Introduction

ICS632: Principles of High Performance Computing

Henri Casanova (henric@hawaii.edu)

Fall 2015

What is this all about?

- High Performance Computing (HPC): "How do we make computers compute bigger problems faster?"
- This field is both old and new, very diverse, complicated, interesting
- Two main issues
  - How do we build faster/bigger computers?
  - How do we write faster software for those computers?
- Several different perspectives, from practical to theoretical
  - Computer Architecture
  - Operating Systems
  - Networks
  - Programming Languages and Models
  - Algorithms

Parallelism

- The key HPC technique is parallel computing
  - It takes 100 hours to run my program on the fastest processor available today
  - We can't make a faster CPU due to technological/physical limitations
  - Let's run the program on 100 such CPUs
  - And perhaps it can run in 1 hour
- Parallel computing is a huge part of this course
- The course is hands-on
  - You will write parallel code
- with a small theoretical part
  - See Prof. Sitchinava grad-level course for full-fledge theory

Syllabus

- Programming assignments (50% of grade)
  - With a short report to write for each based on experiments
  - With occasional "pencil-and-paper" portions
- Research paper presentations (15% of grade)
  - List of papers will be given
  - I know you're not (yet) researchers in this field
  - The point is that presentations are good practice, and great ways to identify/answer questions
- Final project (35% of grade)
  - Possible topics will be provided
  - Defining your own topic is encouraged!
  - Graded based on written report and quality of work
  - In-class project presentations
- No final exam

Programming Language

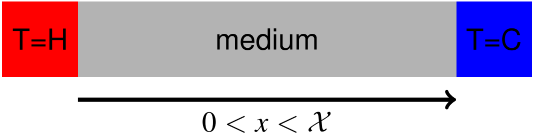
- All programming assignments will be in C
  - Unless you really, really want to do Fortran :)
  - Feel free to use C++ if you want to
- The code we write won't however require that you are a great C developer, but that you know your way around
- Developing in C is pretty typical in the HPC world
- The concept in the course are not really language-specific
- Your final project can be in any language, as long as it's appropriate
  - But if you run into weirdness with a language I know nothing about, I won't be able to help you much

Why HPC?

- The first consumer of HPC was *scientific computing*
  - "640 KiB of RAM" (Bill Gates, 1981) was never enough for scientists
  - It's not enough for anybody now
- Scientific computing has driven HPC for decades
- Traditional scientific and engineering:
  1. Do theory or paper design
  2. Perform experiments or build system
- Limitations:
  - Too difficult – cannot build large wind tunnels
  - Too expensive – cannot build a throw-away airplane
  - Too slow – cannot wait for climate or galactic evolution
  - Too dangerous – drug design, nuclear reactor explosion
- Solution: use a computer to run simulations of physical phenomena

Scientific Computing Example

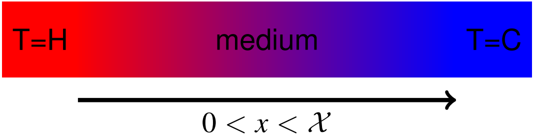
- Consider a scientist studying heat transfer: how temperature changes through a medium over time (in 1 dimension)



- $f(x,t)$: temperature at location x at time t
  - Initial condition: $\forall x, \quad f(x,0) =$ initial temp

Scientific Computing Example

- Consider a scientist studying heat transfer: how temperature changes through a medium over time (in 1 dimension)



- $f(x,t)$: temperature at location x at time t
  - Initial condition: $\forall x, \quad f(x,0) =$ initial temp
- Problem: compute f for all x and $t > 0$ values

Heat Transfer and Physics

- If you've paid attention in Physics you know that:

$$\frac{\partial f(x,t)}{\partial t} = \alpha \frac{\partial^2 f(x,t)}{\partial^2 x}$$

- Boundary conditions: $f(0,t) = H, f(\mathcal{X},t) = L, f(x,0) = \Theta$

- This is a PDE (Partial Differential Equation), and if you've paid attention in Calculus you know that some PDEs can be solved analytically
- Unfortunately, the above equation cannot!
- Therefore it must be solved numerically

