Uses of parallel computing

- OS development
  - Multitasking/multiprocessing allow two processes to run "simultaneously" on one computer.
  - Makes responsive GUI's possible, makes running more than one program at a time possible, etc.
- Game development
  - Modern games use multiple threads to address different facets of the game
  - Rendering engine redraws the scene as quickly as it can
  - Physics engine updates positions of objects as events happen
  - AI engine

- Scientific computing
  - Many problems of scientific interest are simply too large to be solved using a single processor using reasonable amount of memory

The idea

In any kind of parallel processing the idea is the same. Given a task to achieve

- The task is broken in to independent parts
- Multiple processes are started. Each process utilizes independent computing resources that can not be utilized by a single process.
- Each process completes a different part of the task same time. Communication between the processes may be required.
- The results of each part of the task are assembled in to a final result.

Major Parallel computing paradigms

- Shared Memory
  - independent computing resources are multiple cores or multiple cpu's residing within one machine
  - each CPU has access to a common memory store
  - communication, synchronization, etc take place through shared memory
- Distributed Memory
  - independent computing resources are multiple cores or cpu's residing on different machines
  - each CPU has its own dedicated memory – it has no way to directly write to or read from another CPU's memory
  - communication and synchronization take place over a network

Major Parallel computing paradigms

- GPU Computing
  - independent computing resources are streaming multiprocessors (cores) on the GPU
  - SMP's are grouped, and have shared memory within the group. There is no common shared memory for all SMP's.
  - Communication between SMP's is very difficult and costly.

