IO and Full System Performance
Today

• Quiz 7 recap
• IO
Key Points

- CPU interface and interaction with IO IO devices
- The basic structure of the IO system (north bridge, south bridge, etc.)
- The key advantages of high speed serial lines.
- The benefits of scalability and flexibility in IO interfaces
- Disks
  - Rotational delay vs seek delay
  - Disks are slow.
  - Techniques for making disks faster.
IO Devices
IO Devices

Large Hadron Collider
700MB/s
IO Devices

Large Hadron Collider
700MB/s

hard drive
50-120MB/s
IO Devices

- Large Hadron Collider: 700MB/s
- Hard drive: 50-120MB/s
- Keyboard: 10Byte/s
IO Devices

- Large Hadron Collider: 700MB/s
- Hard drive: 50-120MB/s
- Keyboard: 10Byte/s
- 30in display: 60Hz, 1GB/s
- Scanner: 10Byte/s

Image includes visual representations of various IO devices such as a monitor, hard drive, keyboard, scanner, and camera.
Hooking Things to Your (Parents’) Computer

• What do we want in an IO system?
What IO Should be

• Lots of devices
  • Keyboards -- slowest
  • Printers
  • Display
  • Disks
  • Network connection
  • Digital cameras
  • Scanners
  • Scientific equipment

• Easy to hook up
  • “Plug and play”
  • The fewer wires the better.

• Easy to make sw work
  • No drivers!
  • “just works”

• Performance
  • Fast!!!!
  • Low latency
  • High bandwidth
  • low power

• Cost
  • Cheap
  • Low hw and sw development costs
The CPUs World View

• The only IO that CPUs do is load and store
• “Programmed IO”
  • IO devices export “control registers” that drives map into the kernels address space
  • loads and stores to those addresses change the values in the control registers
  • Those address had better _______ and/or _______
  • Fine for small scale accesses

• Direct memory access
  • The CPU is slow for moving bytes around, and it’s busy too!
  • DMA allows devices directly read and write memory
  • Fill a buffer with some data, start the DMA (via PIO), go do other things.
The CPUs World View

• The only IO that CPUs do is load and store
• “Programmed IO”
  • IO devices export “control registers” that drives map into the kernels address space
  • loads and stores to those addresses change the values in the control registers
  • Those address had better Write through and/or _______
  • Fine for small scale accesses

• Direct memory access
  • The CPU is slow for moving bytes around, and it’s busy too!
  • DMA allows devices directly read and write memory
  • Fill a buffer with some data, start the DMA (via PIO), go do other things.
The CPUs World View

• The only IO that CPUs do is load and store
• “Programmed IO”
  • IO devices export “control registers” that drives map into the kernels address space
  • loads and stores to those addresses change the values in the control registers
  • Those address had better Write through and/or uncached
  • Fine for small scale accesses

• Direct memory access
  • The CPU is slow for moving bytes around, and it’s busy too!
  • DMA allows devices directly read and write memory
  • Fill a buffer with some data, start the DMA (via PIO), go do other things.
Interrupts

- IO devices need to get the CPUs attention
  - A DMA finishes
  - A packet arrives
  - A timer goes off
- (simplified) interrupt handling
  - CPU control transfers to the OS -- pipeline flush.
  - Like a context switch or a system call
  - Where control lands depends on the ‘interrupt vector’
  - The OS examines the system state to determine what the interrupt meant and processes it accordingly.
    - Copies data out of disk buffer or network buffer
    - Delivers signal to applications
    - etc.
Connecting Devices to Processors

- **On-chip**
  - Fastest possible connection.
  - Wide -- you can have lots of wires between devices
  - Fast -- data moves at core clock speeds
  - Cheap -- fewer chips means cheaper systems
  - Restricts flexibility -- Design is set at fab time
  - Current uses -- L2 caches, on-chip memory controller
  - Near term uses -- GPUs, network interfaces

AMD Phenom (aka barcelona)
The “Chip set”

• Off-chip is much slower.
  • Fewer wires, slower clocks (less bandwidth), and longer latency.

