COMS4040A & COMS7045A: High Performance Computing & Scientific Data Management Introduction

Hairong Wang

School of Computer Science & Applied Mathematics
University of the Witwatersrand, Johannesburg

WITS UNIVERSITY

Contents

WITS UNIVERSITY

Outline

WITS UNIVERSITY

Course introduction

- Lecture - a double session (Monday 10:15 – 12:00) per week. Lecture slides are provided each week.
- Exercise - no submissions are required usually unless otherwise requested
- Tut/lab session - Monday 12:30 – 13:15
- Course materials available on Ulwazi course site

Course introduction cont.

- You need some C/C++ programming and algorithm and analysis background to follow the lectures and completing the assessment tasks.
- You need to familiarize yourselves with basic Linux command line and remote access via SSH to MSL cluster (you will get access accounts to the MSL cluster from MSL)
- Compile your C/C++ code from command line
- A useful Linux tutorial site: `https://ryanstutorials.net/linuxtutorial/`
- Some reference materials are listed in the course outline
- An up-to-date HPC related website for current top HPC systems and trends: `https://www.top500.org/`

WITS UNIVERSITY

Outline

WITS UNIVERSITY

What is HPC?

- What is high performance computing? High Performance Computing (HPC) most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get from a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.
- Key approach: ‘aggregating computing power’
- Computing power: Processors, memory, communication, I/O, . . .
- Modern processors are often multicore – parallel. How about memory?

Parallel computer and parallel computing

- What is a parallel computer? It is a collection of processing elements that cooperate to solve problems faster.
- What is parallel computing? To perform multiple computations simultaneously by decomposing a problem into smaller independent or interdependent tasks that can be executed in parallel across multiple processing units.

Outline

WITS UNIVERSITY

HPC problems

Table 1.1 Supercomputing Problem Representatives and How They Are Used in Academia, Industry, and Government

Supercomputing Problem Representatives	Academia	Industry	Government
Solution of partial-differential equations	Navier–Stokes equations, Einstein equations, Maxwell equations	Black–Scholes equation, Navier–Stokes equations for compressible flow, oil reservoir modeling	Weather prediction, hurricane modeling, storm-surge modeling, sea-ice modeling
Large systems with pair-wise force interactions	Cosmology, molecular dynamics simulations	Medicine development, biomolecular dynamics	Plasma modeling
Linear algebra	Supporting solution of partial-differential equations, fundamental benchmarks of HPL and high performance conjugate gradients	Search engine PageRank, finite-element simulations	HPC machine evaluation, climate modeling
Graph problems	Systems research, machine learning	Fraud detection	Security services, data analytics
Stochastic systems	Radiation transport, particle physics	Risk analysis in finance, nuclear reactor design, process control	Public health, modeling spread of disease

Figure: Common supercomputing problems and how they are used in academia, industry, and government. (Source: High performance computing modern systems and practice by Thomas Sterling et al.)

HPC problems cont.

Supercomputers drive all aspects of oil and gas workflows for exploration, production, and distribution.