Heat Transfer and Numerical Methods

- The Finite Differences Method:
  - Discretize the domain: the values of $f(x,t)$ are known only for some finite number of values of x and t
  - The discretized domain is called a mesh with all x values separated by Δx and all t values separated by Δt
  - So that we write $f(x,t)$ as $f_{i,m}$ (a 2-D array!)
  - Replace partial derivatives by algebraic differences
  - In the limit, when Δx and Δt go to zero, we get close to the real solution to the PDE
- There are many algebraic difference approximations, e.g., Forward Time, Centered Space:

$$f_{i,m+1} = f_{i,m} + \frac{\alpha \Delta t}{\Delta x^2}(f_{i+1,m} - 2f_{i,m} + f_{i-1,m})$$

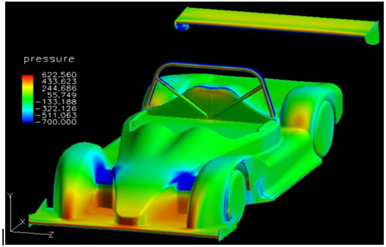
Wait? My beautiful Physics is... two loops?

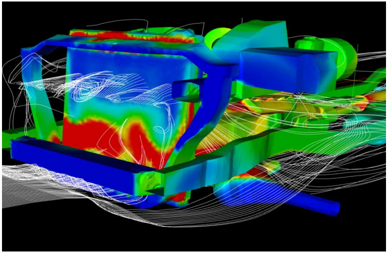
$$f_{i,m+1} = f_{i,m} + \frac{\alpha \Delta t}{\Delta x^2}(f_{i+1,m} - 2f_{i,m} + f_{i-1,m})$$

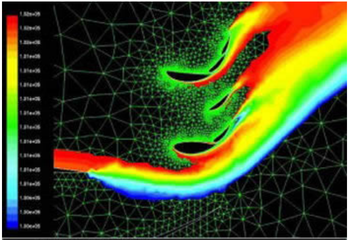
Finite Differences

```
// initialization of boundary conditions
for (i=1; i < I-1; i++) {
  A[i][0] = InitialTemperature;
}
for (m=0; m < M; m++) {
  A[0][m] = Hot;
  A[I-1][m] = Cold;
}

// Computation of all values
factor = alpha * Deltat / (Deltax * Deltax);
for (m=1; m < M; m++) {
  for (i=1; i < I-1; i++) {
     A[i][m] = A[i][m-1] + factor * (A[i+1][m-1] - 2 * A[i][m-1] + A[i-1][m-1];
  }
}

// Draw pretty pictures, understand the universe, etc.
```

Pretty (but Useful) Pictures







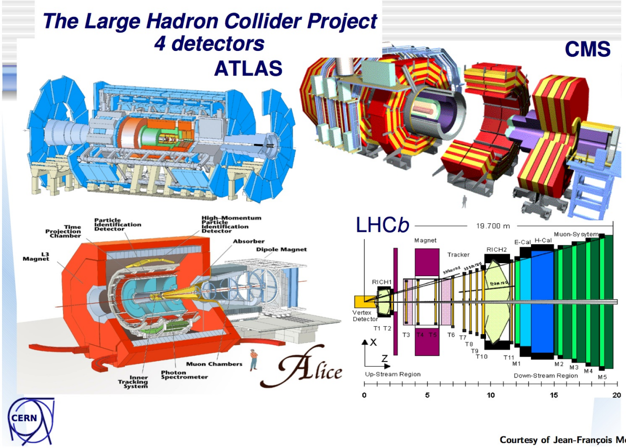
High downforce rear wing configuration
Computational grid & contours of total pressure

Toward HPC

- We replace difficult calculus by simple arrays, iterations, additions, multiplication, and divisions
  - A lot of which is linear algebra in the end
- But there are new challenges
- The data is large
  - And often can be made almost arbitrary large and still useful
- The computation is long
  - And often can be made almost arbitrary long and still useful
- Hence the constant need for HPC (bigger, faster) for scientific computing