• North Bridge - The fast part
  • “Front side bus” in Intel-speak
  • Off-chip memory controller
  • PCI-express
  • Key system differentiator until recently.
    • Server chip sets vs desktop chip sets
  • Memory-like interface
  • Typically 64bits of data
  • Routes PIO requests to other devices
  • Lots of DMA
    • It’s sort of a data movement co-processor
  • >64GB/s of peak aggregate bandwidth
The “Chip set”

- The South bridge -- the slow part
  - Everything else...
  - USB
  - Disk IO
  - Power management
  - Real time clock
  - System status monitoring -- i2c bus
  - 100s of MB/s of bandwidth
Legacy Interfaces

- Serial lines -- RS 232
  - Dead simple and easy to use. Just four wires.
  - Point-to-point
  - Mice, terminals, modems, anything you can hack up.
  - Computers typically had 2
- Parallel ports
  - 8 bits wide
  - Printers, scanners, etc.
  - Computers typically had 1
- Various expansion card interfaces
  - ISA cards
  - Nu-BUS
Legacy Disk Interfaces

• **ATA - “AT Attachment”**
  • 16 bits of data in parallel
  • 40 or 80-conductor “Ribbon cables”
  • Peak of 133MB/s
  • Two drives per cable

• **SCSI -- Small Computer System Interface**
  • Synonymous with high-end IO
  • Fast bus speeds: up to 160Mhz QDR (four data transfers per clock)
  • Many variants up to SCSI Ultra-640: 640MB/s
  • Scalable: up to 16 devices per SCSI bus.
  • Expensive.
PCI/e

- “Peripheral Component Interconnect”
- The fastest general-purpose expansion option
  - Graphics cards
  - Network cards
  - High-performance disk controllers (RAID)
  - Slow stuff works fine too.
- Current generation in PCI Express (PCle)
The Serial Revolution

- Wider busses are an obvious way to increased bandwidth
  - But “jitter” and “clock skew” becomes a problem
  - If you have 32 lines in a bus, you need to wait for the slowest one.
  - All devices must use the same clock.
  - This limits bus speeds.
- Lately, high speed serial lines have been replacing wide buses.
High speed serial

- Two wires, but not power and ground
- “low voltage differential signaling”
  - If signal 1 is higher than signal 2, it’s a one
  - if signal 2 is higher, it’s a 0
  - Detecting the difference is possible at lower voltages, which further increases speed
- Max bandwidth per pair: currently 6Gb/s
- Cables are much cheaper and can be longer and cheaper -- External hard drives.
  - SCSI cables can cost $100s -- and they fail a lot.
Serial interfaces

- **USB** -- universal serial bus
  - Replaces Serial and parallel ports
  - Single differential pair. Up to 480Mb/s
    - Next gen USB will use 2 pairs for double the bandwidth
  - Scalable
    - A USB “bus” is a tree with the computer at the root, “hubs” as internal nodes and devices at the leafs.
    - Up to 255 devices per tree.
  - Complex -- high and slow speed modes, isochronous (predictable latency) operation of media

- **FireWire**
  - 1 differential pair, 400Mb/s
  - Scalable via “daisy chaining”
  - Better performance than USB because there’s less overhead.
Serial interfaces

• SATA -- Serial ATA
  • Replaces ATA
  • The logical protocol is the same, but the “transport layer” is serial instead of parallel.
  • Max performance: 300MB/s -- much less in practice.

• SAS -- Serial attached SCSI
  • Replace SCSI, Same logical protocol.

• PCIe
  • Replace PCI and PCIX
  • PCIe busses are actually point-to-point
  • Between 1 and 32 lanes, each of which is a differential pair.
  • 500MB/s per lane
  • Max of 16GB/s per card -- I don’t know of any 32 lane cards, but 16 is common.
Qualitative Improvements

- **Extensibility**
  - All current interconnect technologies are scalable
  - USB hubs
  - PCIe switches and hubs
  - etc.

- **Easy set up.**
  - No more setting jumpers
  - Auto-negotiation of PIO ranges etc.
  - Power is often included -- USB and firewire

- **Standards make developing new devices much easier**
  - serial-over USB
  - PCI over PCIe

- **Elegant design**
  - Express card (new laptop expansion slot) == PCIe 1x + USB
Qualitative Improvements

- Extensibility
  - All current interconnect technologies are scalable
  - USB hubs
  - PCIe switches and hubs
  - etc.