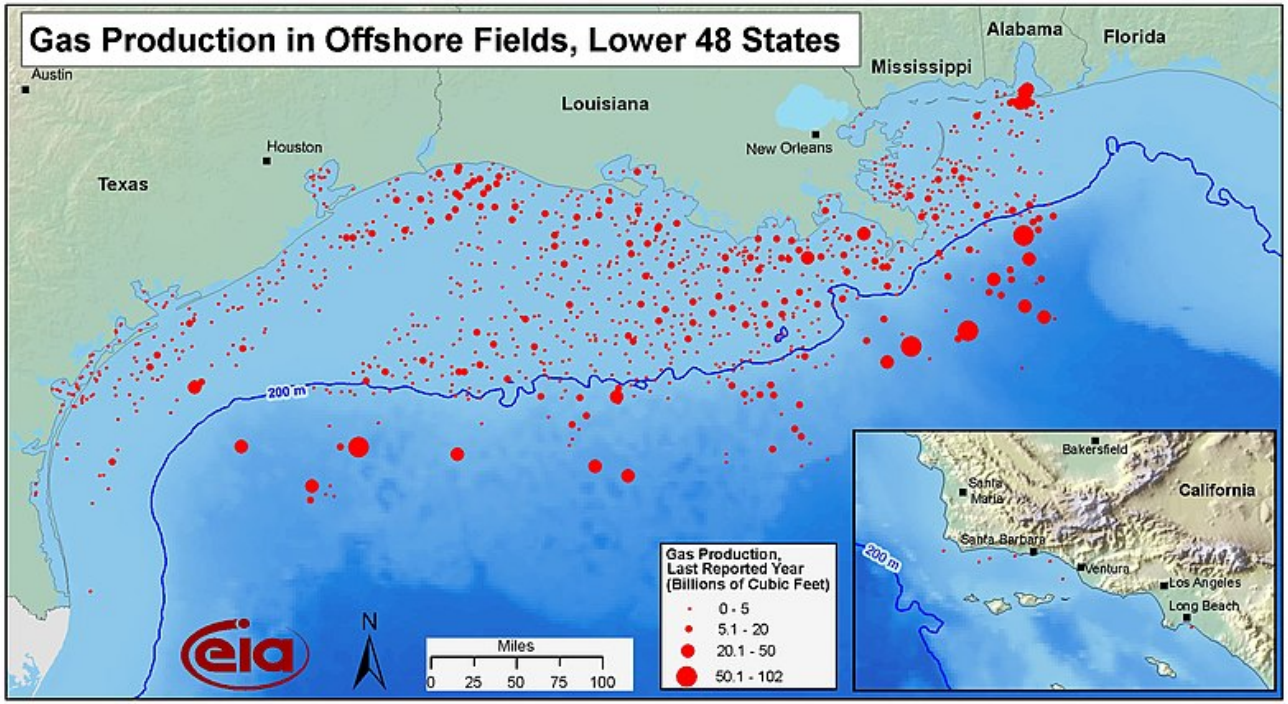

Figure: Researchers at BP use HPC to simulate subsurface geologies, using multidimensional analysis and characterization to identify oil reservoirs accurately. (Source: Image by US Energy Information Administration via Wikimedia Commons.)

WITS UNIVERSITY

HPC applications

Table 1.2 Broader Impacts of Supercomputing: How HPC Is Used by Different Application Domains to Accelerate Time to Innovation and Deliver Socioeconomic Impact

Vertical Segments	Common Workflows
Financial services	Fraud and anomaly detection, backtesting for algorithmic/proprietary trading, risk analytics
Oil and gas	Seismic processing, interpretation, reservoir modeling
Manufacturing	Materials simulation, structural simulations (noise/vibration/hardness and crash), aerodynamics simulations, design space exploration, thermal simulations and many more
Life sciences	Molecular dynamics, drug discovery, virtual modeling, genome sequencing and many more
Earth sciences	Atmospheric modeling, hydrodynamic modeling, ice modeling, coupled climate modeling

Figure: A subset of vertical segments that use HPC in their production environments to give faster time to innovation and deliver socioeconomic impact. (Source: High performance computing modern systems and practice by Thomas Sterling et al.)

A demo video of HPC simulations for Science and Industry:

`https://www.youtube.com/watch?v=iKR_L0xswdw`

WITS UNIVERSITY

Outline

WITS UNIVERSITY

Top performing supercomputers

No 1 supercomputer in the top500 list from Nov 2024 – El Capitan: `https://www.top500.org/system/180307/`;

Site	DOE/NNSA/LLNL
Manufacturer	HPE
Cores	11,039,616
Linpack Performance (Rmax)	1,742.00 PFlop/s
Theoretical Peak (Rpeak)	2,746.38 PFlop/s
Nmax	25,446,528
Power	29,580.98 kW
Processor	AMD 4th Gen EPYC 24C 1.8GHz
Interconnect	Slingshot-11
Operating System	TOSS
Compiler	g++ 12.2.1 and hipcc 6.2.0
Math library	AMD rocBLAS 6.0.2 and Intel MKL 2016
MPI	HPE Cray MPI

Top performing supercomputers cont.

YouTube Videos:

- El Capitan:

  `https://www.youtube.com/watch?v=cFHm_KiqQBs`

  `https://www.youtube.com/watch?v=MPRjIlgVwlg`
- Inside Elon Musk's Colossus Supercomputer:

  `https://www.youtube.com/watch?v=Tw696JVSxJQ`

Figure: El Capitan supercomputer

Outline

WITS UNIVERSITY

Why multicore processor?

