High Performance Numerical Simulation School

2019/11/08

MPI behind the scene

Brice Goglin

TADaaM team – Inria Bordeaux Sud-Ouest

Agenda

- History
- Many ways to use the network
- Configuring OpenMPI
- Checking Performance
  - Am I using the right network?
- What's next?

1 History

MPI is a very old standard

- MPI 1.0 in 1993
- Still widely used despite many complaints
- Updated every ~5 years
  - 2.2 in 2009, 3.1 in 2015, 4.0 in 2020
- Maaaany different implementations
  - One per network and per processor vendor
    - Many of them derive from OpenMPI or MPICH
  - Maaaany different configuration options

1995-2002

- Most HPC clusters used Myrinet or Quadrics networks
  - You had to use the corresponding vendor MPI implementation
    - MPICH-MX, MPICH-Elanlib
    - No serious alternative
      - Vanilla MPICH was not optimized for these networks yet
  - No easy way to use both networks at the same time
  - Things were easy for users and admins

2000-2010 : InfiniBand

- IB is the long-awaited HPC network standard
  - Comes with the OFED open-source network stack
    - “Verbs” API
- MPI implementations are ported to it
  - MVAPICH and OpenMPI
    - MPICH focused on other networks for several years
- Hardware vendors contribute to these implementations
- Things were very still easy for users and admins
  - Once you decided which implementation you’ll use

2005-2010 : Really a standard?

- Standard are nice to users
  - But marketing people prefer talking about raw performance
- Pathscale/Qlogic “TrueScale” InfiniBand isn’t really InfiniBand
  - Users are advised to use PSM instead
  - Intel now recommends PSM2 for OmniPath

2010-2015 : Breaking the standard

- Mellanox hacked the InfiniBand Verbs
  - “Accelerated Verbs”
  - Then MXM
  - Now UCX
- Intel and Mellanox are the only remaining vendors
- The Verbs standard API is obsolete?
  - Deprecated in OpenMPI since 4.0
    - Except for some strange networks (iWarp, RoCE, etc.)
  - Still used by MVAPICH, why?

New Programming Models

- PGAS increasingly used
  - They need support for many network technologies
    - Just like MPI
  - Mellanox pushed OpenSHMEM in OpenMPI
- Task-graph-based runtimes have similar needs

- We need communication libraries that are **programming-model-independent** and **multi-network**
  - libfabric/OFI
  - UCX
- MPI can be implemented on top of them!

2 Many ways to use the network

Different needs

- Applications may use collective communication
- or Point-to-point (send/receive)
- or RDMA (one-sided, put/get)
- or streams of bytes (like TCP)

- Network hardware may support some of these features
  - Other features have to be reimplemented over what's supported

Layers in OpenMPI

- PML = Point-to-point Management Layer
  - UCX
- MTL = Message Transfer Layer
  - PSM, PSM2, OFI
- BTL = Byte Transfer Layer
  - TCP, openib

OpenMPI 4 over Plafrim Miriel

- Miriel001-088 have 40G TrueScale “IB”
  - “openib” BTL = Discouraged
  - PSM MTL
- Miriel001-043 also have 100G OmniPath
  - PSM2 MTL

OpenMPI 4 over Plafrim Mistral and Sirocco01-06

- Mellanox 40G InfiniBand
  - “openib” BTL = Now disabled by default
  - UCX PML = Recommended by Mellanox
  - OFI MTL, with OFI using IB verbs = Why?
  - TCP? over 10G Ethernet or “IP over IB” ?
    - UCX PML or OFI MTL or TCP BTL?

- **The runtime will select the right one for you**
  - Based on hardware and priorities
- **Make sure the right drivers were compiled in!**

3 Configuring OpenMPI

Before compiling

- **Use the latest release**
  - 4.0.2 as of today
  - Subreleases always bring useful fixes
- Former release series are old
  - 3.1.5 is based on 3.1.0 from May 2018
  - 3.0.4 is based on 3.0.0 from September 2017
  - They only get bugfixes, no major changes
    - Network hardware and software changed since then
      - UCX by default, etc.

Open MPI configure script

- Everything detected is enabled by default
- **Add --with-foo to get a failure if foo cannot be found**

```
“configure: error: PSM support requested but not found.  Aborting”
```

  - Good way to make sure PSM/UCX gets enabled

- You should build on a node with development headers for all networks you want to support
  - Doesn’t mean you’ll need all these networks at runtime
    - Plugins are dynamically loaded based on available hardware and libraries

OpenMPI configure for PlaFRIM

- Important
  - --with-ucx --with-psm –with-psm2
    - once UCX is properly installed
- Paranoid
  - --without-ofi --disable-verbs
- Keep your usual options such as --enable-mpirun-prefix-by-default --prefix=…

