\
            process %d of %d: host: %s \n", michostname, \\
            sysconf(_SC_NPROCESSORS_ONLN), rank, size, \\
            hostname);
}

Offload from MPI Tasks: Using both MICs

lu65fok@login12:~/tests> mpirun -n 4 -perhost 2 -host i01r13a01,i01r13a02 ./testmpioffload
Hello world from process 0 of 4: host: i01r13a01
Hello world from process 2 of 4: host: i01r13a02
Hello world from process 3 of 4: host: i01r13a02
Hello world from process 1 of 4: host: i01r13a01
MIC: I am i01r13a02-mic1 and I have 240 logical cores. I was called by process 3 of 4: host: i01r13a02
MIC: I am i01r13a01-mic1 and I have 240 logical cores. I was called by process 1 of 4: host: i01r13a01
MIC: I am i01r13a01-mic0 and I have 240 logical cores. I was called by process 0 of 4: host: i01r13a01
MIC: I am i01r13a02-mic0 and I have 240 logical cores. I was called by process 2 of 4: host: i01r13a02
SuperMIC architecture

SuperMIC / Helios:
  two IB ports

(Figure from M. Haefele)
Improved MPI performance

MIC network performance on the Helios supercomputer – new DAPL provider in dat.conf

dat – direct access transport

/etc/dat.conf for mic0
/etc/dat.conf for mic1

Inter-node new dat.conf

<table>
<thead>
<tr>
<th>host0</th>
<th>CPU1</th>
<th>MIC0</th>
<th>MIC1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3340</td>
<td>3338</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3345</td>
<td>3330</td>
<td></td>
</tr>
</tbody>
</table>

Bandwidth (MB/s)

<table>
<thead>
<tr>
<th>CPU1</th>
<th>MIC0</th>
<th>MIC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>host1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lab: MPI
Thank you!