HPC Overview & Workshop Facility

2024 NSF CyberTraining Workshop

Jan. 8, 2024 – Jan. 19, 2024

Clarkson University

Topics of this Lecture

- Workshop Overview
- HPC Overview
- Laws in Parallel Computing
- Workshop Facility

Workshop Overview

- Virtual Workshop: Zoom Meetings
- Schedule: Week 1: Lectures & Labs; Week 2: Projects
- Time: Jan. 8 – Jan. 19, 2024
- Workshop Website: cybertraining.clarkson.edu
- **This workshop is supported by NSF Award #218079**
- Note: The lecture slides in this course are adapted from Profs. Wellein Hager (FAU) and Wei Zhang (VCU)'s course slides as well as many other training materials from internet (e.g., NVIDIA, OpenMPI, PETSc, C Programming Essentials, etc). Also, Profs. Chunlei Liang, Daqing Hou, Ming-Cheng Cheng (Clarkson) provides their slides for this workshop.

Scope of the Workshop

Hardware coverage:

- Multi-core CPU, Many-core General Purpose GPU (GPGPU)
- Shared-memory nodes, Distributed-memory nodes

Identify basic hardware concepts and how to efficiently use them:

- Distributed-memory Parallel Programming →MPI
- General Purpose GPU Programming →NVIDIA CUDA
- Scalable Mathematics Interface →PETSc, SLEPc, FEniCS

Computational Methodology:

- Proper Orthogonal Decomposition (POD)

Engineering Problems:

- Thermal Behaviors on CPUs, Quantum Physics

Faculty & TAs

Instructor Team: Prof. Daqing Hou (ECE), Prof. Yu Liu (ECE), Prof. Ming-Cheng Cheng (ECE), Prof. Guangming Yao (Math)
TAs: Martin Veresko (M.S. student), Ahmad Suleiman (Ph.D. student)

Topics of this Lecture

- Workshop Overview
- HPC Overview
- Laws in Parallel Computing
- Workshop Facility

Supercomputer - Definition

- *"Supercomputer (sc) is a computer that is only one generation behind what large-scale users want."*
  Neil Lincoln, architect for the CDC Cyber 205 and others

- **A supercomputer does not fit under the desktop! (and you can not plug it into a standard power line) – e.g., cooling issue, power supply issue, green computing expectation …**

- **Assume:**
  - Computer is being used for numerical simulation
  - Compute power of a system is measured by Floating Point Operations (MULT, ADD) for a specific numeric benchmark

SC defeat a World Champion!

Garry Kasparov

Deep Blue

In February 1996, IBM's Deep Blue defeated grandmaster Garry Kasparov. It was then assigned to predict the weather in Atlanta, Georgia, during the 1996 Summer Olympic Games

Computer Arch. & Prog. Models

- **Modern processors:**
  - Single core: Superscalar (Pipelining), VLIW (Very Long Instruction Word)
  - Memory Hierarchy (Caches)
  - Multi-core CPU
  - Many-core GPU
- **Parallel computers: Shared-memory**
  - Shared-memory system architectures: UMA (Uniform Memory Access), NUMA (Non-uniform Memory Access)
  - pthread (POSIX thread)
- **Parallel computers: Distributed-memory**
  - MPI (Message Passing Interface)
- **Parallel computers: General Purpose GPU**
  - NVIDIA CUDA

TOP500

- **Top 500: Survey of the 500 most powerful supercomputers**
  - Solve large dense system of linear equations: e.g., $\boldsymbol{A}\,\boldsymbol{x} = b$ (“LINPACK“ benchmark), solved by LU Decomposition
  - Published twice a year
  - Established in 1993
  - Since Nov. 2018 (Summit/US): 143,500 TFlop/s (TOP1)
  - Performance increase: 81% from 1993 – 2017

- **Performance measure: MFlop/s, GFlop/s, TFlop/s, PFlop/s, EFlop/s**
  - Number of FLOATING POINT operations per second
  - FLOATING POINT operations: double precision (64 bit) Add & Mult ops

- 10^6: MFlop/s; 10^9: GFlop/s; 10^{12}: TFlop/s; 10^{15}: PFlop/s ; 10^{18}: EFlop/s

LINPACK

- **What is the Linpack benchmark**
  - A measure of a computer's floating-point rate of execution
  - Run a computer program that solves a dense system of linear equations
  - The paper "The LINPACK Benchmark: Past, Present, and Future" by **Jack Dongarra**, Piotr Luszczek, and Antoine Petitet

- **LINPACK benchmark implementation**
  - Number the actual implementation of the program can diverge, and HPL is a portable implementation of HPLinpack that was written in C used for TOP500 listing (http://www.netlib.org/benchmark/linpackc).
    - HPL generates a linear system of equations of order n and solves it using LU decomposition.

LU Decomposition - Definition

- LU Decomposition is a critical method to solve a set of linear equations

- For most non-singular matrix A that one could conduct Naïve Gauss Elimination forward elimination steps, one can always write it as:

  $A = L\ U$

  where

  L = lower triangular matrix
  U = upper triangular matrix

LU Decomposition - Application

- **How can this be used?**

Given $A\,x = \mathrm{b}$ => $\mathrm{LUx} = \mathrm{b}$ => $\mathrm{L(Ux)=b}$

1. Decompose A into L (lower triangular matrix) and U (upper triangular matrix)
2. Solve $L\,\mathrm{y} = \mathrm{b}$ for y
3. Solve $U\,x = y$ for x

LU Decomposition - Example

- **We factorize the following 2x2 matrix:**

$$\begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}.$$

$$\begin{aligned} l_{11} \cdot u_{11} + 0 \cdot 0 &= 4 \\ l_{11} \cdot u_{12} + 0 \cdot u_{22} &= 3 \\ l_{21} \cdot u_{11} + l_{22} \cdot 0 &= 6 \\ l_{21} \cdot u_{12} + l_{22} \cdot u_{22} &= 3. \end{aligned}$$

