CEE 618 Scientific Parallel Computing (Lecture 1): Introduction

Albert S. Kim

Department of Civil and Environmental Engineering
University of Hawai‘i at Manoa
2540 Dole Street, Holmes 383, Honolulu, Hawaii 96822
Outline

1. Introduction
2. Background
   - Computation
   - Message Passing Interface (MPI)
   - Simulation Tools
   - Examples
     - Parallel Operation
     - Monte Carlo and $\pi$ Calculation
3. SSH
4. UNIX/LINUX Basics
5. Editors
6. FORTRAN
Parallel Computing for Engineers (CEE 618) is

- **Challenging**
- Formatless and fun
- Of new teaching style
- (Possibly) Research oriented
- **Fundamental learning** and **practical applications**
  - From know-how to know-why
- No preview necessary, but **lecture review** is critical plus **Homework**.
- Quiz tests replace one midterm.
Course Objective

... to introduce the *state-of-the-art high performance parallel computing* to solve engineering and science problems using basic mathematics, computational engineering, physics including fluid mechanics, and mathematics.

Target students: Engineering or science major graduate students (or undergraduate seniors) who do NOT have strong background in mathematics, physics, and computation/programming ...
Outline

1 Introduction

2 Background
   - Computation
   - Message Passing Interface (MPI)
   - Simulation Tools
   - Examples
     - Parallel Operation
     - Monte Carlo and \( \pi \) Calculation

3 SSH

4 UNIX/LINUX Basics

5 Editors

6 FORTRAN
Computation must be completed within a “reasonable” amount of time period.

1. Weather forecasting
2. Real time flight control
3. Daily data input and prediction (SWAT)
4. ...

What would be a possible solution for faster computation?

1. **Parallel (or distributed) computing**: MPI (Message Passing Interface)
2. **Shared memory process (SMP)**: OpenMP (Open Multiple Processing)
Two main objects are

1. **Speed-up**
2. **Scale-up**

What is the ultimate limitation of computational speedup?

- **Speed of light**
- Others: less electronic **resistance**, effective **cooling**, less **energy** consumption, optimal use of resources, and so forth
World Top 500 Supercomputer Rankings

Which site should you visit?

http://www.top500.org

Maui High Performance Computing Center is currently ranked $?^{th}$.

See your homework assignment.
What should we study for parallel computing?

Message Passing Interface (MPI)

What is it?

1. A program library, NOT a language
2. Called from **FORTRAN 77/90/03, C/C++**, and even Java and **Python**
3. **Most widely** used parallel library, but NOT a revolutionary way for parallel computation
4. A **collection** of the best features of (many) existing message-passing systems
Background

Message Passing Interface (MPI)

Quick Access

- MPI Forum
- MPI
  Standard 1.1
- MPI
  Standard 2.0
- MPICH
  Home Page
  ANL/MSU MPI implementation
- Download
  MPICH
  Free implementation of MPI
- MPICH2
  Home Page
  ANL MPI implementation
- Download
  MPICH2
  Free implementation of MPI
- MPI Home Page
- Remove

The Message Passing Interface (MPI) standard

What is MPI?

MPI is a library specification for message-passing, proposed as a standard by a broadly based committee of vendors, implementors, and users.

- The MPI standard is available.
- MPI was designed for high performance on both massively parallel machines and on workstation clusters.
- MPI is widely available, with both free available and vendor-supplied implementations.
- MPI was developed by a broadly based committee of vendors, implementors, and users.
- Information for implementors of MPI is available.
- Test Suites for MPI implementations are available.

How can I learn about MPI?

- Materials for learning MPI
- Papers discussing the design of MPI and its implementations
- Attend meetings on MPI: Euro PVM MPI 2008

What Libraries and applications are available in MPI?

A number of libraries and applications that use MPI are available.

Where is MPI going?

The MPI Forum has completed an effort to extend MPI. Information is available from the MPI Forum Home Page.

Figure: Mathematics and Computer Science Division, Argonne National Laboratory (http://www.mcs.anl.gov/research/projects/mpi/)

www.mcs.anl.gov/research/projects/mpi/
Regarding MPI and MPICH

- Where and when was MPI developed?
  1. In 1994, by researchers at Argonne National Laboratory and Mississippi State University
  2. They decided just a protocol of how each routine works, but did NOT make codes or libraries.

- OpenMPI (from open (free) source community)
  1. For Open Source High Performance Computing
  3. Used by many Top500 supercomputers

- What is MPICH?
  1. A portable implementation of MPI (a specification)
  2. Where is “CH” from?
Background

Message Passing Interface (MPI)

Figure: **CHameleon.** Because **MPI CH** is independent of operating systems such as Unix/Linux, Sun OS, Cray, IBM, Windows, Mac, ...
Programming Language (1)

Languages for Scientific and Engineering Computations

- **FORTRAN 2003/90(/77) = FORMula TRANslator**
  1. Fast access to memory
  2. Dynamic memory allocation
  3. C/C++ style of flexible coding
     
     *But, f77 is generally at least not slower than f90. Unified versions are available.*
  4. f03 (FORTRAN 2003) includes the Object-Oriented Programming (OOP) features (to be covered soon) for better data management.