Large Data Analysis

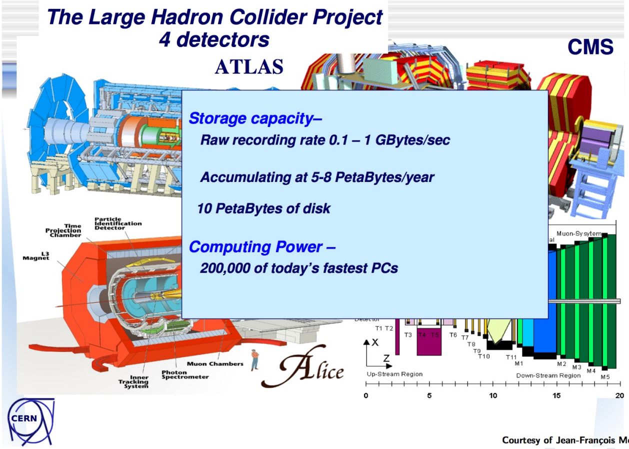
- In may science/engineering applications, HPC is used to analyze scientific data, as opposed to producing it



Courtesy of Jean-François Méhaut

Large Data Analysis

- In may science/engineering applications, HPC is used to analyze scientific data, as opposed to producing it



Courtesy of Jean-François Méhaut

HPC is Important

- Although HPC has decades of success for science and engineering, it is now applied across all kinds of domains
  - The "Big Data" craze of course
- Parallel computing is key and occurs at many levels (from within a core to across the Internet)
  - It has been re-branded with many buzzwords, but fundamental principles we learn in this course apply
- So, how do we teach this?

What do we run on?

- Some of you have already asked me: "What kind of machine do I need for this HPC course???"
- Years ago, a student in this course build his own parallel platform at home with 8 machines he had in his studio apartment (he had to crank the A/C way up).
- This was awesome, but not expected of anybody
- You won't need anything but a Linux (virtual) box for this course

- The typical approach is to provide students with accounts on a parallel platforms
  - Many such platforms are available at universities

Using a Parallel Platform to teach HPC?

Using a Parallel Platform to teach HPC?

- ☺ Students get to work on real systems

Using a Parallel Platform to teach HPC?

- ☺ Students get to work on real systems
- ☹ There are a lot of warts (including downtimes!) which can obfuscate general principles, not good for "beginners"

Using a Parallel Platform to teach HPC?

- ☺ Students get to work on real systems
- ☹ There are a lot of warts (including downtimes!) which can obfuscate general principles, not good for "beginners"
- ☹ "Hey, can you give accounts to my inexperienced students on your system that's being used by eager researchers?" is often a tough sell because systems are fragile and busy

Using a Parallel Platform to teach HPC?

- ☺ Students get to work on real systems
- ☹ There are a lot of warts (including downtimes!) which can obfuscate general principles, not good for "beginners"
- ☹ "Hey, can you give accounts to my inexperienced students on your system that's being used by eager researchers?" is often a tough sell because systems are fragile and busy
- ☹ The instructor keeps saying "if the machine had a better network...", "if the machine was larger...", "if the machine was heterogeneous...", which is not great pedagogy

Using a Parallel Platform to teach HPC?

- ☺ Students get to work on real systems
- ☹ There are a lot of warts (including downtimes!) which can obfuscate general principles, not good for "beginners"
- ☹ "Hey, can you give accounts to my inexperienced students on your system that's being used by eager researchers?" is often a tough sell because systems are fragile and busy
- ☹ The instructor keeps saying "if the machine had a better network...", "if the machine was larger...", "if the machine was heterogeneous...", which is not great pedagogy
- ☹ Before an assignment is due students compete for the machine with other users and with each other, which leads to ugliness if the machine is small/busy

Using a Parallel Platform to teach HPC?