- **We can conveniently require the lower triangular matrix L to be a unit triangular matrix (i.e. set all the entries of its main diagonal to ones), then:**

$$\begin{aligned} l_{21} &= 1.5 \\ u_{11} &= 4 \\ u_{12} &= 3 \\ u_{22} &= -1.5 \end{aligned}$$

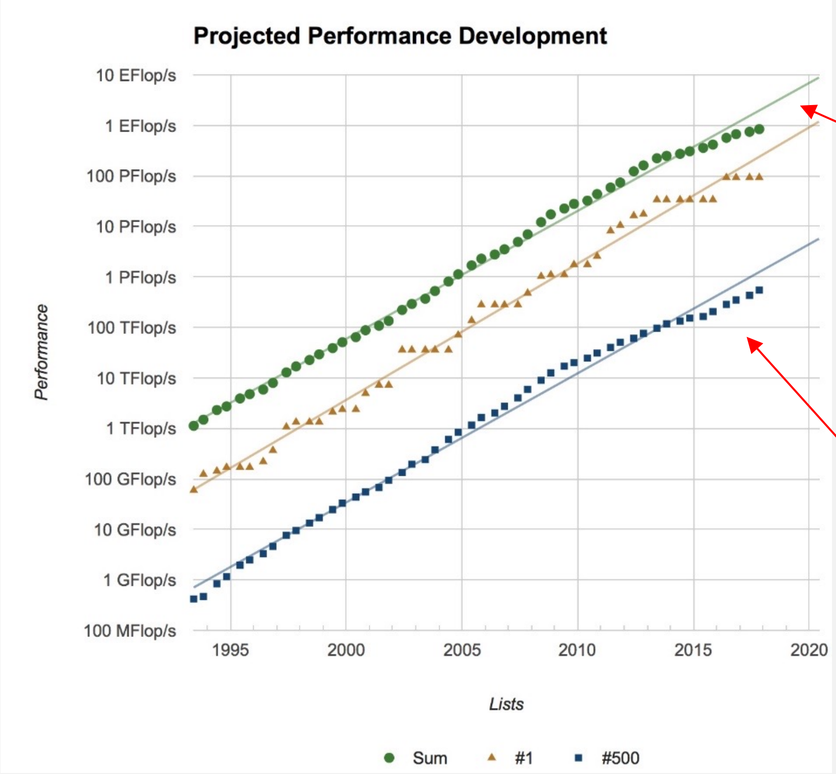
$$\rightarrow \begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 0 & -1.5 \end{bmatrix}.$$

TOP10 as of November 2020

Rank	System	Cores	(TFlop/s)	(TFlop/s)	(kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646

GPU merges into the mainstream of high-performance computing !

Performance Trend & Projection

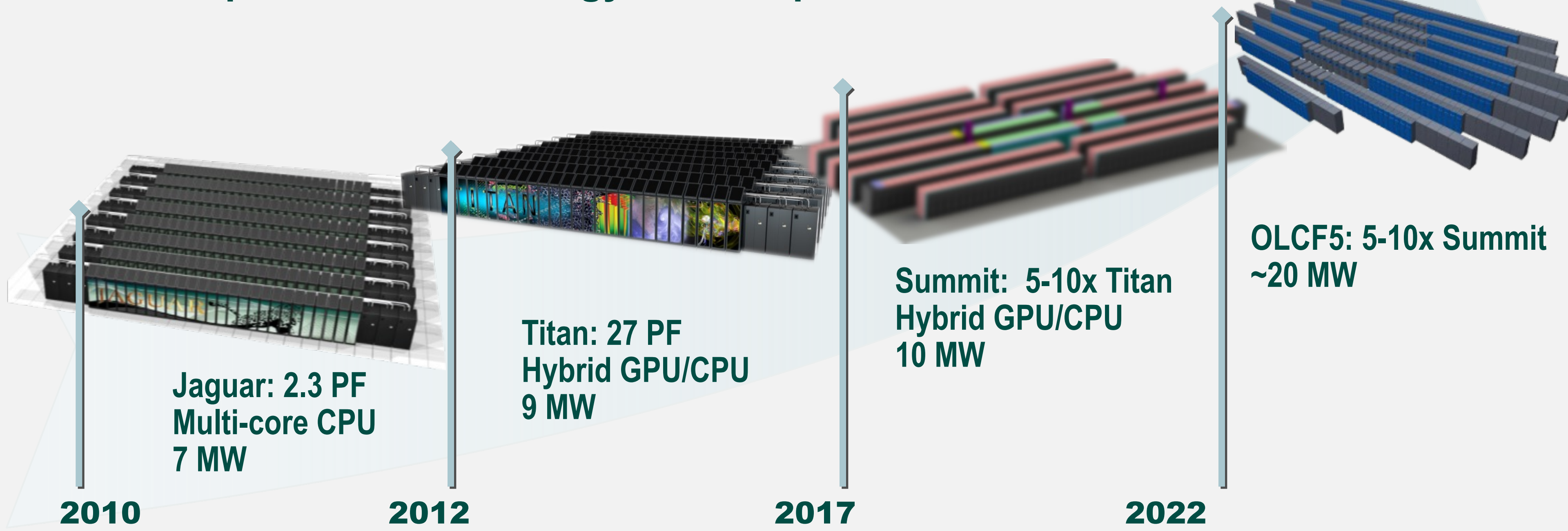


ExaFlop/s machine in the 20’s decade?

