HPC

JCN-9000

2019-05-20

HPC

- High Performance Computer : Supercomputer
- High Performance Computing : Parallel
- BOINC is a platform for high-throughput computing "Seti@Home", "Folding@Home", (List of distributed computing projects)

Definition

HPC High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

Wikipedia IT

HPC Con High Performance Computing (HPC) (in italiano calcolo ad elevate prestazioni), in informatica, ci si riferisce alle tecnologie utilizzate da computer cluster per creare dei sistemi di elaborazione in grado di fornire delle prestazioni molto elevate nell'ordine dei PetaFLOPS, ricorrendo tipicamente al calcolo parallelo.

FLOPS

- FLOPS Unit https://en.wikipedia.org/wiki/Unit_prefix

		Metric prefixes in everyday use	
Text	Symbol	Factor	Power
yotta	Y	1000000000000000000000000	10E24
zetta	Z	1000000000000000000000	10E21
exa	E	1000000000000000000	10E18
peta	P	1000000000000000	10E15
tera	T	1000000000000	10E12
giga	G	1000000000 (PC)	10E9
mega	M	1000000	10E6
kilo	k	1000	10E3
hecto	h	100	10E2
deca	da	10	10E1
(none)	(none)	1	10E0

Local FLOPS using HPL/Linpak

- Rate Your PC

Netlib Linpak

- HPL from Netlib
- http://www.netlib.org/benchmark/hpl/
- Build HPL.dat

```
cd ~/GIT/Polito-HPC/hpl-2.3/testing
mpirun -np 2 ./xhpl
```

Intel Linpak

1. Download Intel Linpak (link)
2. Extract
3. cd .../linux/mkl/benchmarks/linpack
4. ./runme_xeon64

```
cd ~/GIT/Polito-HPC/l_mklb_p_2018.3.011
cd benchmarks_2018/linux/mkl/benchmarks/linpack
./runme_xeon64
```

TOP_500

- TOP_500 https://www.top500.org/
- 1-10 LIST
- 1-100

Hardware components

- Power Supply : 10MW (Medium Town)
- Space and Cooling
- Box: IBM, Cray, Fujitsu, Lenovo, HPE, Bull, Dell
- CPU: IBM, Intel, AMD, ARM, Custom
- GPU: AMD, NVIDIA
- Network
  - Mellanox Infiniband
  - Intel OmniPath
  - Ethernet $>$ 10GBit
- Shared Filesystem
  - NFS (Mostly Read)
  - pNFS
  - Lustre
  - Gluster
  - BeeGFS
  - Ceph
- A Lot of $$$

- Useful Tools to manage / use a HPC Cluster
  - Ansible
  - GIT
  - Docker, Singularity, Podman
  - Nagios, Ganglia, Zabbix
  - Job Scheduler (Slurm,PBS,Grid Engine)

Software Components

- Application Code (rewritten from Serial to Parallel)
- Libraries for IPC
- Libraries to use GPU: CUDA, OpenCL
- IDE
- GUI to High Level Tools (Jupyter Notebook)

IPC - Inter Process Communication

- shared files / memory + semaphores
- pipes (named, unnamed)
- message queues (unidirectional)
- sockets (memory, network) (bi-directional)
- signals
- RPC
  - ONC/RPC, XML-RPC -> SOAP, CORBA, JSON-RPC, gRPC

Approaches to message passing

PVM **Parallel Virtual Machine** is a software tool for parallel networking of computers. It is designed to allow a network of heterogeneous Unix and/or Windows machines to be used as a single distributed parallel processor. PVM was a step towards modern trends in distributed processing and grid computing but has, since the mid-1990s, largely been supplanted by the much more successful MPI standard

MPI **Message Passing Interface** is a standardized and portable message-passing standard

- Implementations
  - MPICH
  - MVAPICH
  - OpenMPI
  - Commercial : Intel, HP, Microsoft

- - :

HPC Single Node

Compile and Run - Hello World

```
int main ( int argc, char *argv[] );
{
  printf ( "  Hello, world!\n" );

  return 0;
}
```

- mpicc -o hello hello.c
- mpirun hello
- mpirun --use-hwthread-cpus hello
- mpirun --use-hwthread-cpus -np 4 -tag-output hello
- mpirun --use-hwthread-cpus -np 4 --bind-to hwthread -report-bindings -tag-output hello