- **C and C++**
  1. Large scale Object-Oriented Programming (OOP) using C++.
  2. Very low level control using C, but dangerous for numerical computation.
  3. Reversed indexes of \((i,j)\), starting with 0 (not 1).
Octave + gnuplot

1. A high-level language, primarily intended for numerical computations
2. A free clone of MATLAB, 99% compatible
3. No fancy toolboxes and supportive documentations

Others: Python, ADA, Ruby, Java, and so forth

1. Not mainstream languages for computation
2. But, recently Python is paid a close attention for scientific computation using SciPy (http://www.scipy.org/) and NumPy (http://www.numpy.org/). Kivy provides an excellent python libraries for cross-platform GUI applications on Linux, Windows, MacOSX and Android (http://kivy.org/).

Mac, Linux, and Windows versions are all available.
Other Tools

Text Editors
1. vi and vim (vi improved)
2. pico and nano
3. emacs

Compilers: f03/f90/f77/C/C++
1. Intel: ifort & icc
2. Portland Group
3. GNU

Programming Managing/Maintaining Tool
1. Make Utility

Queueing System (only for mainframe supercomputer)
1. Portable Batch System (PBS)
2. Load Sharing Facility (LSF)
3. Maui Scheduler
4. ...
Summation Trick

Calculated by *Carl Fredrick Gauss* when he was very young.

\[ S = 1 + 2 + 3 + \ldots + 98 + 99 + 100 = \sum_{i=1}^{100} i \]

\[ S = 100 + 99 + 98 + \ldots + 3 + 2 + 1 = \sum_{i=100}^{1} i \]

\[ 2S = 101 + 101 + 101 + \ldots + 101 + 101 + 101 = 100 \times 101 \]

\[ S = 5050 \]
Parallel Summation

Calculation using 10 processors (0–9)

\[ S = 1 + 2 + 3 + \ldots + 98 + 99 + 100 = \sum_{i=1}^{10} S_i = 5050 \]

\[ S_0 = 1 + 2 + 3 + \ldots + 8 + 9 + 10 = 55 \]

\[ S_1 = 11 + 12 + 13 + \ldots + 18 + 19 + 20 = \]

\[ S_2 = 21 + 22 + 23 + \ldots + 28 + 29 + 30 = \]

\[ S_3 = 31 + 32 + 33 + \ldots + 38 + 39 + 40 = \]

\[ S_4 = 41 + 42 + 43 + \ldots + 48 + 49 + 50 = \]

\[ S_5 = 51 + 52 + 53 + \ldots + 58 + 59 + 60 = \]

\[ S_6 = 61 + 62 + 63 + \ldots + 68 + 69 + 70 = \]

\[ S_7 = 71 + 72 + 73 + \ldots + 78 + 79 + 80 = \]

\[ S_8 = 81 + 82 + 83 + \ldots + 88 + 89 + 90 = \]

\[ S_9 = 91 + 92 + 93 + \ldots + 98 + 99 + 100 = \]
What is Monte Carlo?

1. **A City Name**: The wealthiest of Monaco’s various administrative areas, but not a capital. Monte Carlo is known for its casinos, **gambling**, glamour, and for sightings of famous people.

2. A Monte Carlo method is a computational **algorithm** that relies on **repeated random sampling** to compute its results. Monte Carlo methods are often used when simulating physical and mathematical, **plus engineering** systems.
Monte Carlo methods were originally practiced under more generic names such as “statistical sampling”.

The name “Monte Carlo” was popularized by physics researchers Stanislaw Ulam, Enrico Fermi, John von Neumann, and Nicholas Metropolis among others; the name is a reference to a famous casino in Monaco which Ulam’s uncle would borrow money to gamble at.

The use of randomness and the repetitive nature of the process are analogous to the activities conducted at a casino.

A standard Monte Carlo algorithm is called Metropolis Monte Carlo (MMC).
What is π value?

