NE 155, Class 2, S14

Rachel Slaybaugh

January 23, 2015

How do we measure utility?

IPS is often specified in the millions, giving MIPS. MIPS is used to measure the integer performance of a computer. Examples of integer operation include data movement (A to B) or value testing (If $A = B$, then C). MIPS as a performance benchmark is adequate for the computer when it is used in database query, word processing, spreadsheets, or to run multiple virtual operating systems

The **clock rate** of a CPU is normally determined by the frequency of an oscillator crystal. The clock rate determines how fast a processor can do work - it controls how quickly bits can be flipped (remember, what's really happening is switching between 1s and 0s).

FLOPS measures the computing ability of a computer and includes the concept of clock rate. As such, FLOPS is a useful measure of supercomputer performance. This is the term we will use most frequently in this class.

$$FLOPS = cores \times clock \times \frac{FLOPs}{cycle} \tag{1}$$

Computing Machines, Origins

Devices have been used to aid computation for thousands of years, mostly using one-to-one correspondence with fingers.

The abacus was a key early invention used for arithmetic tasks. What we now call the Roman abacus was used in Babylonia as early as *2400 BC*. This was the first known computer: the construction of **a tool that extended computation beyond one-to-one correspondence.**

Following the abacus there were a series of analog computers - machines designed to aid in the computation of specific tasks. These were found in ancient China, India, Greece, etc. None of the early computational devices were really computers in the modern sense, but they **took the concept of building a tool to do computation, and extended it to complex machinery.**

Scottish mathematician and physicist John Napier discovered that the multiplication and division of numbers could be performed by the addition and subtraction, respectively, of the logarithms of those numbers. The slide rule was thus invented in the 1620s to allow multiplication and division operations to be carried out significantly faster than was previously possible.

Now we're combining machines with improved ways to think about computation.

Early Development of Computing

Next, mechanical calculators, punched card data processing, and calculators using electric motors were developed.

Charles Babbage, an English mechanical engineer, originated the concept of a programmable computer. Considered the "father of the computer", he conceptualized and invented the first mechanical computer in the early 19th century. After working on his revolutionary difference engine, designed to aid in navigational calculations, he realized in 1833 that a much more general design, an Analytical Engine, was possible.

The input of programs and data was to be provided to the machine via punched cards, a method being used at the time to direct mechanical looms.

For output, the machine had a printer, a curve plotter, and a bell.

First Electromechanical Computer

The era of modern computing began with a flurry of development before and during World War II.

The Z3 was an electromechanical computer designed by Konrad Zuse and was completed in Berlin in 1941. It was the world's first working programmable, fully automatic digital computer. The Z3 was built with 2000 relays, implementing *22-bit words* that operated at a clock frequency of about *5–10 Hz*. Program code and data were stored on punched film.

The ENIAC (Electronic Numerical Integrator And Computer) was the first electronic programmable computer built in the U.S., announced to the public in 1946. It was digital, and capable of being reprogrammed (defined by the states of its patch cables and switches) to solve a full range of computing problems. The machine weighed 30 tons, used 200 kilowatts of electric power and contained over 18,000 vacuum tubes, 1,500 relays, and hundreds of thousands of resistors, capacitors, and inductors.

Stored Programs

Early computing machines had fixed programs. For example, a desk calculator is a fixed program computer. It can do basic mathematics, but it cannot be used as a word processor or a gaming console. By design, a stored-program computer includes an instruction set and can store in memory a set of instructions (a program) that details the computation.

The Manchester Small-Scale Experimental Machine was the world's first stored-program computer. It was built at the Victoria University of Manchester by Frederic C. Williams, Tom Kilburn, and Geoff Tootill, and ran its first program on 21 June 1948. It had a demonstrated speed of 1.1 kIPS.

It was the first working machine to contain all of the elements essential to a modern electronic computer. As soon as the SSEM had demonstrated the feasibility of its design, a project was initiated at the university to develop it into a more usable computer, the Manchester Mark 1. The Mark 1 in turn quickly became the prototype for the Ferranti Mark 1, the world's first commercially available, general-purpose computer.

The first application of a computer in an office environment was in 1951. By 1954 IBM introduced a "smaller", "more affordable" computer that proved very popular. The IBM 650 weighed over 900 kg, the attached power supply weighed around 1350 kg and both were held in separate cabinets of roughly 1.5 meters by 0.9 meters by 1.8 meters. It cost $500,000 ($4.35 million in 2014 $s) or could be leased for $3,500 a month ($30,000 in 2014 $s).

Microprogramming, Magnetic Storage, Transistors

Transistorized electronics improved not only the CPU, but also the peripheral devices. Removable / interchangeable disks could be made, which meant that much more data could be kept on hand.

Supercomputers

The early 1960s saw the advent of supercomputing. The Atlas Computer was a joint development between the University of Manchester, Ferranti, and Plessey, and was first installed at Manchester University and officially commissioned in 1962 as one of the world's first supercomputers.

In the U.S., a series of computers at Control Data Corporation (CDC) were designed by *Seymour Cray* to use innovative designs and parallelism to achieve superior computational peak performance.

The CDC 6600 was a mainframe computer delivered in 1965 to CERN, where it was used to analyse the 2-3 million photographs of bubble-chamber tracks that CERN experiments were producing every year. In 1966 another CDC 6600 was delivered to the Lawrence Radiation Laboratory, part of the University of California at Berkeley, where it was used for the analysis of nuclear events photographed inside the Alvarez bubble chamber. *The CDC*

6600 is generally considered to be the first successful supercomputer. the CDC 6600 was the world's fastest computer from 1964 to 1969.