Compile and Run - hello_mpi

- hello_mpi.c (*see on Editor*)
- mpicc -o hello_mpi hello_mpi.c
- mpirun hello_mpi
- mpirun --use-hwthread-cpus -np 4 hello_mpi
- mpirun --use-hwthread-cpus --bind-to hwthread -np 4 -report-bindings hello_mpi

Output of hello_mpi

```
  Process 3 says 'Hello, world!'

  Process 2 says 'Hello, world!'
  Process 1 says 'Hello, world!'
HELLO_MPI - Master process:
  C/MPI version
  An MPI example program.

  The number of processes is 4.

  Process 0 says 'Hello, world!'
  Elapsed wall clock time = 0.000342 seconds.

HELLO_MPI - Master process:
  Normal end of execution: 'Goodbye, world!'

10 May 2019 07:43:00 PM
```

Compile and Run - ring

- ring_c.c (*see on Editor*)
- mpicc -o ring_c ring_c.c
- mpirun --use-hwthread-cpus --bind-to hwthread -np 4 -report-bindings ring_c

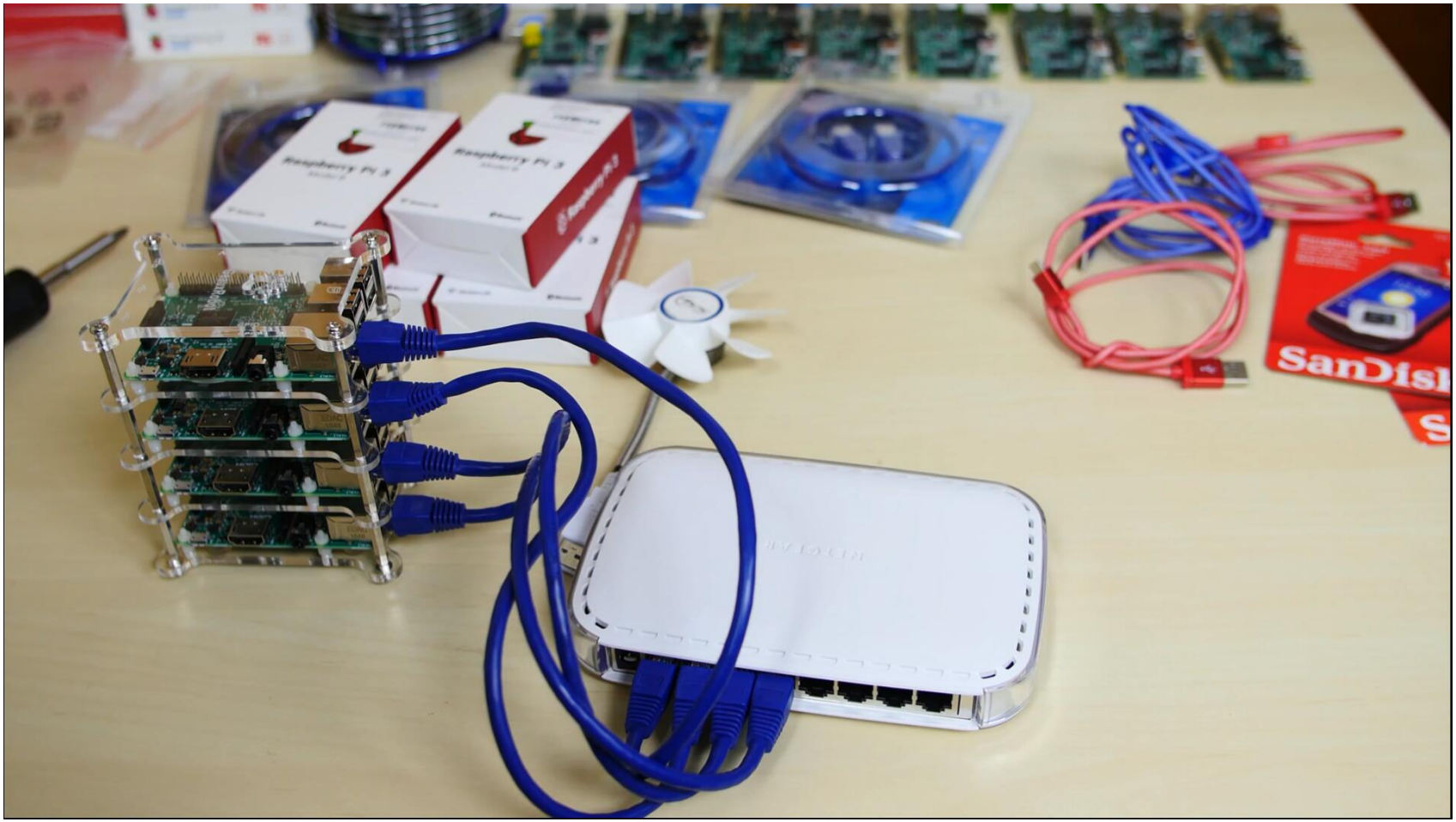
Compile and Run - Search Serial

- search_serial.c (*see on Editor*)
- gcc -Ofast -o search_serial search_serial.c
- ./search_serial
- Elapsed CPU time is 23.3898

Compile and Run - Search MPI

- search_mpi.c (*see on Editor*)
- mpicc -Ofast -o search_mpi search_mpi.c
- (*use nmon to see CPU load*)
- mpirun --bind-to hwthread -np 3 search_mpi
- Elapsed wallclock time is 7 / 15.3633

HPC Multinode



HPC Build Your Own

- http://www.admin-magazine.com/HPC/Articles/Building-an-HPC-Cluster
- http://hpc.fs.uni-lj.si/sites/default/files/HPC_for_dummies.pdf
- https://openhpc.community/downloads/
- https://opensource.com/article/18/1/how-build-hpc-system-raspberry-pi-and-openhpc
- http://bccd.net/
- https://www.rocksclusters.org/
- https://pelicanhpc.org/

Pelican HPC

PelicanHPC
https://www.pelicanhpc.org/index.html

PelicanHPC over VM

- cd hpl-2.0
- sh SetupForPelican
- cd bin/Pelican
- mpirun --hostfile /home/user/tmp/bhosts -np 2 xhpl
- mpirun --hostfile /home/user/tmp/bhosts -np 4 xhpl