Basic trend:
Slope changes → performance increase slows down

Supercomputer - Issues

- Energy consumption is major issue for centers and users!
- Example: ORNL SC Energy Consumption Trend

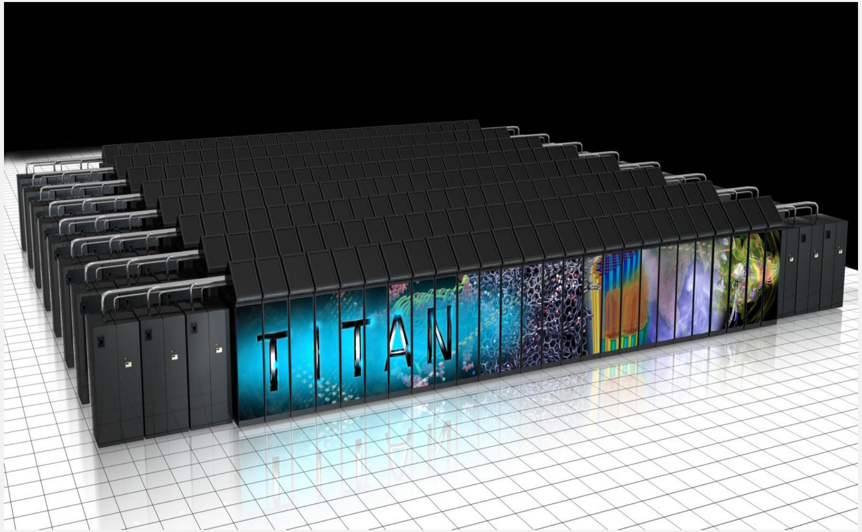


ORNL - Titan

Vendors: Cray™ / NVIDIA™

- 27 PF peak

- 18,688 Compute nodes, each with
  - 1.45 TF peak
  - NVIDIA Kepler™ GPU - 1,311 GF
    - 6 GB GDDR5 memory
  - AMD Opteron™- 141 GF
    - 32 GB DDR3 memory
  - PCIe2 link between GPU and CPU
- Cray Gemini 3-D Torus Interconnect system



https://www.olcf.ornl.gov/olcf-resources/compute-systems/titan/

ORNL - Summit

Vendors: IBM™ / NVIDIA™

- Number of Nodes. 4,608.
- Node performance. 42 TF.
- Memory per Node. 512 GB DDR4 + 96 GB HBM2.
- NV memory per Node. 1600 GB.
- Total System Memory. >10 PB DDR4 + HBM2 + Non-volatile.
- Processors. 2 IBM POWER9™ 9,216 CPUs. 6 NVIDIA Volta™ 27,648 GPUs.
- File System. 250 PB, 2.5 TB/s, GPFS™

https://www.olcf.ornl.gov/summit/

LLNL - Sierra

Vendors: IBM™ / NVIDIA™

- Number of Nodes. 4,320.
- Node performance. 42 TF.
- Memory per Node. 512 GB DDR4 + 96 GB HBM2.
- Total System Memory. >1.38PB
- Total Cores. 190,080
- Processor Architecture. IBM POWER9™ CPU, NVIDIA Volta™ GPU.
- Operation System. RHEL

https://computation.llnl.gov/computers/sierra

Topics of this Lecture

- Workshop Overview
- HPC Overview
- Laws in Parallel Computing
- Workshop Facility

Moore's Law

The experts look ahead

Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore
Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many new areas.

Integrated circuits will lead to such wonders as home computers—or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today.

But the biggest potential lies in the production of large systems. In telephone communications, integrated circuits in digital filters will separate channels on multiplex equipment. Integrated circuits will also switch telephone circuits and perform data processing.

machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the construction of larger processing units. Machines similar to those in existence today will be built at lower costs and with faster turn-around.

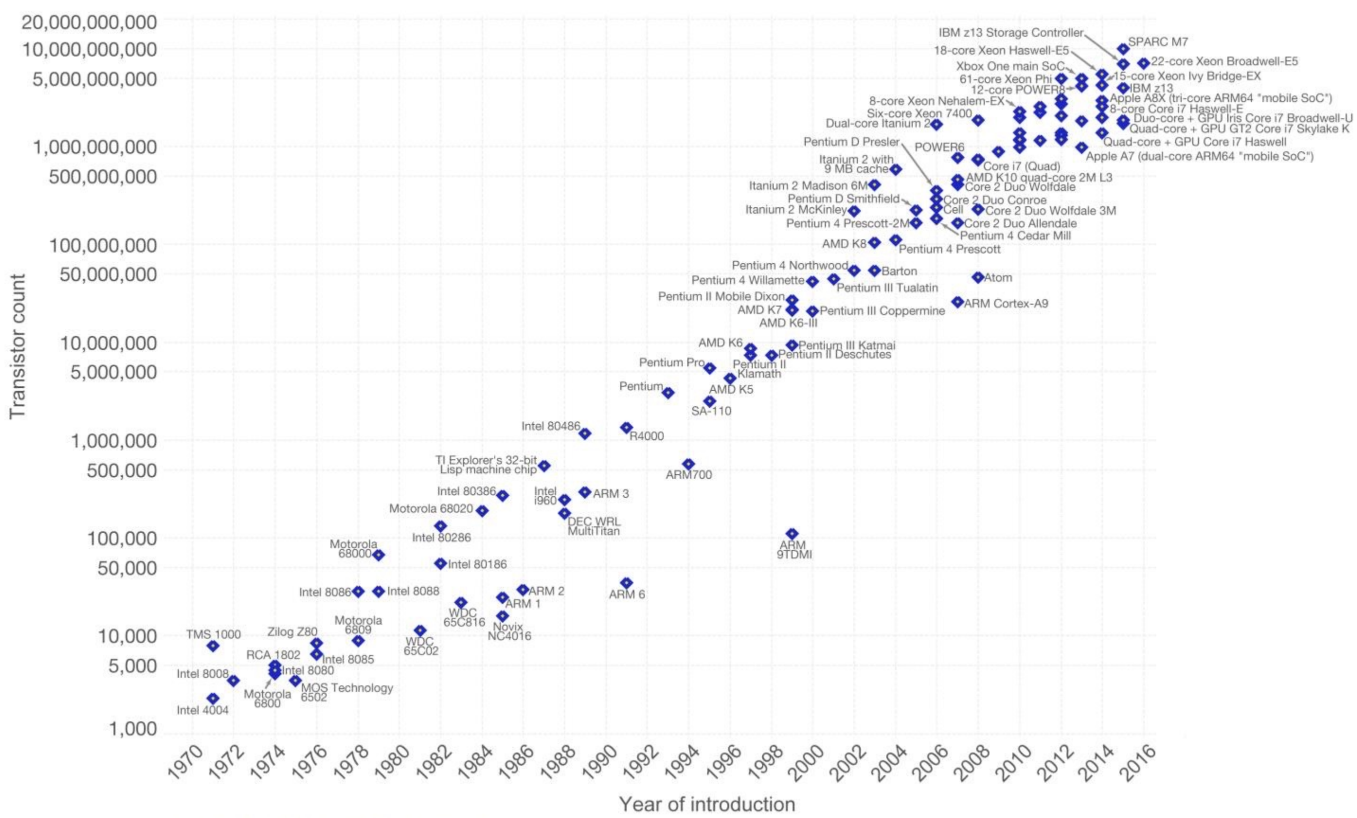
Present and future

By integrated electronics, I mean all the various technologies which are referred to as microelectronics today as well as any additional ones that result in electronics functions supplied to the user as irreducible units. These technologies were first investigated in the late 1950's. The object was to miniaturize electronics equipment to include increasingly complex electronic functions in limited space with

