Here are some topics that we talked about in the colloquium, 11th September 2019.

Contents

1 Nodes and clusters

Someone asked what nodes and clusters are. For high-performance computing, the answer is:

"An HPC cluster is a collection of many separate servers (computers), called nodes, which are connected via a **fast** interconnect."

https://www.hpc.iastate.edu/guides/introduction-to-hpc-clusters/what-is-an-hpc-cluster

2 High-performance computing (HPC) versus distributed systems

There is a difference between high-performance computing (HPC) and distributed systems. In our assignments, we use HPC.

Distributed systems

- Many computers cooperating
- "Normal" networking. Slow, unstable network connections.
- Nodes acting on their own
    - Running services independently of each other
- They do not share memory. They are usually "shared-nothing."
- Horizontal scaling
    - Adding power by adding more machines and balancing the load between the computers.
    - For example, to scale a web-based system with many users, we can add more computers to handle more incoming user requests than before.
- Cloud computing is based on distributed computing on a large scale.

High-performance computing (HPC)

- Many computer cooperating
- Connected with expensive equipment (InfiniBand or similar). Very high-throughput, low-latency networking. Comparable to connecting the computers to a shared bus.
- Acting together as one big computer
    - Parallel computing
    - They can all run the same application as together as one (Single Program, Multiple Data)
- Sharing of memory, e.g., with remote direct memory access (RDMA) (https://en.wikipedia.org/wiki/Remote_direct_memory_access)
- Vertical scaling
    - More power by increasing the resources of a single machine, i.e., more processing power, more memory, more storage.
    - We can view HPC as being one powerful machine.
- Large HPC clusters are super-computers.

3 Handling node and network failures in the assignment

Handling failures, such as nodes crashing and network connections failing, is very important in distributed systems. They are built to treat these types of failures as normal.

Things are a bit different when it comes to high-performance computing. We generally expect our cluster to work correctly, or at least that the problems are hidden from us and handled at a lower level. We thus do not worry too much about networks failures and computers that crash in our assignment.

4 Uvcluster.cs.uit.no

Open a bash terminal, such as Ubuntu WSL on Windows 10.

Log in to the cluster with SSH (replace xxx111 with your own username):
ssh xxx111@uvcluster.cs.uit.no

You will then be logged in to a front-end node for the cluster. You will see a welcome message telling how to run scripts to get info about the cluster and compute nodes.

Run this line if you want to compile inside the cluster:
scl enable devtoolset-8 bash

To compile for UDP, remember to set the environment variable first. The programs are compiled for different conduits, so if you change to another, you must set the variable and compile a new version:

export UPCXX_GASNET_CONDUIT=udp

PS: For only running locally at one node, such as locally on your own PC, use this instead:
export UPCXX_GASNET_CONDUIT=smp

Here is one way of running your program in the cluster, by using a comma-separated list with the nodes that you want to use. You can use other compute nodes, of course. The number after -n has to be equal to the number of nodes that you are using:

/share/apps/upcxx-2019.3.2/bin/upcxx-run -ssh-servers xxx111@compute-2-0,xxx111@compute-2-1,xxx111@compute-2-2,xxx111@compute-2-3,xxx111@compute-2-4 -n 5 ./a.out

Clean up by running this script before logging out:
/share/apps/ifi/cleanup.sh

File copy

Files can be copied from your local computer to your home directory in the cluster's frontend node (change the two xxx111 to your own username):
scp some_file.cpp xxx111@uvcluster.cs.uit.no:/home/xxx111/

5 Example scenarios for the queue assignment

Read chapter 10. We are mostly interested in unbounded, non-locking types of queues.

I will try to find some example scenarios for the queue assignment.