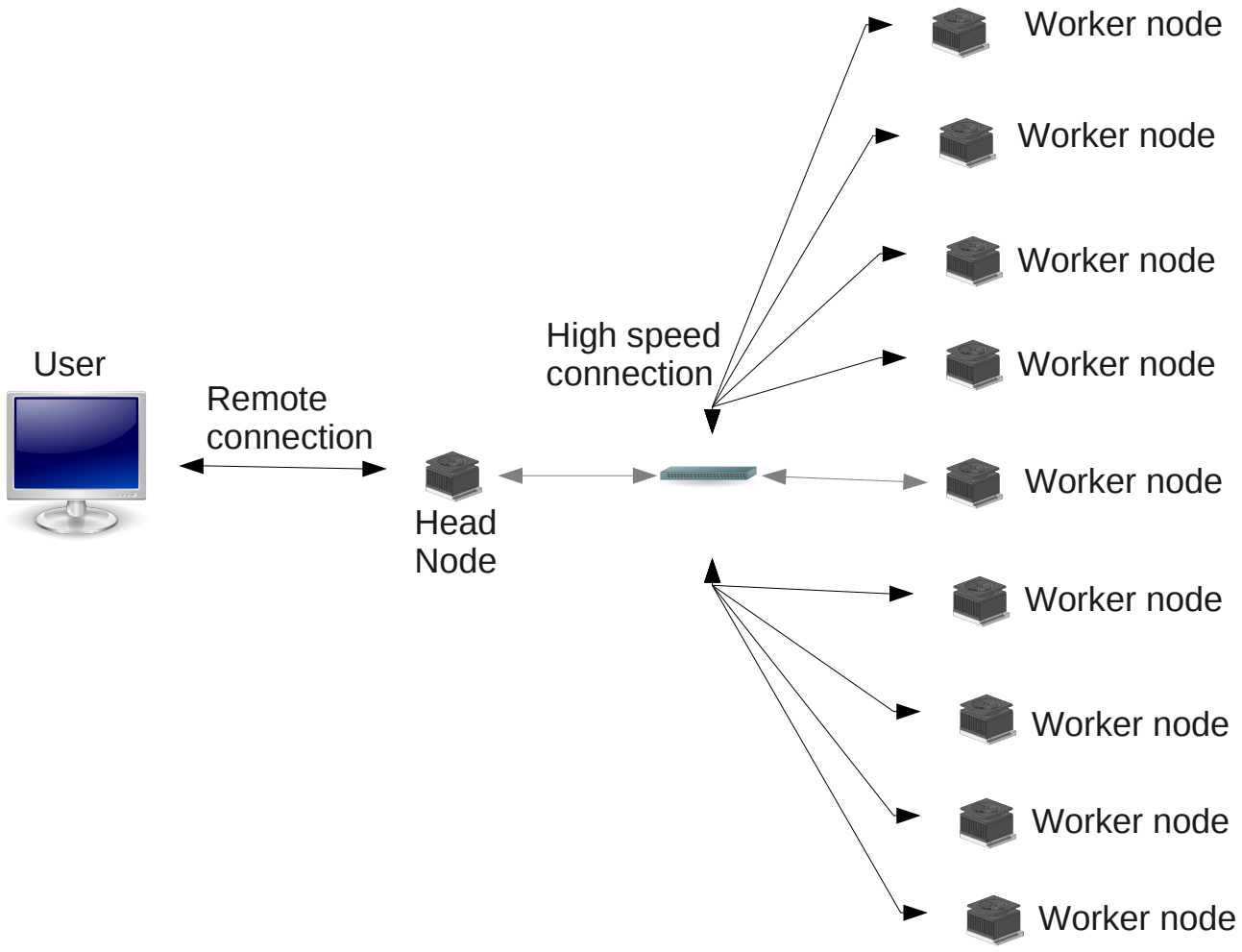
Parallel computing resources on campus

- Shared memory computing
  - any modern computer has multiple cores
  - dayhoff.math.rose-hulman.edu has 48 cores
- Distributed memory computing
  - can always be simulated using multiple cores on your laptop
  - mpi-master.csse.rose-hulman.edu has 32 nodes with a total of 192 AMD 6172 2.1 GHz cores available
- GPU computing
  - Any modern video card made by nVidia supports GPU computing. Most modern AMD video cards do as well
  - clive.math.rose-hulman.edu has 2 Tesla M1060 GPGPU's available

Using the mpimaster cluster

- The mpi-master cluster is a typical example of a beowulf class cluster.
- This is a shared computing resource. You log in to the cluster remotely, from your own computer.
- The OS on the cluster is CentOS
- Our language will be C
- We will use the MPI library/ runtime environment to facilitate communication between nodes, remote execution, etc.

Typical cluster configuration



Connecting to the head node using **S**ecure **SH**ell protocol

- From Windows
  - Use Secure CRT/Secure FX
  - protocol is SSH2, hostname is mpi-master.csse.rose-hulman.edu, use SFTP for FTP protocol.
- From Linux/Mac
  - ssh mpi-master.csse.rose-hulman.edu from command line.

Transferring files to the shared file system

- This cluster is the only machine on campus (that I'm aware of) that has its own, non-networked filesystem. Files on AFS, DFS, anywhere are not available on this file system.
- From Windows
  - Use SFTP net-drive.
    https://www.eldos.com/files/sftpnetdrive2/SftpNetDriveFree.exe
    Mounts your remote directory as a local folder.
  - Use Secure FX. Less nice, but works ok.
- From Linux
  - sshfs (I'm not sure this is installed automatically. You might need yum install sshfs or apt-get install sshfs)
    sshfs username@mpimaster.csse.rose-hulman.edu mountpoint
    example
    sshfs eichholz@mpimaster.csse.rose-hulman.edu ./CSSE
- From Mac
  - Maybe sshfs? For sure some sftp program can do this.

Once you have connected and can transfer files

The machines that make up this cluster are running CentOS. You will need to be minimally proficient using a Linux based OS to use this cluster.

Handy Linux commands

- To run an executable that is not in the path:
  `$ ./executablename`
  for example:
  `$ ./sayhi`
- `kpasswd` – change kerberos password
- `ls` – list contents of current directory
- `rm` *filename* – remove file
- `rmdir` *dirname* – remove directory
- `mkdir` *dirname* – make a directory
- `mv` *origfile newfile* – move a file
- `cat` *filename* – show contents of file

More Linux commands

- `CTRL+Z` – stop the current process
- `CTRL+C` – kill the current process
- `man` – get help on any of the commands listed so far. For example,
  `$ man ls`
  gives you the help (manual) pages about `ls`.
- http://www.tuxfiles.org/linuxhelp/cli.html is a pretty good read
- http://www.tuxarena.com/static/intro_linux_cli.php is also very useful
- http://www.cplusplus.com/doc/tutorial/