Electronics

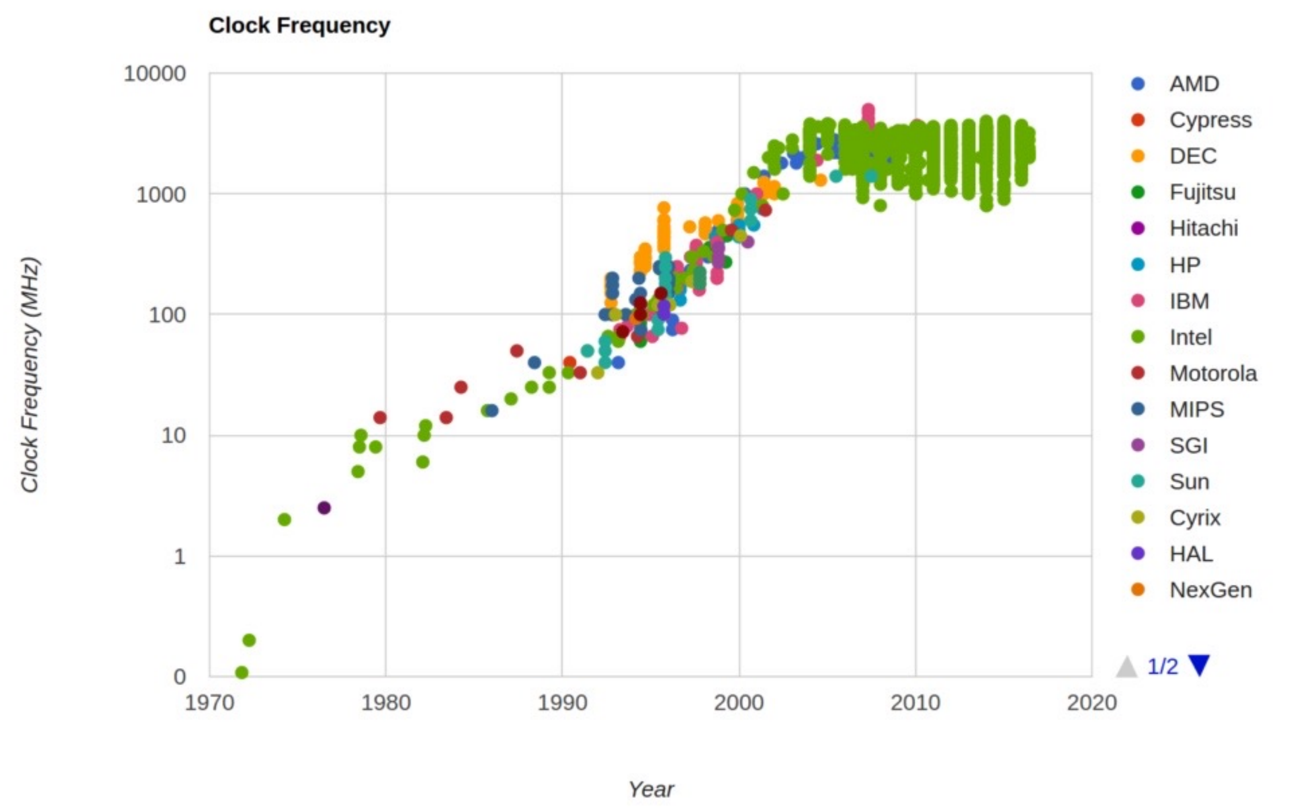
19 April 1965

Moore's Law Continue – or it does not



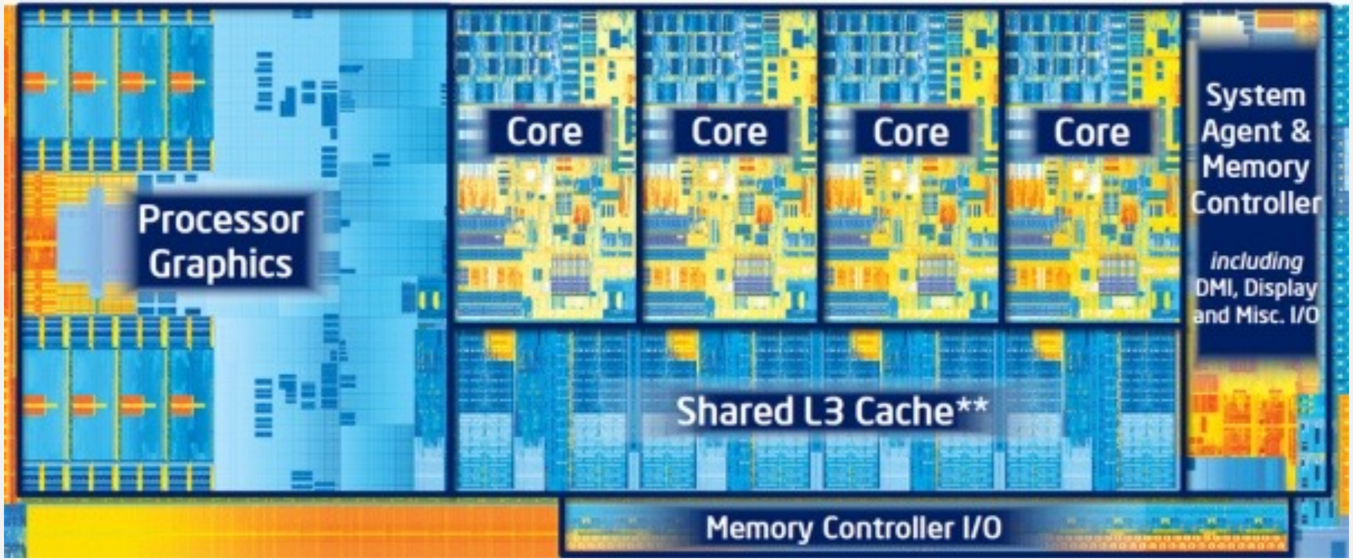
Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Clock Speeds have statured