- A primary reason is **the power wall** – refers to the electric energy consumption of a chip as a limiting factor for processor frequency increase.

Clock rate increase:
3400/12.5 =272
Power increase:
77/3.3 = 23.33

1.7 The Power Wall

Figure 1.16 shows the increase in clock rate and power of eight generations of Intel microprocessors over 30 years. Both clock rate and power increased rapidly for decades, and then flattened off recently. The reason they grew together is that they are correlated, and the reason for their recent slowing is that we have run into the practical power limit for cooling commodity microprocessors.

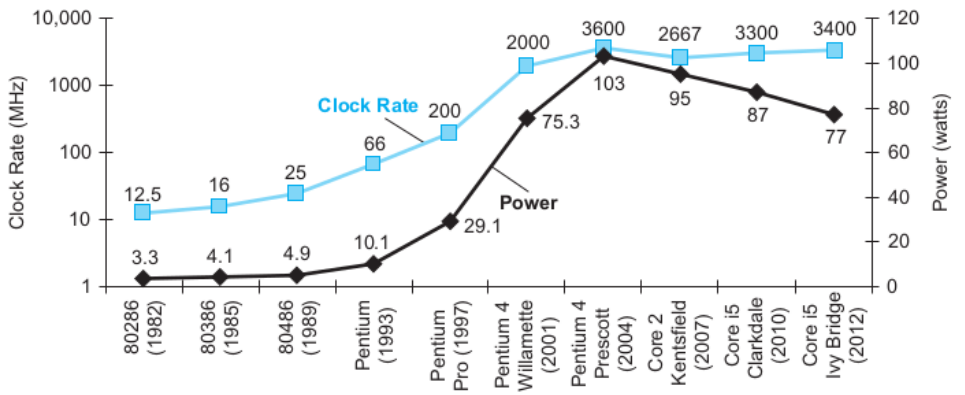

FIGURE 1.16 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. The Core i5 pipelines follow in its footsteps.

Figure: From book – Computer Organization and Design: The Hardware/Software Interface

WITS UNIVERSITY

Power wall explained

Although power provides a limit to what we can cool, in the PostPC Era the really critical resource is energy. Battery life can trump performance in the personal mobile device, and the architects of warehouse scale computers try to reduce the costs of powering and cooling 100,000 servers as the costs are high at this scale. Just as measuring time in seconds is a safer measure of program performance than a rate like MIPS (see Section 1.10), the energy metric joules is a better measure than a power rate like watts, which is just joules/second.

The dominant technology for integrated circuits is called CMOS (complementary metal oxide semiconductor). For CMOS, the primary source of energy consumption is so-called dynamic energy—that is, energy that is consumed when transistors switch states from 0 to 1 and vice versa. The dynamic energy depends on the capacitive loading of each transistor and the voltage applied:

$$Energy \propto Capacitive\ load \times Voltage^2$$

This equation is the energy of a pulse during the logic transition of $0 \rightarrow 1 \rightarrow 0$ or $1 \rightarrow 0 \rightarrow 1$. The energy of a single transition is then

$$Energy \propto 1/2 \times Capacitive\ load \times Voltage^2$$

The power required per transistor is just the product of energy of a transition and the frequency of transitions:

$$Power \propto 1/2 \times Capacitive\ load \times Voltage^2 \times Frequency\ switched$$

Frequency switched is a function of the clock rate. The capacitive load per transistor is a function of both the number of transistors connected to an output (called the *fanout*) and the technology, which determines the capacitance of both wires and transistors.

With regard to Figure 1.16, how could clock rates grow by a factor of 1000 while power grew by only a factor of 30? Energy and thus power can be reduced by lowering the voltage, which occurred with each new generation of technology, and power is a function of the voltage squared. Typically, the voltage was reduced about 15% per generation. In 20 years, voltages have gone from 5 V to 1 V, which is why the increase in power is only 30 times.

Figure: From book – Computer Organization and Design: The Hardware/Software Interface

WITS UNIVERSITY

Power wall explained cont.

The problem today is that further lowering of the voltage appears to make the transistors too leaky, like water faucets that cannot be completely shut off. Even today about 40% of the power consumption in server chips is due to leakage. If transistors started leaking more, the whole process could become unwieldy.