3.1415926535897932384626433832795028841971693993751058209749445
923078164062862089986280348253421170679821480865132823066470938
446095505822317253594081284811174502841027019385211055596446229
489549303819644288109756659334461284756482337867831652712019091
456485669234603486104543266482133936072602491412737245870066063
155881748815209209628292540917153643678925903600113305305488204
665213841469519415116094330572703657595919530921861173819326117
931051185480744623799627495673518857527248912279381830119491298
336733624406566430860213949463952247371907021798609437027705392
171762931767523846748184676694051320005681271452635608277857713
427577896091736371787214684409012249534301465495853710507922796
892589235420199561121290219608640344181598136297747713099605187
072113499999983729780499510597317328160963185950244594553469083
026425223082533446850352619311881710100031378387528865875332083
814206171776691473035982534904287554687311595628638823537875937
519577818577805321712268066130019278766111959092164201989380952
“Hello, MC” program

How to calculate $\pi$ value using Monte Carlo

$A_{\text{square}} = 4L^2$, \hspace{1cm} $A_{\text{circle}} = \pi L^2$, \hspace{1cm} $\frac{A_{\text{circle}}}{A_{\text{square}}} = \frac{\pi}{4}$

$\pi = 4 \frac{A_{\text{circle}}}{A_{\text{square}}} = \left[ 4 \frac{N_{\text{circle}}}{N_{\text{square}}} \right]$

where $N_{\text{circle}}$ and $N_{\text{square}}$ are the numbers of dots in the circle and square, respectively.
Selection Criteria due to Symmetry

The Entire Range

- Within the square: 
  \[-1 < \frac{x}{L}, \frac{y}{L} < +1\]
- Within the circle: 
  \[\left(\frac{x}{L}\right)^2 + \left(\frac{y}{L}\right)^2 < 1\]

The First Quarter

- Within the quarter-square: 
  \[0 < \frac{x}{L}, \frac{y}{L} < 1\]
- Within the quarter-circle: 
  \[\left(\frac{x}{L}\right)^2 + \left(\frac{y}{L}\right)^2 < 1\]

\[\pi = 4 \frac{A_{\text{circle}}}{A_{\text{square}}} = 4 \frac{\frac{1}{4} N_{\text{circle}}}{\frac{1}{4} N_{\text{square}}}\]

*Use symmetry for easier and faster calculations!*
Programming Algorithm

Pseudo Code

1. Decide a sampling number \( N \) (i.e., 10,000) and set \( N_{hit} = 0 \)

2. For \( i = 1, N \)
   - (1) Generate \( 0 < X, Y < 1 \), where \( X \) and \( Y \) are random numbers.
   - (2) Calculate \( R = X^2 + Y^2 \). What are min. and max. of \( R \)?
   - (3) If \( R \leq 1 \), then \( N_{hit} = N_{hit} + 1 \).

3. \( \pi = \frac{4N_{hit}}{N} \) where \( N_{hit} \) and \( N \) correspond to areas of circle \( (A_{circle}) \) and square \( (A_{square}) \), respectively.

4. Compare it with the exact value: \( \pi = \cos(-1) \), where \( \cos \) is the arc cosine function.

This procedure is called Importance Sampling or Statistical Sampling in terms of the probability that a dot hits the circular target. This method can be extensively used for complex geometries and sophisticated scenarios, especially of multiple dimensions.
! Computing PI using Monte Carlo
program calcpi
implicit none
integer :: i, n, id, nhit
real    :: x, y, r, pi, pi_act, rerror, ran1
write(*,*) 'Choose n: '
read(*,*) n
id = -5
nhit = 0
do i=1, n
  x = ran1(id)
y = ran1(id)
  ! Calculate radius
  ! NOTE: no need to waste computation time on square root
  ! since we are comparing to 1.
  r = x**2.0 + y**2.0
  if (r<1.0) then
    nhit=nhit+1
  endif
endo

! Find the ratio of points in the circle to the total points used
write(*,*) n, nhit
pi = 4.0 * (real(nhit)/real(n))

! Calcualte Actual Value of Pi and % Error
pi_act = acos(-1.0)
  rerror = abs((pi_act - pi)/ pi_act)
write(*,printstring) n, pi, pi_act, rerror
!
end program
Compile and Run “pi.f90”

**Simple way**

Using shell commands
- `ifort pi.f90 -o pi.x`
- `.\pi.x`

**Complex way**

Using Makefile
- `make`
- `make run`

Contents of Makefile
- `all: ifort pi90.f -o pi90.x`
- `run: ./pi90.x`
Outline

1 Introduction

2 Background
   ● Computation
   ● Message Passing Interface (MPI)
   ● Simulation Tools
   ● Examples
      ● Parallel Operation
      ● Monte Carlo and $\pi$ Calculation

3 SSH

4 UNIX/LINUX Basics

5 Editors

6 FORTRAN
Remote Login using ssh: temporarily

1. Use program “SSH Secure Client Shell” to login fractal.eng.hawaii.edu.

2. Click “Quick Connect” and input the following information
   1. Host Name: fractal.eng.hawaii.edu
   2. User Name: guest
   3. Password: guest
   4. Port Number: 22
   5. Authentication Method: Password