New Solution – Multi-core

Intel Core i5-3470 Ivy Bridge Processor:



Amdahl's Law

- Speedup= $time_{without\ enhancement}$ / $time_{with\ enhancement}$

- Suppose an enhancement speeds up a fraction f of a task by a factor of S

  $$time_{new} = time_{old} \cdot ((1-f) + f/S)$$
  $$S_{overall} = 1 / ((1-f) + f/S)$$

Question?

- A program runs in 100 seconds on a machine, with multiply operations responsible for 80 seconds of this time. How much do I have to improve the speed of multiplication if I want my program to run 5 times faster ?

$$20 = 100 \cdot ((1-0.8) + 0.8/S)$$

$$S = ?$$

Amdahl's Law

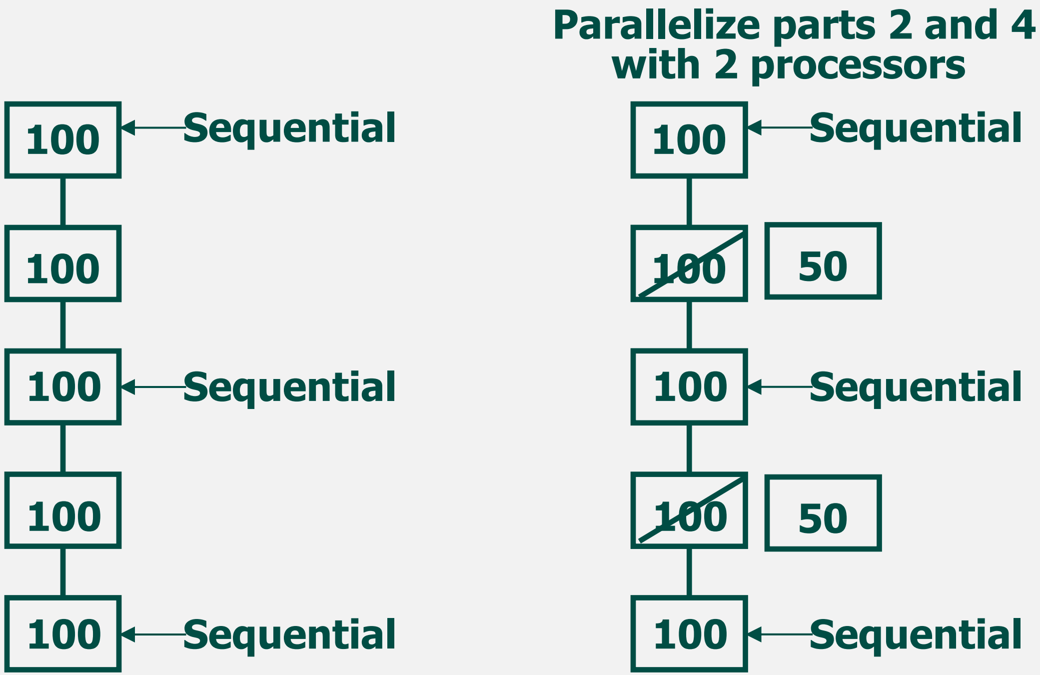
- Begins with Simple Software Assumption (Limit Arg.)
  - Fraction F of execution time perfectly parallelizable
  - No Overhead for
    - Scheduling
    - Communication
    - Synchronization, etc.
  - Fraction 1 – F Completely Serial
- Time on 1 core = (1 – F) / 1 + F / 1 = 1
- Time on N cores = (1 – F) / 1 + F / N

Amdahl's Law

$$\text{Amdahl's Speedup} = \frac{1}{\frac{1 - F}{1} + \frac{F}{N}}$$

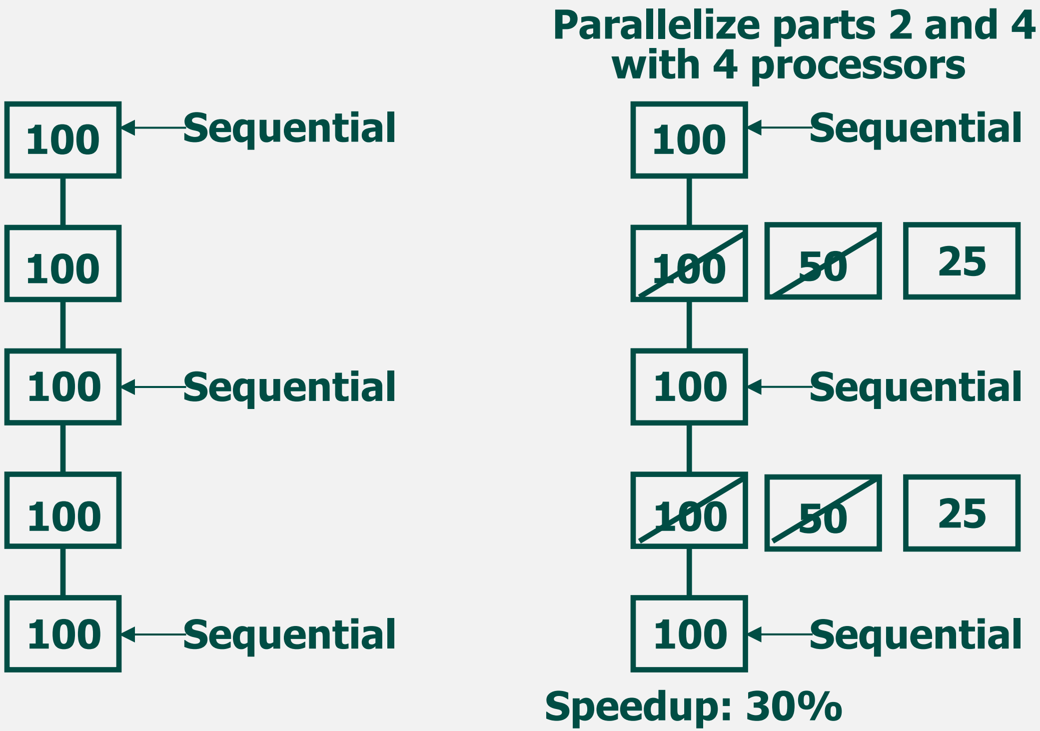
- For mainframes, Amdahl expected 1 - F = 35%
  - For a 4-processor speedup = 2
  - For infinite-processor speedup < 3
  - Therefore, stay with mainframes with one/few processors