MPI on multiple Nodes

- mpicc -o hello hello.c
  - mpirun hello
- mpicc -o hello_mpi hello_mpi.c
  - mpirun hello_mpi
- mpicc -o ring_c ring_c.c
  - mpirun --hostfile /home/user/tmp/bhosts -np 4 ring_c
- gcc -Ofast -o search_serial search_serial.c
  - ./search_serial
- mpicc -Ofast -o search_mpi search_mpi.c
  - (*use nmon to see CPU load*)
  - mpirun -hostfile /home/user/tmp/bhosts -np 3 search_mpi

URLS

- HPC @ Polito http://hpc.polito.it/
- https://openhpc.community

JupyterLab

- https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks

Tensorflow

```
docker run -it --rm -p 8888:8888 --mount
type=bind,source=/home/avaresio/GIT/Polito-
HPC/Jupyter,destination=/home/jovyan/work
jupyter/tensorflow-notebook start.sh jupyter lab
```

DataScience

```
docker run -it --rm -p 8888:8888 --mount
type=bind,source=/home/avaresio/GIT/Polito-
HPC/Jupyter,destination=/home/jovyan/work
jupyter/datascience-notebook start.sh jupyter lab
```

Questions

Sponsor

- DoIT - http://www.doit-systems.it
- DoIT - Work With US

- Alberto 'JCN-9000' Varesio
- mailto:Alberto.Varesio@doit-systems.it
- mailto:Alberto.Varesio@gmail.com
- Slides on GIT: https://github.com/JCN-9000/Polito-HPC

Presentation Tools used

TXT2TAGS(git) + LaTEX Beamer

```
alias S='/home/avaresio/GIT/txt2tags/txt2tags -t txt2t Slid
pandoc -s -t beamer -V theme:Copenhagen -V colortheme:wolve
sed -i "/^ *:$/d" Slides.tex
sed -i "/^ *-$/d" Slides.tex
pdflatex Slides.tex > Slides.plog
rm Slides{.out,.vrb,.toc,.snm,.nav,.aux,.log,.plog}'
```

Impressive

```
alias I='impressive --noquit -f -d 1:30:00 -M -g 1280x1024
 --page-progress --time-display --tracking \
 Slides.pdf'
```

Skip

Further IDEAS

- Q commands
- Queuing system
- GitPitch

PelicanHPC over KVM

PelicanHPC over Virtualbox

- sudo apt update
- sudo apt install libatlas3-base
- sudo apt install gpm