- **Guix will (soon) do all this for you**

Check the summary at the end of configure

```
Transports
-----------------------
Cisco usNIC: no
Cray uGNI (Gemini/Aries): no
Intel Omnipath (PSM2): yes      ← for miriel001-043
Intel TrueScale (PSM): yes      ← for all miriels
Mellanox MXM: no
Open UCX: yes                   ← for mistral/sirocco
OpenFabrics OFI Libfabric: no   ← I was paranoid
OpenFabrics Verbs: no           ← I was paranoid
Portals4: no
Shared memory/copy in+copy out: yes
Shared memory/Linux CMA: yes
Shared memory/Linux KNEM: no
Shared memory/XPMEM: no
TCP: yes
```

Check available components later

```
$ ompi_info  | grep ucx
    MCA osc: ucx (MCA v2.1.0, API v3.0.0, Component v4.0.2)
    MCA pml: ucx (MCA v2.1.0, API v2.0.0, Component v4.0.2)
$ ompi_info  | grep psm
    MCA mtl: psm (MCA v2.1.0, API v2.0.0, Component v4.0.2)
    MCA mtl: psm2 (MCA v2.1.0, API v2.0.0, Component v4.0.2)

$ ompi_info  | grep Configure
  Configure command line: '--with-ucx' '--with-psm' '--with-
psm2' '--disable-verbs' '--without-ofi' '--enable-mpirun-
prefix-by-default' '--prefix=...'
```

4

Checking Performance

Am I using the right network ?

Which network am I actually using?

- No easy way to be sure
- Use a simple benchmark
  - e.g. Intel MPI Benchmark (“Pingpong” test)
    - https://github.com/intel/mpi-benchmarks/

```
make IMB-MPI1 CC=/my/ompi/bin/mpicc CXX=/my/ompi/bin/mpicxx
```

- Pingpong between 2 nodes, one process per node

```
mpiexec
  -np 2 -H mistral02,mistral03
  --map-by node --bind-to core
  IMB-MPI1 Pingpong
```

100G OmniPath network? (miriel001-043)

- I should get ***about*** 10GB/s unidirectional

```
$ mpiexec -np 2 … IMB-MPI1 Pingpong
            0          1000          1.45          0.00
            1          1000          1.62          0.62
[...]
      2097152            20        215.75       9720.19
      4194304            10        439.37       9546.20
```

- Performance disappears if disabling PSM2
  - --mca mtl ^psm2
  - Or if forcing PSM1
    - --mca mtl psm

40G TrueScale IB network? (miriel001-088)

- I should get ***about*** 4GB/s unidirectional

```
$ mpiexec --mca mtl ^psm2 -np 2 … IMB-MPI1 Pingpong
            0          1000          1.56          0.00
            1          1000          1.68          0.59
[...]
      2097152            20        689.57       3041.23
      4194304            10       1325.47       3164.39
```

- Performance disappears if disabling PSM1 too
  - --mca mtl ^psm2,psm

TCP?

- Plafrim has 10Gbit/s, I should get ***about*** 1GB/s unidirectional

```
$ mpiexec --mca mtl ^psm2,psm -np 2 … IMB-MPI1 Pingpong
            0          1000        16.39          0.00
            1          1000        18.08          0.06
[...]
      2097152            20      2314.69        906.02
      4194304            10      5323.38        787.90
```

Mellanox 40G IB with UCX? mistral and sirocco01-06

- I should get ***about*** 4GB/s unidirectional

```
$ mpiexec -np 2 … IMB-MPI1 Pingpong
            0         1000         1.64         0.00
            1         1000         1.66         0.60
[…]
      2097152           20       567.42      3695.97
      4194304           10      1126.84      3722.18
```

- Performance disappears if disabling UCX
  - --mca pml ^ucx

Mellanox 40G IB with (obsolete) openib? mistral and sirocco01-06

- **Not possible if compiled with --disable-verbs**
- I should get ***about*** 4GB/s unidirectional?

```
$ mpiexec -np 2 --mca pml ^ucx
  --mca btl_openib_allow_ib 1 … IMB-MPI1 Pingpong
            0         1000         1.48         0.00
            1         1000         1.54         0.65
[...]
      2097152           20       576.53      3637.52
      4194304           10      1128.36      3717.16
```

- No too bad for a deprecated stack?

5 What's next?

Are they going to make all this easy?

- Lots of political reason behind all these ways to use the network
  - Intel pushes libfabric, Mellanox (now NVIDIA) pushes UCX
- Developers are looking at easier ways to disable/enable some networks
  - I don’t want TCP, even through OFI
- And easier ways to report what’s used
  - Summary at the end of the job

Standardisation?

- The MPI standard is about the API
  - Not about mpiexec options
    - But we’re looking at improving this anyway
- PMIx is being standardized too
  - Gives a way for application to query the resource manager and runtimes about available resources, networks, …
    - Stay tuned

Questions?

Brice.Goglin@inria.fr