High-Performance Scientific Computing

INFO 0939, Fall 2017

http://www.montefiore.ulg.ac.be/~geuzaine/INFO0939/

Based in part on material from Victor Eijkhout’s SSC 335/394 course at the Texas Advanced Computing Center
Introduction
Mathematics & Science

• In science, we use mathematics to understand physical systems.

• Different fields of science explore different ‘domains’ of the universe, and have their own sets of equations, encapsulated in theories.

• Determining the theories and governing equations requires observation or experimentation, and testing hypotheses.
**THE GRAND CHALLENGE EQUATIONS**

\[ B_i A_i = E_i A_i + \rho_i \sum_j B_j A_j F_{ji} \]

\[ \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \]

\[ \vec{F} = m \ddot{a} + \frac{dm}{dt} \dot{v} \]

\[ dU = \left( \frac{\partial U}{\partial S} \right)_V dS + \left( \frac{\partial U}{\partial V} \right)_S dV \]

\[ \nabla \cdot \vec{D} = \rho \]

\[ Z = \sum_j g_j e^{-E_j/kT} \]

\[ F_j = \sum_{k=0}^{N-1} f_k e^{2\pi i j k/N} \]

\[ \nabla^2 u = \frac{\partial u}{\partial t} \]

\[ \nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{j} \]

\[ \nabla \cdot \vec{B} = 0 \]

\[ P(t) = \frac{\sum_i W_i B_i(t) P_i}{\sum_i W_i B_i(t)} \]

\[ p_{n+1} = r p_n (1 - p_n) \]

\[ -\frac{\hbar^2}{8\pi^2 m} \nabla^2 \Psi(r,t) + V \Psi(r,t) = -\frac{\hbar}{2\pi i} \frac{\partial \Psi(r,t)}{\partial t} - \nabla^2 u + \lambda u = f \]

\[ \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{1}{\rho} \nabla p + \gamma \nabla^2 \vec{u} + \frac{1}{\rho} \vec{F} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f \]

*NEWTON’S EQUATIONS* • *SCHROEDINGER EQUATION (TIME DEPENDENT)* • *NAVIER-STOKES EQUATION* •
*POISSON EQUATION* • *HEAT EQUATION* • *HELMHOLTZ EQUATION* • *DISCRETE FOURIER TRANSFORM* •
*MAXWELL’S EQUATIONS* • *PARTITION FUNCTION* • *POPULATION DYNAMICS* •
*COMBINED 1ST AND 2ND LAWS OF THERMODYNAMICS* • *RADIOSITY* • *RATIONAL B- SPLINE* •

[Courtesy of San Diego Supercomputer Center]
Scientific Computing

- Why should we care about scientific computing?
  - Computational research has emerged to complement experimental methods in basic research, design, optimization, and discovery in all facets of engineering and science
  - In certain cases, computational simulations are the only possible approach to analyze a problem:
    - Experiments may be cost prohibitive (e.g. flight testing a 1,000 fuselage/wing-body configurations for a modern fighter aircraft)
    - Experiments may be impossible (e.g. interaction effects between the International Space Station and Shuttle during docking)
  - Simulation capabilities rely heavily on the underlying compute power (e.g. amount of memory, total compute processors, and processor performance)
    - Fostered the introduction and development of super-computers starting in the 1960’s
    - Large-scale compute power is tracked around the world via the Top500 List (more on that later)
Scientific Computing: a definition

• “The efficient computation of constructive methods in applied mathematics”
  – Applied math: getting results out of application areas
  – Numerical analysis: results need to be correctly and efficiently computable
  – Computing: the algorithms need to be implemented on modern hardware
Examples of Scientific Computing

(it really is everywhere)

Streamlines for workstation ventilation

Heating, ventilation, and air conditioning

Electrical Engineering

Automotive
Examples of Scientific Computing

(it really is everywhere)
Examples of Scientific Computing

(it really is everywhere)

Temperature and natural convection currents in the eye following laser heating.

Fields induced in human body close to power lines.
Report on Blackout Is Said To Describe Failure to React

A report on the Aug. 14 blackout identifies specific lapses by various parties, including FirstEnergy's failure to react properly to the loss of a transmission line, people who have seen drafts of it say.

A working group of experts from eight states and Canada will meet in private on Wednesday to evaluate the report, people involved in the
The Top500 List

• [http://www.top500.org](http://www.top500.org)

• Owner submitted benchmark performance since 1993
  – based on a dense linear system solve
Top 500 by Overall Architecture

ARCHITECTURES

- SIMD
- MPP
- SMP
- Constellations
- Clusters

Year: '93 to '12
Top 500 by Microprocessor

CHIP TECHNOLOGY

- ALPHA
- IBM
- MIPS
- HP
- INTEL
- SPARC
- AMD
- PROPRIETARY
Top 500 by Operating System
HPC and CECI

- CECI is the “Consortium des Equipements de Calcul Intensif” in Wallonia/Brussels
  - [http://www.ceci-hpc.be](http://www.ceci-hpc.be)
  - Once you create an account you can use all the CECI clusters
  - Funded by FNRS
  - UCL, ULB, FUNDP, UMons, ULg
  - Single login for all clusters (more on that later)

- Machines in CECI grid:
  - NIC4 @ ULg
  - HMEM & Lemaitre2 @ UCL
  - Vega @ ULB
  - Hercules @ FUNDP
  - Dragon1 @ UMons
NIC4 System Summary

- 120 compute nodes with two 8-cores Intel E5-2650 processors at 2.0 GHz and 64 GB of RAM (4 GB/core)
- QDR Infiniband network
- 144 TB FHGFS parallel filesystem
- The cluster is especially designed for massively parallel jobs (MPI, several dozens of cores) with many communications and/or a lot of parallel disk I/O, 2 days max.
- SSH to nic4.segi.ulg.ac.be with the appropriate login and id_rsa.ceci file (more on that later).
External Power and Cooling
Class Goals/Topics

• Remember that definition “The efficient computation of constructive methods in applied mathematics”
  – Numerical analysis/algorithms, (parallel) computation, and how to combine them

• Theory topics: architecture, numerical analysis, implementing the one on the other

• Practical skills: the tools of scientific computing
Class Goals/Topics

• UNIX Exposure
  – shells/command line
  – environment
  – compilers
  – libraries

• Ideally: good practices for scientific software engineering
  – version control
  – build systems
  – data storage
  – debugging skills
Class Setup

• Theory classes on Tuesday @ 2pm
• 3 or 4 projects, combining theory and programming
Computer Accounts

• CECI clusters
  – You should create an account now, and run your codes over there
  – Must access page from the ULg subnet: https://login.ceci-hpc.be

• All required info is available on http://www.ceci-hpc.be

• Don’t forget to read the FAQ!

• Jobs run in a managed environment
  – login to the login node
  – submit jobs to the scheduler
  – wait
  – collect results

• Production runs on the login node are forbidden
  – avoid resource intensive tasks
  – exceptions include compilers, “standard” UNIX commands (ls, mkdir, etc.)
Remote Login on CECI Clusters

• Only SSH access is allowed
  – Windows users: Get a client
    • PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/)
    • or use Cygwin (http://www.cygwin.com/) and follow the UNIX instructions

• All required info is available on
  – http://www.ceci-hpc.be

• Don’t forget to read the FAQ: