Glossary

A comment was made: "One of the problems we have as HPC pro's is discussing the technologies that exist or that we support in a way that's accessible to our clients and customers who may not be as well versed in those technologies as we are." We hope this glossary is helpful.

HPC

High performance computing. Implies a program too large for, or that takes too long on, a laptop or workstation. Also HTC (high throughput computing) similar but oriented to processing many small compute jobs.

Node aka Compute Node

A single computer in a box, functionally similar to a desktop computer but typically more powerful and packaged for rackmount in a datacenter. Usually two CPU sockets or four sockets with very large memory vs. one socket for a desktop. Ocelote standard nodes have 28 cores and 192GB memory.

Cluster

A group of nodes connected to each other by a fast network. The network in ElGato and Ocelote is 56Gb Infiniband. What this gains for the user is the ability to connect nodes together to perform work beyond the capacity of a single node. Some jobs use hundreds of cores and terabytes of memory.

Supercomputer

A large and powerful cluster. We currently have three: Puma, Ocelote, and ElGato.

Login node

A cluster node accessible to users and dedicated to logins, editing, moving data, submitting jobs.

Head node

The head node is for managing the cluster and is not available to users.

Data mover node

A node connected to the public internet and dedicated to moving data to/from external computers. We have two DTN nodes known collectively as filexfer.hpc.arizona.edu.

HPC scheduler

A program that maintains a list of batch jobs to be executed on a cluster, ranks them in some priority order, and executes batch jobs on compute nodes as they become available. It tries to keep the cluster from being overloaded or idle. Puma, Ocelote, and ElGato use SLURM.

CPU/processor/socket/core

These terms are often used interchangeably, especially processor and CPU. The most straight forward way to think of the compute nodes is that they contain two physical sockets (or processor chips) which are located under their heatsinks. Each socket contains multiple cores. Each core functions like a separate processor. Ocelote has 2 sockets with 14 cores in each so all you need to know is that there are 28 cores. ElGato has 2 sockets with 6 cores in each, for a total of 12 cores. If your laptop is quad core, it has one socket with four cores, as a comparison.

Hyperthreading

Intel processors (in this case "cores") have hyper-threading which can make one core look like two; but it does not add compute capacity in most HPC cases, so we turn it off.

GPU

A graphical processing unit, a specialized type of CPU derived from a graphics card. Effectively has hundreds of small cores. For certain tasks (those that can be effectively parallelized), a GPU is much faster than a general-purpose CPU.

Network latency

In HPC terms, it is usually the delay in the network for messages being passed from one node to another. This is optimized by a hardware technology called RDMA (Remote Direct Memory Access)

Network bandwidth

The amount of data that can be moved over a network per second. For FDR Infiniband on Ocelote that is 56Gbps (Giga bits per second)

Shared memory computing

A program that runs multiple tasks or software threads, each of which sees the same available memory available from the operating system, and shares that memory using one of the multiple shared memory /multi-threading communication methods (OpenMP, pthreads, POSIX shm, MPI over shared memory, etc.). Shared memory programs cannot run across multiple nodes. Implies a limit (a little less than the amount of memory in the node) to the memory size of the running program.

Distributed memory computing

In software, a program or group of programs that run on multiple nodes or shared-memory instances and use programs such as MPI to communicate between the nodes. In hardware, a cluster that runs distributed-memory programs. Distributed-memory programs are limited in memory size only by job limits to support many users.

MPI computing

Message passing interface, software standard used for most programs that use distributed memory. MPI calls lower-level functions, either networking or shared memory. On a cluster that means it can run transparently either on one node or multiple nodes. MPI has multiple implementations (OpenMPI, MVAPICH, OpenMPI or Intel MPI) that must be used consistently to both compile and run an MPI program.

Single-threaded computing

A software program that cannot take advantage of multi-threading because it was written without multi-threading support. Essentially can use only one core on one node regardless of the number of cores available. Multiple single-threaded programs can be run on a single node on multiple cores.

Storage hierarchy

Each tier of storage is larger and slower than the preceding tier. The first is data in the processor including the processor cache. The next tier is memory. Page or swap is an extension of memory but is very inefficient since it actually writes to disk. You should next consider /tmp which is the local disk on each node. You have no access to /tmp once the job ends. Shared storage is all of /home, /groups/PI, and /xdisk, and is the slowest.

Scratch storage

A temporary file system, designed for speed rather than reliability, and the first tier in the storage hierarchy. On Ocelote and ElGato these are internal SATA disks and referenced as /tmp.

SSD

Solid state disk, memory chips packaged with an interface that appears to the computer to be a disk drive. Faster than rotating disk drives and still more expensive, though decreasing in price over time.

VM or virtual machine

This compute model is not usually found in the HPC environment. It is a method of running several or many virtual machines on one physical machine. Since HPC nodes are busy most of the time the cost of the VM overhead and management is not worthwhile.

Parallel programming

A program that is either multi-tasking (like MPI) or multi-threaded (like OpenMP) or both, in order to effectively use more cores and more nodes and get more computing done. May be either shared-memory or distributed-memory. Unlike a serial program.

Parallel scaling

The efficiency of a parallel program, usually defined as the parallel speedup of the program divided by the number of cores occupied. Speedup is defined as the serial run time divided by the parallel run time. Usually parallel computing introduces overhead, and scaling is less than 1 (or 100%). In most cases, scaling starts at 1 on 1 core (by definition) and decreases as more cores are added, until some point is reached at which adding more cores adds overhead and makes the program slower.

Embarrassingly parallel

A program where little effort is involved in separating the code into parallel tasks and so parallel scaling is very efficient. Some astronomy codes fit this model.