Amdahl’s Law



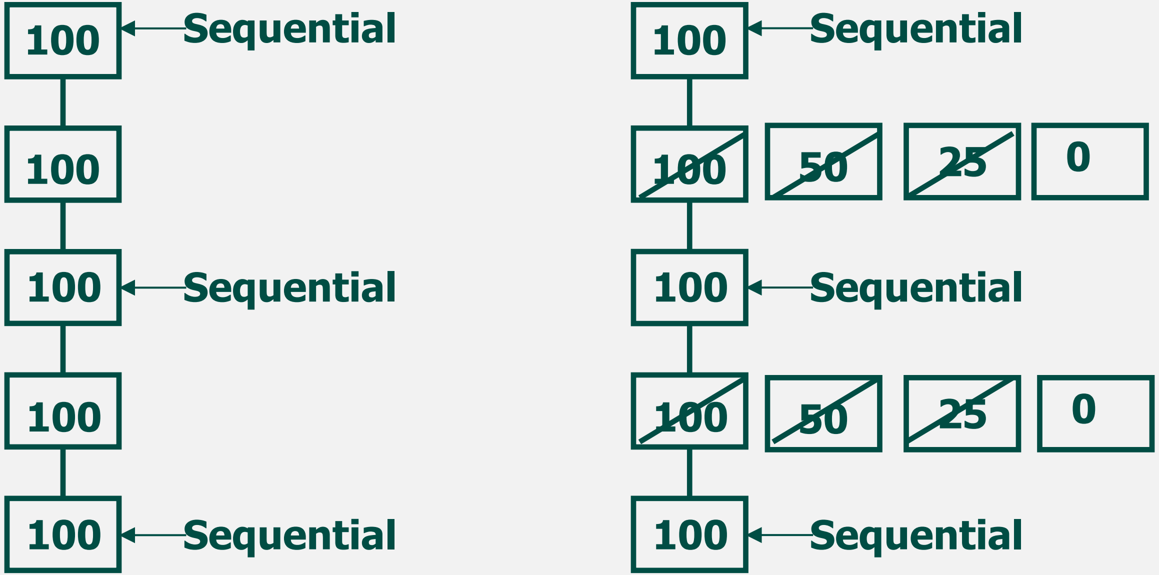
Speedup: 20%

Amdahl's Law (cont'd)



Amdahl's Law (cont'd)

Parallelize parts 2 and 4 with <u>infinite</u> processors



Speedup: only 40%

Multi-core doesn't look very appealing!

Gustafson's Law

- Fix execution of on a single processor as
  - $s + p$ = serial part + parallelizable part **= 1**
- S(n) = (s +p)/(s + p/n)

  = 1/(s + (1 – s)/n) = Amdahl's law

- Now let, W is execution time on a single core computer, and W(s) is that on a parallel computer, with p = parallel part.
  - W = (1-p)W + pW
  - W(n) = (1-p)W + npW
  - $S_s(n)$ = W(n)/W = 1 - p + np

  n: the number of cores

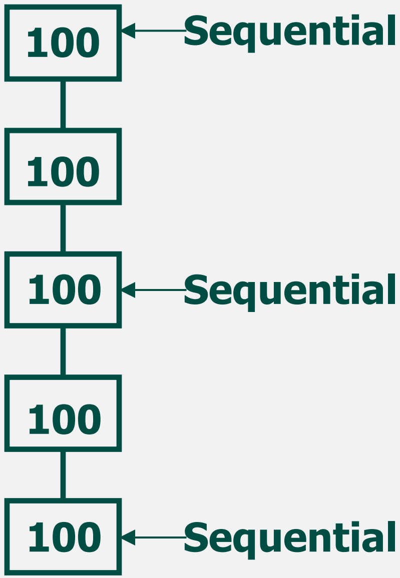
  p: ratio of the time spent in the parallel portion of the program versus the total execution time

More on Gustafson’s Law

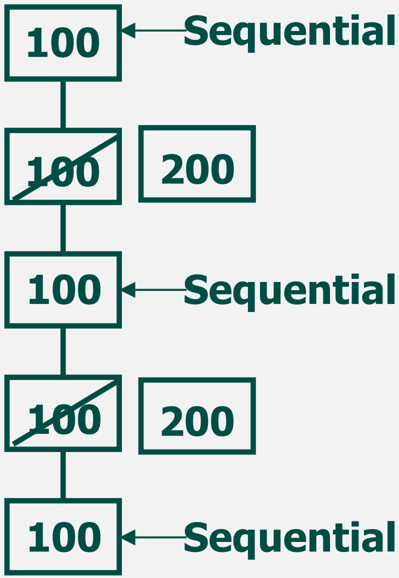
- Derived by fixing the parallel execution time (Amdahl fixed the problem size -> fixed serial execution time)
  - For many practical situations, Gustafson’s law makes more sense
    - Have a bigger computer, solve a bigger problem.
- Amdahl’s law turns out to be too conservative for high-performance computing.

Gustafson's Law (cont'd)

Boxes contain units of work now!