To try to address the power problem, designers have already attached large devices to increase cooling, and they turn off parts of the chip that are not used in a given clock cycle. Although there are many more expensive ways to cool chips and thereby raise their power to, say, 300 watts, these techniques are generally too expensive for personal computers and even servers, not to mention personal mobile devices.

Since computer designers slammed into a power wall, they needed a new way forward. They chose a different path from the way they designed microprocessors for their first 30 years.

Elaboration: Although dynamic energy is the primary source of energy consumption in CMOS, static energy consumption occurs because of leakage current that flows even when a transistor is off. In servers, leakage is typically responsible for 40% of the energy consumption. Thus, increasing the number of transistors increases power dissipation, even if the transistors are always off. A variety of design techniques and technology innovations are being deployed to control leakage, but it's hard to lower voltage further.

Elaboration: Power is a challenge for integrated circuits for two reasons. First, power must be brought in and distributed around the chip; modern microprocessors use hundreds of pins just for power and ground! Similarly, multiple levels of chip interconnect are used solely for power and ground distribution to portions of the chip. Second, power is dissipated as heat and must be removed. Server chips can burn more than 100 watts, and cooling the chip and the surrounding system is a major expense in Warehouse Scale Computers (see Chapter 6).

Figure: From book – Computer Organization and Design: The Hardware/Software Interface

Power wall explained cont.

- Power consumed by a transistor: dynamic and static power
- High power consumption leads to high heat (needs to be removed)
- Hence, power is a critical design constraint in modern processors
- For example,

Apple M1 laptop	13W
Intel Core i9 10900K (in desktop CPU)	95W
Nvidia RTX 4090 GPU	450W
Mobile phone processor	2W
World's fastest SCs	MegaW
Standard Micorwave oven	900W

Memory wall

- Memory wall is caused by the mismatch of the increases in CPU performance and memory speed.
- CPU performance has grown roughly according to the Moore's Law, however, the memory speed has grown at a much slower pace. (Please read an easy to follow introduction [1]of the problem.)

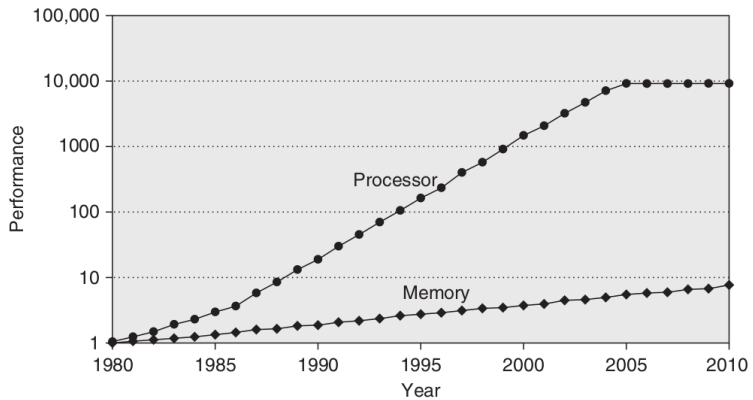

Figure: The gap between CPU performance and memory speed. Plots single processor performance projections against the historical performance improvement in time to access main memory. The processor line shows the increase in memory requests per second on average (i.e., the inverse of the latency between memory references), while the memory line shows the increase in DRAM accesses per second (i.e., the inverse of the DRAM access latency). The situation in a uniprocessor is actually somewhat worse, since the peak memory access rate is faster than the average rate, which is what is plotted.

---

[1]Wulf, W. A. and McKee, S. A. (1995). Hitting the memory wall: Implications of the obvious. *SIGARCH Comput. Archit. News*, 23(1):20–24

WITS UNIVERSITY

Outline

WITS UNIVERSITY

Summary

- Definitions of HPC, parallel computer, and parallel computing
- Understanding the scale of modern supercomputers and the applications of supercomputers
- Modern processors embrace multicore design path. This is because single-core or single thread of control performance is improving very slowly, to run programs significantly faster, programs must utilize multiple processing elements. Thus, we need to know how to write parallel and efficient code.
- Writing parallel programs can be challenging - requires problem partitioning, communication, and synchronization. To write an efficient parallel (even serial) code, you need to have knowledge of machine characteristics, and also very importantly, understand data movement.