Lecture 16: HPC Capabilities

Monday, February 26, 2024 10:29 AM

Topics for today:
- Basics of HPCs
- Introduction to PACE
- SLURM Scheduler
  - Resources to request

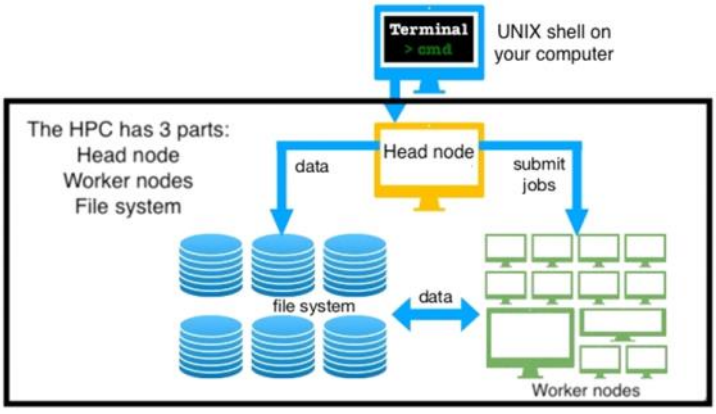
Announcements:
- Proposal: due this Friday

Material References:
- GT PACE online documentation & tutorial slides
- Victor Eijkhout - "Introduction to High Performance Scientific Computing"
- Various online resources

Basics of High Performance Computing (HPC):

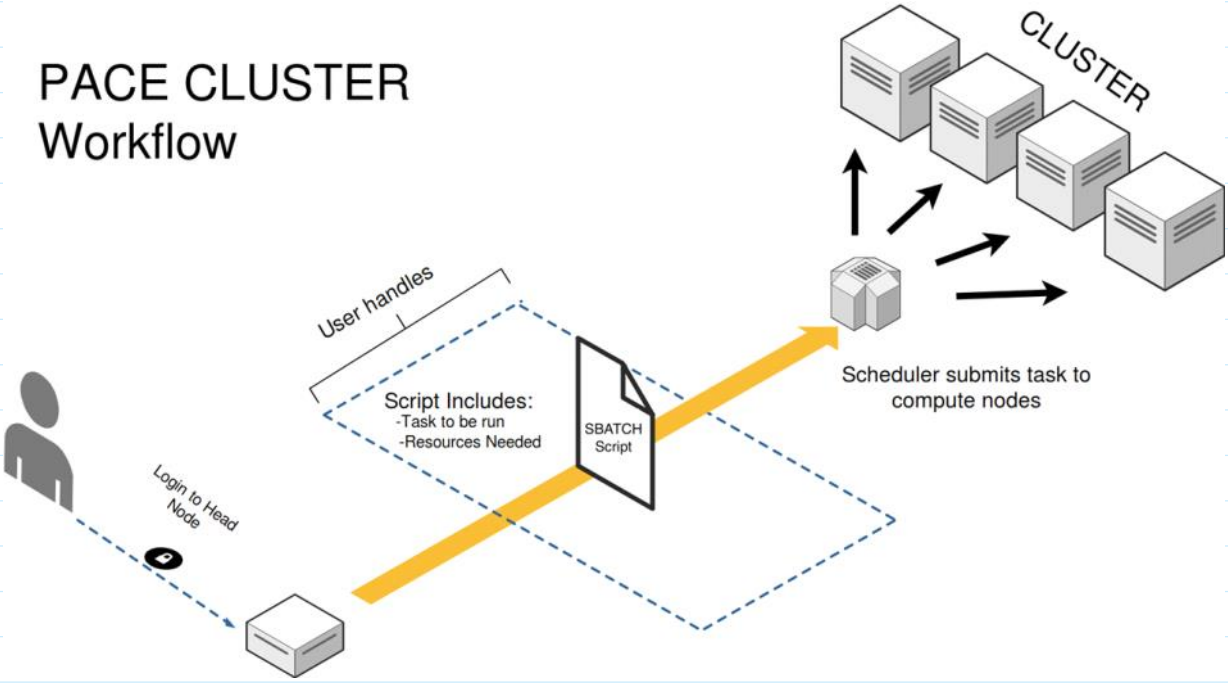
- HPC - technology that uses clusters of powerful processors working in parallel to process large multi-dimensional data and solve complex problems at high speeds
  - typically increase speed 1,000,000 x that of a normal computer
  - allows us to tackle problems that would orindarily be difficult (or impossible) to solve