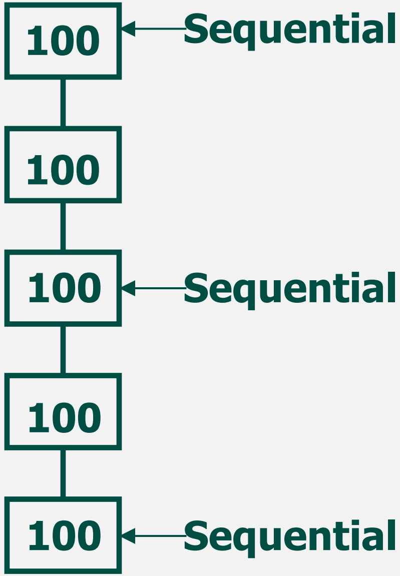
500 units of time, but 700 units of work!



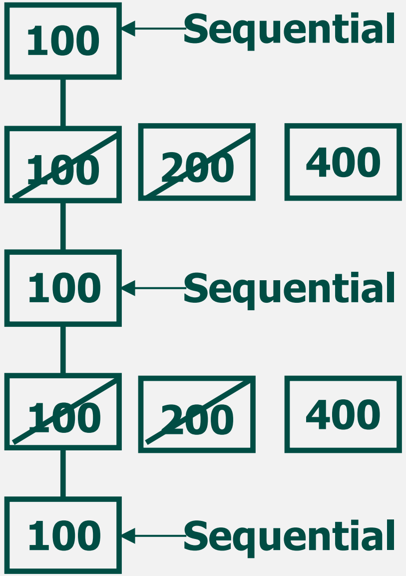
Speedup: 40%

Gustafson's Law (cont'd)

Boxes contain units of work now!



500 units of time, but 1100 units of work!



Speedup: 120%

Gustafson Law (cont'd)

- Gustafson found important observation
  - As processors grow, people scale problem size
  - Serial bottlenecks do not grow with problem size
- Increasing processors gives linear speedup
  - 20 processors roughly twice as fast as 10
- Multi-core computing is promising!

Reference: **http://www.scl.ameslab.gov/Publications/Gus/AmdahlsLaw/Amdahls.html**

Topics of this Lecture

- Workshop Overview
- HPC Overview
- Laws in Parallel Computing
- Workshop Facility

Computing Facility

- Servers: *cybertraining.clarkson.edu*

  & *reu-hpc.clarkson.edu (for the GPU lecture)*
- Accounts have been created for you
- Your account name is as same as your Clarkson account
- Default password is abc123
- If you try to login this server outside of Clarkson campus, you will need to use Clarkson's VPN firstly.

Our Server

- **DELL T7920 Server Tower:**
  - CPU: 2 x Intel Gold 6130 Xeon CPU (2.1GHz 16 core): total 32 cores
  - Memory: 512GB
  - OS: Ubuntu Linux 20.04
  - Hard disk: 12TB

- **This server can be used for your in-class practice and labs**
- **ssh cybertraining.clarkson.edu**

- **GPU programming lecture will be on a different Linux server**
- **ssh reu-hpc.clarkson.edu**

Clarkson ACRES Cluster

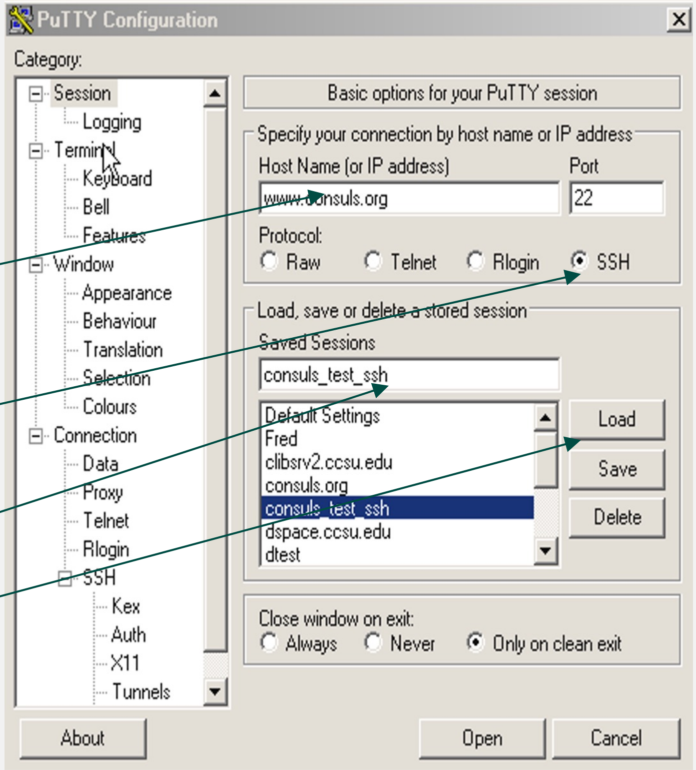
- **ACRES:**
  - Account name and password are as same as your Clarkson email's
  - ssh acres.clarkson.edu
  - https://sites.clarkson.edu/acres/
  - Sponsored by NSF Award (#1925596)
  - Managed by slurm (a quick guide can be found at: https://support.ceci-hpc.be/doc/_contents/QuickStart/SubmittingJobs/SlurmTutorial.html)

SLURM

- **Useful Commands:**
  - sinfo
  - sbatch
  - squeue
  - squeue –j
  - scancel

Remote Login by SSH - Putty

Add the session information:



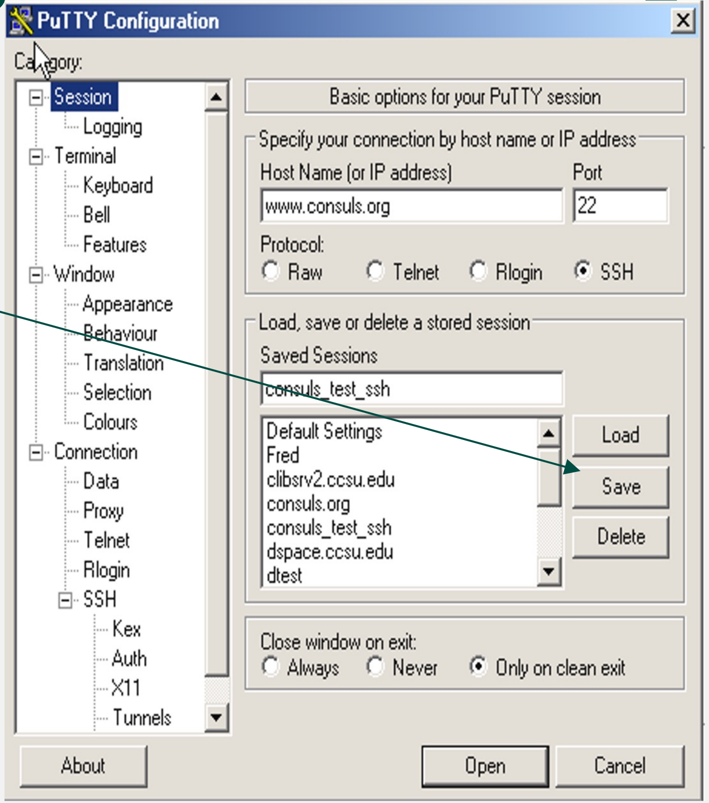
This is where you put in the address (e.g., cybertraining.clarkson.edu).

Make sure you select ssh

Give it a name and click save

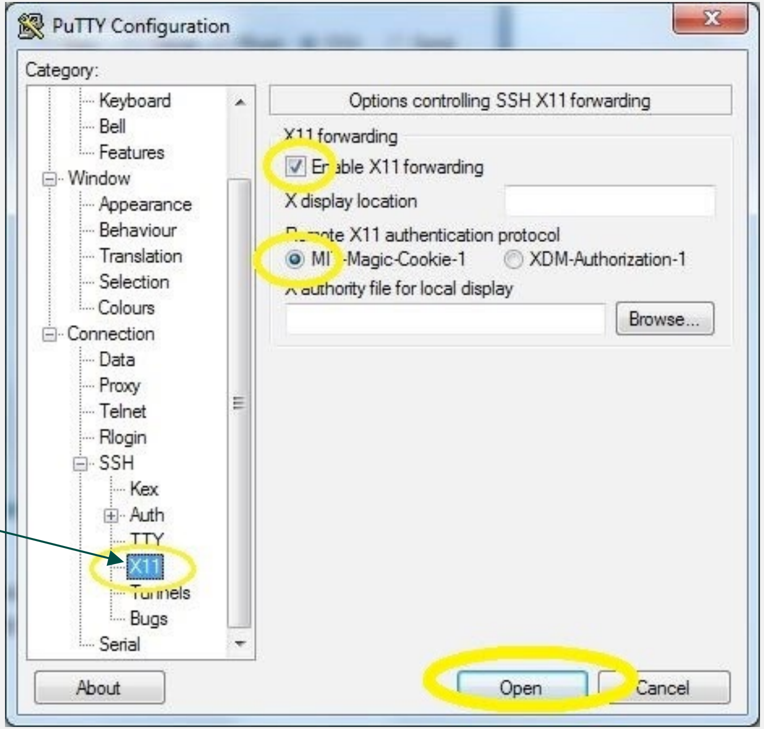
Remote Login by SSH - Putty

- Make sure that after you make any changes you click on session and save.
- Otherwise nothing will be saved and you will have to do it all over again.



Xwindow By SSH - XMing

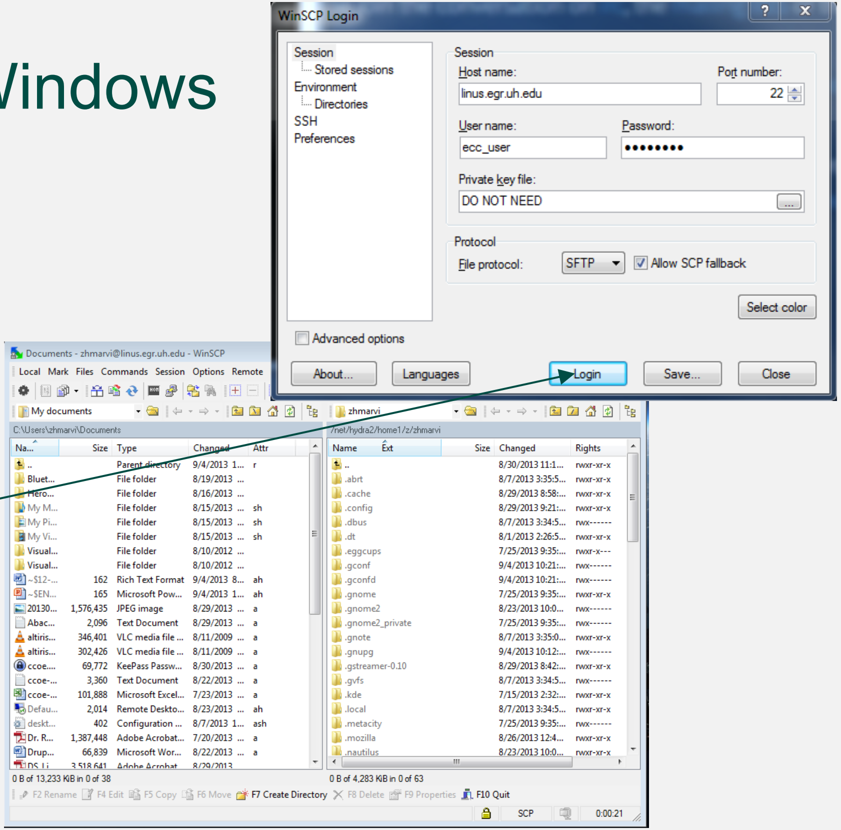
- Xming is a free X-server application for windows
- Run Xming, runs in the background. No initial setup is required.
- While Xming is running, start Putty session in X11 enabled mode



File Transfer By SSH - WinSCP

- WinSCP: Transfer files from Windows to Linux and vice versa

- Login:
  - Open WinSCP program
  - Enter user credentials, click login



MacOS - Terminal

- MacOS: Go->Utilities->Terminal
- Login:

  ssh hostname/IP address