CSC2/452 Computer Organization
Systems of Tomorrow

Sreepathi Pai

URCS

Nov 21, 2022
Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow
Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow
Administrivia

- Assignment #4 is due tomorrow 1159PM
- Assignment #2 Redux will be out Nov 28, due Nov 30
- Assignment #5 (last!) will follow
Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow
Tomorrow

The future is already here – it’s just not very evenly distributed.

▶ William Gibson
1945: First Draft of the EDVAC

- Not very long ago
- Birth of the “CPU + RAM” design
- “Organization”
Moore Scaling: 1965

Moore’s Law – The number of transistors on integrated circuit chips (1971-2018)

Moore’s law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore’s law.

Observation that number of transistors in a given area was doubling every 2 years

This is exponential scaling

### TABLE I

**Scaling Results for Circuit Performance**

<table>
<thead>
<tr>
<th>Device or Circuit Parameter</th>
<th>Scaling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device dimension $t_{ox}$, $L$, $W$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Doping concentration $N_a$</td>
<td>$\kappa$</td>
</tr>
<tr>
<td>Voltage $V$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Current $I$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Capacitance $\epsilon A/t$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Delay time/circuit $VC/I$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Power dissipation/circuit $VI$</td>
<td>$1/\kappa^2$</td>
</tr>
<tr>
<td>Power density $VI/A$</td>
<td>1</td>
</tr>
</tbody>
</table>

The Bounty from Scaling

- More transistors, Same Area, Same Power
- Same number of transistors, Less Area, Less Power
  - Half the size, could run at twice the frequency
- Hence popular notion: speed of computers doubles every 18 months
- Enabled everything from Pixar to mobile phones in your pocket
When it all broken down: The Noughties

- Around 2004 to 2006, Dennard Scaling broke down
  - Transistors still growing smaller, but power usage not decreasing
  - More transistors, more power
  - More power in smaller area, heat dissipation problems
- Moore Scaling still applies
  - But pace has slowed down significantly
  - Lower limit is atomic size
- Modern chips have billions of transistors
  - NVIDIA’s Grace Hopper GPU has 80 billion transistors
  - Will we get to hundreds of billions?
Dark Silicon?

- Even if we can build chips with hundreds of billions of transistors, can we switch them all on?
- Dark Silicon is the idea that we will be unable to power all of the chip simultaneously
  - Some parts of the chip will be without power, i.e. “dark”
  - These parts will only be switched on demand
Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow
Multicore CPUs

- Performance cannot be improved by reducing $t$ (average time per unit of work)
- Increase $P$, parallelism!
- Thus, in 2004, Intel announces multicore CPUs
- But these are not transparent to programmers
  - You must rewrite software to take advantage of them!
If you have to rewrite software, are there alternatives to CPUs?

Lots of accelerators proposed around 2004–2006
- Cell BE (Sony PS3)
- Clearspeed
- Graphics Processing Units (GPUs)

An accelerator is a processor that can speed up some tasks
- CPU offloads these tasks to the accelerator
Graphics Processing Units: GPUs

- Originally designed to speed up graphics
  - Also games
- Could only do limited computation
- This changed around 2006
  - Could actually run C-like programs
GPUs enable HPC + Deep Learning + Bitcoin

- 2006 to 2011, GPUs enabled high performance computing
  - Mostly matrix multiply
  - Nearly all supercomputers had GPUs by then
- In 2011, GPUs were also found to be really fast at training “deep” neural networks
  - 100x than CPUs
  - (Re-)started the AI hype
- A few years later, GPUs also enabled Bitcoin/Ethereum, etc.
Domain-specific accelerators: When GPUs are too slow

- GPUs and CPUs are general-purpose
  - Can perform most computations
- Higher performance can be obtained by specializing
- Google's Tensor Processing Unit is an example
  - Specialized for tensor operations

By Zinskauf - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=77299254
Accelerators Galore

- Apple’s A8 Mobile System-on-Chip
- CPU, GPU, Caches and Memory Controllers highlighted.
- What about the rest of the chip?

Hill and Reddi, Accelerator-level Parallelism: Mobile SoCs as Harbinger of the Future, ISPASS 2019
Programmer’s View of this World

- Fixed-function accelerators
  - Custom interfaces to each
  - Use libraries
- General-purpose accelerators
  - New programming languages
  - GPUs: CUDA, OpenCL, ...
- Multicore CPUs
  - Parallel Programming Models (PThreads, OpenMP, etc.)
- Heterogeneity
  - Which pieces of your program should run where?
Current form of DRAM was invented in 1968

Modern computers use DDR SDRAM
  ▶ Introduced in 1992

Slow, much focus on improving memory bandwidth
  ▶ GPUs usually have around 300GByte/s
  ▶ Usually 256 bits/cycle

Figure by Glogger at English Wikipedia - Transferred from en.wikipedia to Commons., CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=5549293
Stacked DRAM: HBM and HMC

- Stack layers of DRAM in 3 dimensions
- High-bandwidth memory (HBM)
  - supports up to 4096 bits/cycle
- Hybrid Memory Cube
  - abandoned now
- Supported in some GPUs
  - upto 1TByte/s
  - A DVD is around 4–8GB

Courtesy AMD
embedded DRAM

- Move DRAM onto the main processor package
- Can use it for building caches instead of SRAM
  - Enables much larger caches
- Featured here: Intel Haswell GT3e

Arabinda Das, Intel’s Embedded DRAM: New Era of Cache Memory
Storage: Non-volatile Memory

- Rotating Media
  - Magnetic Hard Disks (Shingled – 10TB+)
  - Tape Systems (10TB+)
- Still very important, even when most data is stored remotely
  - Amazon Snowball Edge – 100TB (featured here)
  - Used to transfer data to Amazon’s servers

Source: https://aws.amazon.com/blogs/aws/aws-snowball-edge-more-storage-local-endpoints-lambda-functions/
Can transfer 100PB/trip

Who generates this much data?