Integrated Circuit

This new development heralded an explosion in the commercial and personal use of computers and led to the invention of the microprocessor. While the earliest microprocessor ICs literally contained only the processor, i.e. the CPU of a computer, their progressive development naturally led to chips containing most or all of the internal electronic parts of a computer.

This revolution led to the third generation of computers, and the developments since then have been more familiar.

Moore's Law (2)

This plot shows transistor counts (which is an analogue for integrated circuit complexity), clock speeds (how fast bits can be flipped), power consumption, and instruction-level parallelism (ILP). The doubling of transistor counts every $\approx$18 months is known as Moore's law, but over time, assumptions about performance and power consumption were also made and shown to advance along similar lines.

For decades, microprocessors followed what's known as Dennard scaling. Dennard predicted that oxide thickness, transistor length, and transistor width could all be scaled by a constant factor. Dennard scaling is what gave Moore's law its teeth.

CPU power consumption increased by 17.9x while CPU frequency improved by 125x. Moore's law held true into the mid-2000s, at which point the power consumption and clock speed improvements collapsed. The problem at 90nm was that transistor gates became too thin to prevent current from leaking out into the substrate. From 2007 to 2011, maximum CPU clock speed (with Turbo Mode enabled) rose from 2.93GHz to 3.9GHz, an increase of 33%. From 1994 to 1998, CPU clock speeds rose by 300%.

The death of conventional scaling has sparked a sharp increase in the number of companies researching various types of specialized CPU cores.

The fact that transistor density continues to scale while power consumption and clock speed do not has given rise to a new term: dark silicon. It refers to the percentage of silicon on a processor that cant be powered up simultaneously without breaching the chip's thermal design power (TDP).

Forms of Parallelism

Bit-level parallelism is a form of parallel computing based on increasing processor word size. Increasing the word size reduces the number of instructions the processor must execute in order to perform an operation on variables whose sizes are greater than the length of the word. (For example, consider a case where an 8-bit processor must add two 16-bit integers. The processor must first add the 8 lower-order bits from each integer, then add the 8 higher-order bits, requiring two instructions to complete a single operation. A 16-bit processor would be able to complete the operation with single instruction.)

ILP: A goal of compiler and processor designers is to identify and take advantage of as much ILP as possible. Ordinary programs are typically written under a sequential execution model where instructions execute one after the other and in the order specified by the programmer. ILP allows the compiler and the processor to overlap the execution of multiple instructions or even to change the order in which instructions are executed.

OS parallelism can be broken into

- **data**: distributing the data across different parallel computing nodes, each processor performs the same task on different pieces of distributed data

- **task**: distributing execution processes (threads) across different parallel computing nodes, each processor executes a different thread (or process) on the same or different data

Types of Memory

- **Shared Memory**: single memory space used by all processors; processors do not have to know where data resides
- **Distributed Memory**: each processor has its own private memory; tasks can only operate on local data; communicate with a remote processor to get remote data
- **Distributed Shared Memory**: each node of a cluster has access to a large shared memory in addition to each node's limited non-shared private memory.

In a distributed memory system there is typically a processor, a memory, and some form of interconnection that allows programs on each processor to interact with each other. The network topology is a key factor in determining how the multi-processor machine scales.

The key issue in programming distributed memory systems is how to distribute the data over the memories.

The advantage of shared memory is that it offers a unified address space in which all data can be found.

The advantage of distributed memory is that it excludes race conditions, and that it forces the programmer to think about data distribution.

GPUs

GPUs were originally developed for graphics processing (shocking) and are heavily used by the gaming industry. It turns out that the types of things needed in graphics processing are also needed by engineers and scientists (recent developments have branched into matrix and vector operations).

Scientists and engineers are starting to use GPUs for non-graphics computations, which is an interesting and challenging development.

GPUs can be

- dedicated: has RAM just for it
- integrated (IGP): use a portion of the computer's RAM rather than a dedicated amount (cheaper but less useful for gaming)
- hybrid: shared memory with a small amount of dedicated RAM

General Purpose GPU (GPGPU): using a GPU for computation applications usually handled by the CPU. Any GPU providing a functionally complete set of operations performed on arbitrary bits can compute any computable value. Additionally, the use of multiple graphics cards in one computer, or large numbers of graphics chips, further parallelizes the already parallel nature of graphics processing. OpenCL is the currently dominant open general-purpose GPU computing language. The dominant proprietary framework is Nvidia's CUDA.

MICs

MIC: Many Integrated Core architecture combines many CPU cores onto a single chip. Developers interested in programming these cores can use standard C, C++, and FORTRAN source code. The same program source code written for Intel MIC products can be compiled and run on a standard Intel Xeon processor. The MIC architecture uses a high degree of parallelism in smaller, lower-power performance Intel processor cores. The result is advanced performance on highly parallel applications.

Top 500 Computers

- Total combined performance of all 500 systems has grown to 309 Pflop/s, compared to 274 Pflop/s in June and 250 Pflop/s one year ago. T

- There are 50 systems with performance greater than 1 petaflop/s on the list, up from 37 six months ago.
- The No. 1 system, Tianhe-2, and the No. 7 system, Stampede, use Intel Xeon Phi processors to speed up their computational rate. The No. 2 system, Titan, and the No. 6 system, Piz Daint, use NVIDIA GPUs to accelerate computation.
- A total of 75 systems on the list are using accelerator/co-processor technology, up from 62 from November 2013.
- Ninety-six percent of the systems use processors with six or more cores and 85 percent use eight or more cores.