- ☺ Students get to work on real systems
- ☹ There are a lot of warts (including downtimes!) which can obfuscate general principles, not good for "beginners"
- ☹ "Hey, can you give accounts to my inexperienced students on your system that's being used by eager researchers?" is often a tough sell because systems are fragile and busy
- ☹ The instructor keeps saying "if the machine had a better network...", "if the machine was larger...", "if the machine was heterogeneous...", which is not great pedagogy
- ☹ Before an assignment is due students compete for the machine with other users and with each other, which leads to ugliness if the machine is small/busy
- ☹ The instructor must deal with "the cluster is broken" issues and doesn't want to teach HPC anymore?

Simulation!!!

- 12 years ago I started developing a simulator for parallel/distributed computing
- It's since become a large open-source project: **http://simgrid.gforge.inria.fr**
- Only for distributed-memory assignments
  - Shared-memory assignments will be on your own machines, since in this day and age we all have multi-core machines
  - If you don't, I'll give you an account somewhere...
- Let's see what's good/bad about simulation...

Using Simulation to Teach HPC?

- ☹ The simulation could be inaccurate
  - But decade of simulation accuracy research behind us! ☺

Using Simulation to Teach HPC?

- ☹ The simulation could be inaccurate
  - But decade of simulation accuracy research behind us! ☺
- ☹ There are some things we cannot simulate (well)
  - Applications with data-dependent behavior

Using Simulation to Teach HPC?

- ☹ The simulation could be inaccurate
  - But decade of simulation accuracy research behind us! ☺
- ☹ There are some things we cannot simulate (well)
  - Applications with data-dependent behavior
- ☹ / ☺ Students don’t experience a real platform

Using Simulation to Teach HPC?

- ☹ The simulation could be inaccurate
  - But decade of simulation accuracy research behind us! ☺
- ☹ There are some things we cannot simulate (well)
  - Applications with data-dependent behavior
- ☹ / ☺ Students don't experience a real platform
- ☺ The simulation is repeatable, reliable, and fast

Using Simulation to Teach HPC?

- ☹ The simulation could be inaccurate
  - But decade of simulation accuracy research behind us! ☺
- ☹ There are some things we cannot simulate (well)
  - Applications with data-dependent behavior
- ☹ / ☺ Students don't experience a real platform
- ☺ The simulation is repeatable, reliable, and fast
- ☺ The simulation allows to explore many scenarios
  - A huge pedagogical advantage

Using Simulation to Teach HPC?

- ☹ The simulation could be inaccurate
  - But decade of simulation accuracy research behind us! ☺
- ☹ There are some things we cannot simulate (well)
  - Applications with data-dependent behavior
- ☹ / ☺ Students don't experience a real platform
- ☺ The simulation is repeatable, reliable, and fast
- ☺ The simulation allows to explore many scenarios
  - A huge pedagogical advantage
- ☺ The instructor should be happy-er..-ish?
  - Much easier to help you debug a simulation than to deal with a flaky cluster with angry users
    - But there are non-flaky clusters with ~~nice~~ ok users!

Last Semester

- Last time I taught ICS632 it was **100% simulation**
- The result was very positive
  - Most students were happy with it, and thought it made learning/understanding easier
- But there was some mild "disappointment" to not get to work with a real big machine

- This semester, **we'll do both**

UHM's Cray CS Series "Supercomputer"

- UHM now has a big Cray machine and we are getting accounts! you're welcome
- We'll also do simulation because pedagogically it's great
- `http://www.hawaii.edu/its/ci`
- More information to come...

Word of "Warning"

- Several students from last semester came to my office and said: "Wow! You're using a real supercomputer this semester, I am so jealous!"
  - Even though I had been telling them that it wasn't a big deal
- They requested a demo
- So I gave one
- ...
- Let's just say it's a bit anticlimactic...
  - Login via ssh, type a few commands, wait a long time, see nothing, oooh I got an e-mail telling me my program is done

Course Web Site

http://henricasanova.github.io/ics632_fall2015/

Announcement: Upcoming Xeon Phi Workshop

- http://citi.clemson.edu/workshop/hawaii-phi15/
- Would a large fraction of you be interested?