- Satellite imagery is apparently around 10PB/year

Source: Move exabytes of data to cloud in weeks
Flash Memory

- Solid-state storage
  - No moving parts
  - Limited endurance (number of writes)
- Most common
  - USB Flash
  - SSD
3D XPoint

- Developed by Intel/Micron
  - Stores data using “bulk properties” of materials
- Two primary uses:
  - SSD (Flash memory replacement)
  - NVDIMM (DRAM replacement)
- Legend
  - Ochre: Selector
  - Green: Memory cell
  - Gray: Power

Picture by Trolomite - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45277623
Non-volatile Main Memory?

- Intel Optane DC
  - Claimed to be 8X denser than DRAM (6TB per system)
  - Others on their way
- Data remains in memory when you:
  - Switch off
  - Program ends
- How will writing programs change?
- Killed by Intel a few months ago

Programming NVM

- The Good:
  - No more translating between in-memory and on-disk representations
- The Bad:
  - What if a program crashes?
  - How to detect inconsistent state?
- The Strange:
  - Writes much slower than Reads
  - Both slower than DRAM
- Heterogenity in RAM
  - Which parts of data structures in DRAM? Which parts in NVRAM?
Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow
Mobile Computing

- Mobile phones obviously
- Virtual Reality
  - e.g. Head-mounted computers
- Augmented Reality
  - Glasses (mostly)
- Key challenges
  - Size, energy, power, performance

Hololens image by Ramadhanakbr - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=52546267
Data Centre Computing

- Consolidate compute, data storage in one central, physical location
- Access data and applications remotely
- Key challenges:
  - Size, energy, power, performance

By A5b - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=30380190
Cloud Computing

- On-demand access to computing resources
- Arguably, made famous by Amazon
  - Amazon AWS
- Amazon builds and runs data centres
  - You rent portions of it
  - Pay per time/data transferred, etc. (like a utility)
Intermittent Computing

- Featured here, a device that runs off the energy of radio waves
  - Think Wifi-powered devices
- Such energy-harvesting devices compute “intermittently”
  - Power could be lost any moment

Lucia et al., "Intermittent Computing: Challenges and Opportunities", SNAPL 2017
Where will computing systems appear next?
HINT: Remember the chip shortage?
Programming Tiny Computers in your Pocket and Computers in the Cloud

- Programming Languages and Tools?
- Program support?
  - Processes?
  - Virtual memory?
  - Parallelism?
  - Networking?
Virtualization

- Cloud computing requires a machine to run multiple operating systems
  - Virtualize entire machine
  - Can stop a virtual machine and move it to another machine in the cloud
- OS for the cloud
  - Virtual machines (e.g. Qemu, VirtualBox, VMware)
  - Hypervisors (e.g. Xen)
  - “Ultravisors”
Containers

- Same OS, different "applications"
- Isolating applications from each other
  - Different “tenants” in the cloud
  - Different installed apps
- As lightweight as possible
  - “Serverless” computing
- Inspired from the Plan9 OS
  - From Bell Labs
Challenges

- How to secure tenants?
  - Similar to protecting processes from each other
- How to account and bill for resources?
  - CPU time, bandwidth, storage, etc.?
Side-channel Attacks

- No direct channel to steal data
- But tenants are running on the same machine
- Can they use shared parts of hardware to leak information?
  - e.g., through cache hit/miss information?
- Such attacks are called side-channel attacks
  - e.g. Spectre
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID:
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

Courtesy:
https://www.cs.cmu.edu/~awb/linux.history.html
Certainly not "OS of tomorrow", even when it came out in 1990s

- 30-year old monolithic design
- built by a student like you, by reading 80386 manuals

But has enabled nearly every technology advance since then

- Supercomputers
- Mobile Phones
- Cloud
- 3D printers
- Airplane entertainment systems
- Networking devices
- Oscilloscopes

MS now ships Linux as part of Windows

- WSL2 1.0 was released last week
- For a 90s kid, this is the equivalent of hell freezing over
Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow
What about the rest of the system?

- Programming languages of tomorrow
  - Not C, possibly not Python, possibly not Java
  - Rust? Go?
  - WebAssembly? (not a direct programming language)
- Applications of tomorrow?
  - AI?
Reflections

- Machines will be increasingly heterogeneous
  - Both in processors and RAM
- Programming models will be very different
  - General-purpose
  - Domain-specific
  - Fixed-function
- Performance is now the responsibility of the programmer!
- Security is a major concern
  - Poor programming can hurt people