- Easy set up.
  - No more setting jumpers
  - Auto-negotiation of PIO ranges etc.
  - Power is often included -- USB and firewire

- Standards make developing new devices much easier
  - serial-over USB
  - PCI over PCIe

- Elegant design
  - Express card (new laptop expansion slot) == PCIe 1x + USB

This is Architecture: Building abstractions for dealing with the physical world.
## IO Interfaces

<table>
<thead>
<tr>
<th>Layer</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Layer</td>
<td>What commands are legal and when?</td>
</tr>
<tr>
<td></td>
<td>What do they mean?</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>How do you send a chunk of data?</td>
</tr>
<tr>
<td></td>
<td>Negotiating access?</td>
</tr>
<tr>
<td>Physical Layer</td>
<td>How do you send a bit?</td>
</tr>
<tr>
<td></td>
<td>What shape should connector be?</td>
</tr>
<tr>
<td></td>
<td>Voltage level?</td>
</tr>
</tbody>
</table>

- The protocol layer is largely independent of the lower layers
  - RS232 over USB
  - “IP over everything and everything over IP”
  - USB hard drives use the SCSI command set
Intel’s Latest: Tylersburg Chipset

North bridge

X58
IOH

South bridge

ICH10
ICH10R

- PCI Express 2.0
- Graphics Support for Multi-card configurations: 1x16, 2x16, 4x8 or other combination
- Up to 36 lanes
- 2 GB/s DMI
- 480 Mb/s each
- 500 MB/s each x1
- Intel Integrated 10/100/1000 MAC
- GLCI, LCI
- Intel Gigabit LAN Connect
- LPC or SPI
- BIOS Support
- Intel Extreme Tuning Support
- Intel High Definition Audio
- 6 Serial ATA Ports; eSATA; Port Disable
- Intel Matrix Storage Technology
- Intel Turbo Memory with User
- Optional
Hard Disks

- Hard disks are amazing pieces of engineering
  - Cheap
  - Reliable
  - Huge.
Disk Density

1 Tb/square inch
Hard drive Cost

- Yesterday at newegg.com: $0.008 GB ($0.000008/MB)
- Desktop, 1.5 TB
### The Problem With Disk: It’s Slooooonwww

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost</th>
<th>Access Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>on-chip cache</td>
<td>2.5 $/MB</td>
<td>&lt; 1ns</td>
</tr>
<tr>
<td>off-chip cache</td>
<td>0.07 $/MB</td>
<td>5ns</td>
</tr>
<tr>
<td>main memory</td>
<td>0.000008 $/MB</td>
<td>60ns</td>
</tr>
<tr>
<td>Disk</td>
<td>0.000008 $/MB</td>
<td>10,000,000ns</td>
</tr>
</tbody>
</table>
Why Are Disks Slow?

- They have moving parts :-(
  - The disk itself and the a head/arm
- The head can only read at one spot.
- High end disks spin at 15,000 RPM
  - Data is, on average, 1/2 an revolution away: 2ms
  - Power consumption limits spindle speed
  - Why not run it in a vacuum?
- The head has to position itself over the right “track”
  - Currently about 150,000 tracks per inch.
  - Positioning must be accurate with about 175nm
  - Takes 3-13ms
Making Disks Faster

- **Caching**
  - Everyone tries to cache disk accesses!
  - The OS
  - The disk controller
  - The disk itself.

- **Access scheduling**
  - Reordering accesses can reduce both rotational and seek latencies
RAID!

- Redundant Array of Independent (Inexpensive) Disks
- If one disk is not fast enough, use many
  - Multiplicative increase in bandwidth
  - Multiplicative increase in Ops/Sec
  - Not much help for latency.
- If one disk is not reliable enough, use many.
  - Replicate data across the disks
  - If one of the disks dies, use the replica data to continue running and re-populate a new drive.
- Historical foot note: RAID was invented by one of the text book authors (Patterson)
There are several ways of ganging together a bunch of disks to form a RAID array. They are called “levels”.

Regardless of the RAID level, the array appears to the system as a sequence of disk blocks.

The levels differ in how the logical blocks are arranged physically and how the replication occurs.
RAID 0

• Double the bandwidth.
• For an n-disk array, the n-th block lives on the n-th disk.
• Worse for reliability
  • If one of your drives dies, all your data is corrupt-- you have lost every nth block.
Real Disks

- Live Demo