3. Go to a directory of your UH ID:
   $ cd youruhid
Remote Login using ssh: next class

1. Use program “SSH Secure Client Shell” to login fractal.eng.hawaii.edu.

2. Click “Quick Connect” and input the following information
   1. Host Name: fractal.eng.hawaii.edu
   2. User Name: youruhid
   3. Port Number: 22
   4. Authentication Method: Password

3. Change your password
   1. After you logged in, prepare a new password; type and enter the following command to change the password to your own
      » passwd

4. The first work: a simple parallel simulation using 16 processors
   1. Type and enter: “cd Test” and “ls -l”
   2. Type and enter: “make”, “make run”, and “qstat”
SSH Tutorials

Visit URL\(^1\) for

- Comparative evaluation
- SSH Client for Windows, Putty, OpenSSH
- Tutorial videos

\(^1\)http://albertsk.org/tutorial-videos-2/programming/
Outline

1 Introduction

2 Background
   • Computation
   • Message Passing Interface (MPI)
   • Simulation Tools
   • Examples
     • Parallel Operation
     • Monte Carlo and π Calculation

3 SSH

4 UNIX/LINUX Basics

5 Editors

6 FORTRAN
Basic Unix Commands for Directories

After login, `cd youruhid`

1. To list files and directories within the current directory
   - Command name: `ls`
   - Type and enter: “`ls`”, “`ls -l`”, and “`ls -l`”
   - Try: “`man ls`”. To stop man (manual) page, use “`q`”.
   - Note: Do NOT delete any files or directories beginning with “`.`” (a dot) because those are generated by OS, i.e., system files.

2. To make a new directory
   - Command name: `mkdir`
   - Type and enter: “`mkdir cee618`”
   - Note: It is NOT recommended to use spaces for names of files and directories in Unix/Linux systems, although it is possible.
Basic Unix Commands for Directories

3. To go to (change) the directory
   - Command name: `cd`
   - Type and enter: “cd cee618”
   - Try “cd .” (A dot indicates the current directory)
   - Try “cd ..” (Double dots indicate the parent directory)
   - Try `pwd` to see present working directory
   - Try “cd cee618” again
   - Make a directory ‘lecture01’: “mkdir lecture01”
   - Make a directory ‘parallel’: “mkdir parallel”

4. To remove a directory
   - Command name: `rmdir`
   - Type and enter: “rmdir parallel”
   - This command will generate an error message unless “parallel” directory is completely empty.
   - To remove a directory (saying mydesk) which contains files and subdirectories: “rm -rf mydesk”.
Basic Unix Commands for Files

1. To show the first part of a file, just as much as will fit on the screen. Just hit the space bar to see more or “q” to quit.
   - Command name: `more`
   - Type and enter: “`more pi.f90`” (in the directory where pi.f90 exists.)

2. To copy a file to another.
   - Command name: `cp`
   - Type and enter: “`cp pi.f90 pi2.f90`” (in the directory where pi.f90 exists.)

3. To remove (or delete) a file.
   - Command name: `rm`
   - Type and enter: “`rm pi2.f90`” (in the directory where pi2.f90 exists.)

4. To move (or rename) a file.
   - Command name: `mv`
   - Type and enter: “`mv pi.f90 pi2.f90`” (in the directory where pi.f90 exists.)
5. To compare two files.
   - Command name: `diff`
   - Type and enter: "diff pi.f90 pi2.f90" (in the directory where pi.f90 and pi2.f90 exist.)

6. To see how many lines, words, and characters there are in a file
   - Command name: `wc`
   - Type and enter: "wc pi.f90" (in the directory where pi.f90 exists.)

7. To edit a text file.
   - Command name: `nano` (or `vi`)
   - Type and enter: "vi pi.f90" (in the directory where pi.f90 exists.)
- Visit URL\(^2\) for UNIX/LINUX command video tutorials
- Refer to “UNIX command summary”

\(^2\)http://albertsk.org/tutorial-videos-2/programming/
Outline

1 Introduction

2 Background
   - Computation
   - Message Passing Interface (MPI)
   - Simulation Tools
   - Examples
     - Parallel Operation
     - Monte Carlo and $\pi$ Calculation

3 SSH

4 UNIX/LINUX Basics

5 Editors

6 FORTRAN
vi (vim) editor or nano

- Visit URL\(^3\) for UNIX/LINUX command video tutorials
- Refer to “How to use the vi editor”
- Refer to “The nano Text Editor”

\(^3\)http://albertsk.wordpress.com/tutorial-videos-2/programming/
Outline

1 Introduction

2 Background
   • Computation
   • Message Passing Interface (MPI)
   • Simulation Tools
   • Examples
     • Parallel Operation
     • Monte Carlo and π Calculation

3 SSH

4 UNIX/LINUX Basics

5 Editors

6 FORTRAN
FORTRAN: Resources

Visit URL⁴ for FORTRAN video tutorials

⁴http://albertsk.org/tutorial-videos-2/programming/