- Head vs. compute node
  - Head: the machine that you log into
    - shared resource
    - use for editing code & data management
    - not good for doing computations
  - Compute: machines that run all calculations
    - no direct access by users
    - allocated per-job by scheduler



Introduction to PACE :

- Partnership for Advanced Computing Environment



- multiple clusters : ICE ★
  Phoenix
  Hive
  Firebird

- How to log into Head node?
  1. connect to GT VPN
  2. open terminal w/ SSH client
  3. `ssh <GT_user_ID>@<headnode>.pace.gatech.edu`

     Phoenix - login-phoenix.pace.gatech.edu

     Hive - login.hive.pace.gatech.edu

③ ssh <GT_user_ID>@<headnode>.pace.gatech.edu

Phoenix - login-phoenix.pace.gatech.edu

Hive - login-hive.pace.gatech.edu

ICE- login-ice.pace.gatech.edu

- how does the data storage work?
  - data is accessible from all head & compute nodes
  - 3 storage directories:
    - ① home - 10 GB on Phoenix & Hive
      - 15 GB on ICE
    - ② project storage
      - storage depends on amt purchased by your PI, single quota for group
      - not available on ICE
    - ③ scratch - 15 TB (Phoenix), 7 TB (Hive), 100 GB (ICE)
      - deleted every 60 days

- data transfers btwn local computer & PACE

- For fast and reliable data migration, please use **Globus** (https://www.globus.org) via these endpoints:
  - Hive – PACE Hive
  - Phoenix – PACE Phoenix
  - http://docs.pace.gatech.edu/storage/globus/
- For small file copies, you may use scp
  - scp -r ~/mylocalstuff <username>@<login-node>.pace.gatech.edu:~/

- loading pre-installed software on PACE

- `module spider`: Lists all software and its available versions on cluster
- `module avail`: Lists all available modules that can be loaded with current environment
- `module list`: Displays all the modules that are currently loaded
- `module load`: Loads a module to the environment
- `module rm`: Removes a module from the environment
- `module purge`: Removes all loaded modules

```
$ module load matlab/r2021a
```

module load python

IN-CLASS EXAMPLE - run python script on head node

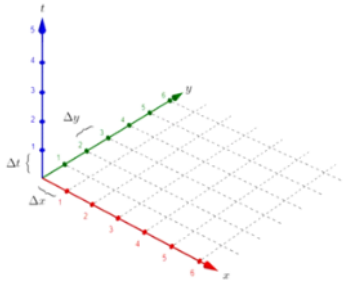
consider the heat equation in 2D:

$$\frac{\partial u}{\partial t} - \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \longrightarrow ①$$

α ↑ diffusivity constant

*want to solve for u in x&y for all t

Approach this via finite difference method:

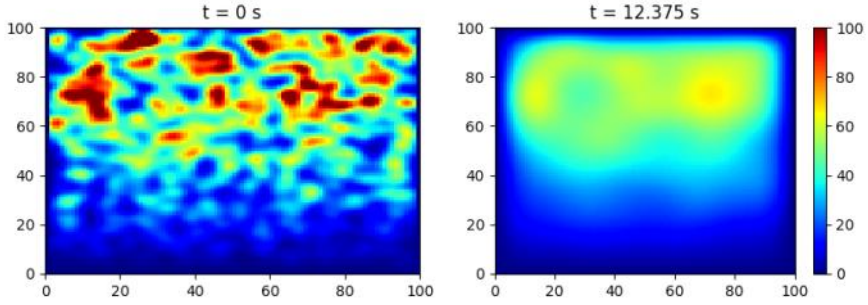


- discretize spatial & time domain
- apply definition of derivative to Eq. ①

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\Delta t} - \alpha\left(\frac{u_{i+1,j}^{k} - 2u_{i,j}^{k} + u_{i-1,j}^{k}}{\Delta x^2} + \frac{u_{i,j+1}^{k} - 2u_{i,j}^{k} + u_{i,j-1}^{k}}{\Delta y^2}\right) = 0$$

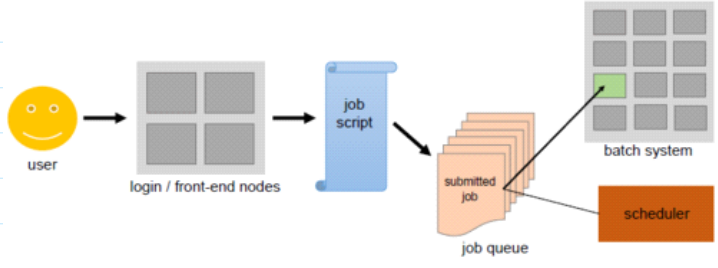
$$\Rightarrow u_{i,j}^{k+1} = \gamma\left(u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - 4u_{i,j}^{k}\right) + u_{i,j}^{k}$$

where $\gamma = \alpha\frac{\Delta t}{\Delta x}$ and $\Delta x = \Delta y$ (square grid)



SLURM scheduler:

- users submit batch jobs (non-interactively) to the scheduler
- scheduler stores batch job, evaluates the resource requirements and priorities, then distributes to compute nodes



- a few different popular schedulers

- a few different popular schedulers

  SLURM ★
  PBS
  Torque

- how to write a batch script for SLURM

  ↳ Nodes (--nodes)
  - check on pace to see available nodes
  - depends on: CPU of job
    memory job needs
    GPU or not?
  - could specify range of nodes -N [8-16]

  ↳ CPU per node (--ntasks-per-node)
  - how many CPU cores does job need.
  - could omit -N if unsure and only specify CPUs

  ↳ memory per CPU (--mem-per-cpu)
  - depends on complexity of job & how much data it generates
    ex. 4 CPU, 1 node, 100 GB/CPU:
    ⇒ node will provide 400GB

  ↳ GPU (--gres)
  - only use if job specifically uses GPU

```
#!/bin/bash
#SBATCH -JSlurmPythonExample                # Job name
#SBATCH -N1 --ntasks-per-node=4             # Number of nodes and cores per node required
#SBATCH --mem-per-cpu=1G                    # Memory per core
#SBATCH -t15                                # Duration of the job (Ex: 15 mins)
#SBATCH -oReport-%j.out                     # Combined output and error messages file
#SBATCH --mail-type=BEGIN,END,FAIL          # Mail preferences
#SBATCH --mail-user=gburdell3@gatech.edu    # E-mail address for notifications
cd $SLURM_SUBMIT_DIR                        # Change to working directory

module load anaconda3                       # Load module dependencies
srun python test.py                         # Example Process
```

- cheat sheet for common commands on PACE:

  ① pace-quota: check available storage / charge acct
  ② pace-check-queue ice-cpu -c
  → PACE compute node availability
  ③ pace-job-summary <job #>
  → overview of job

- Place compute node accordingly

3. pace-job-summary <job #>
   → overview of job
4. salloc -N1 --ntasks-per-node=#

5. srun → run script
6. sbatch → submit batch job
7. squeue -A <GT-username>
   → check on queue status

IN-CLASS PROBLEM – Submit batch script to SLURM
– run GPU ex. on compute node
