I/O System

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Introduction (1)

- **I/O devices can be characterized by**
  - Behavior: input, output, storage
  - Partner: human or machine
  - Data rate: bytes/sec, transfers/sec
- **I/O bus connections**
## Introduction (2)

<table>
<thead>
<tr>
<th>Device</th>
<th>Behavior</th>
<th>Partner</th>
<th>Data rate (Mbit/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>Input</td>
<td>Human</td>
<td>0.0001</td>
</tr>
<tr>
<td>Mouse</td>
<td>Input</td>
<td>Human</td>
<td>0.0038</td>
</tr>
<tr>
<td>Voice input</td>
<td>Input</td>
<td>Human</td>
<td>0.2640</td>
</tr>
<tr>
<td>Sound input</td>
<td>Input</td>
<td>Machine</td>
<td>3.0000</td>
</tr>
<tr>
<td>Scanner</td>
<td>Input</td>
<td>Human</td>
<td>3.2000</td>
</tr>
<tr>
<td>Voice output</td>
<td>Output</td>
<td>Human</td>
<td>0.2640</td>
</tr>
<tr>
<td>Sound output</td>
<td>Output</td>
<td>Human</td>
<td>8.0000</td>
</tr>
<tr>
<td>Laser printer</td>
<td>Output</td>
<td>Human</td>
<td>3.2000</td>
</tr>
<tr>
<td>Graphics display</td>
<td>Output</td>
<td>Human</td>
<td>800.0000–8000.0000</td>
</tr>
<tr>
<td>Cable modem</td>
<td>Input or output</td>
<td>Machine</td>
<td>0.1280–6.0000</td>
</tr>
<tr>
<td>Network/LAN</td>
<td>Input or output</td>
<td>Machine</td>
<td>100.0000–10000.0000</td>
</tr>
<tr>
<td>Network/wireless LAN</td>
<td>Input or output</td>
<td>Machine</td>
<td>11.0000–54.0000</td>
</tr>
<tr>
<td>Optical disk</td>
<td>Storage</td>
<td>Machine</td>
<td>80.0000–220.0000</td>
</tr>
<tr>
<td>Magnetic tape</td>
<td>Storage</td>
<td>Machine</td>
<td>5.0000–120.0000</td>
</tr>
<tr>
<td>Flash memory</td>
<td>Storage</td>
<td>Machine</td>
<td>32.0000–200.0000</td>
</tr>
<tr>
<td>Magnetic disk</td>
<td>Storage</td>
<td>Machine</td>
<td>800.0000–3000.0000</td>
</tr>
</tbody>
</table>
I/O System Characteristics

- **Dependability is important**
  - Particularly for storage devices

- **Performance measures**
  - Latency (response time)
  - Throughput (bandwidth)
  - Desktops & embedded systems
    - Mainly interested in response time & diversity of devices
  - Servers
    - Mainly interested in throughput & expandability of devices
Interconnecting Components

- Need interconnections between
  - CPU, memory, I/O controllers

- Bus: shared communication channel
  - Parallel set of wires for data and synchronization of data transfer
  - Can become a bottleneck

- Performance limited by physical factors
  - Wire length, number of connections

- More recent alternative: high-speed serial connections with switches
  - Like networks
Bus Types

- **Processor-Memory buses**
  - Short, high speed
  - Design is matched to memory organization

- **I/O buses**
  - Longer, allowing multiple connections
  - Specified by standards for interoperability
  - Connect to processor-memory bus through a bridge
Bus Signals

- **Data lines**
  - Carry address and data
  - Multiplexed or separate

- **Control lines**
  - Indicate data type, synchronize transactions

- **Synchronous**
  - Uses a bus clock

- **Asynchronous**
  - Uses request/acknowledge control lines for handshaking
## I/O Bus Examples

<table>
<thead>
<tr>
<th></th>
<th>Firewire</th>
<th>USB 2.0</th>
<th>PCI Express</th>
<th>Serial ATA</th>
<th>Serial Attached SCSI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intended use</strong></td>
<td>External</td>
<td>External</td>
<td>Internal</td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td><strong>Devices per channel</strong></td>
<td>63</td>
<td>127</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td><strong>Data width</strong></td>
<td>4</td>
<td>2</td>
<td>2/lane</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td><strong>Peak bandwidth</strong></td>
<td>50MB/s or 100MB/s</td>
<td>0.2MB/s, 1.5MB/s, or 60MB/s</td>
<td>250MB/s/lane</td>
<td>300MB/s</td>
<td>300MB/s</td>
</tr>
<tr>
<td><strong>Hot pluggable</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Depends</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Max length</strong></td>
<td>4.5m</td>
<td>5m</td>
<td>0.5m</td>
<td>1m</td>
<td>8m</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>IEEE 1394</td>
<td>USB Implementers Forum</td>
<td>PCI-SIG</td>
<td>SATA-IO</td>
<td>INCITS TC T10</td>
</tr>
</tbody>
</table>
Typical x86 PC I/O System

- **Intel Xeon 5300 processor**
- **Front Side Bus (1333 MHz, 10.5 GB/sec)**
- **FB DDR2 667 (5.3 GB/sec)**
- **PCIe x16 (or 2 PCIe x8) (4 GB/sec)**
- **Main memory DIMMs**
- **Serial ATA (300 MB/sec)**
- **ESI (2 GB/sec)**
- **PCIe x8 (2 GB/sec)**
- **Disk**
- **LPC (1 MB/sec)**
- **I/O controller hub (south bridge) Enterprise South Bridge 2**
- **USB 2.0 (60 MB/sec)**
- **Parallel ATA (100 MB/sec)**
- **CD/DVD**

**Notes:**

- PCI-X bus (1 GB/sec)
- PCI-X bus (1 GB/sec)
- PCIe x4 (1 GB/sec)
- PCIe x4 (1 GB/sec)
I/O Management

- I/O is mediated by the OS
  - Multiple programs share I/O resources
    - Need protection and scheduling
  - I/O causes asynchronous interrupts
    - Same mechanism as exceptions
  - I/O programming is fiddly
    - OS provides abstractions to programs
I/O Commands

- **I/O devices are managed by I/O controller h/w**
  - Transfers data to/from device
  - Synchronizes operations with software

- **Command registers**
  - Cause device to do something

- **Status registers**
  - Indicate what the device is doing and occurrence of errors

- **Data registers**
  - Write: transfer data to a device
  - Read: transfer data from a device
I/O Register Mapping

- Memory mapped I/O
  - Registers are addressed in the same space as memory
  - Address decoder distinguishes between them
  - OS uses address translation mechanism to make them only accessible to kernel

- I/O instructions
  - Separate instructions to access I/O registers
  - Can only be executed in kernel mode
  - Example: x86
Polling

- **Periodically check I/O status register**
  - If device ready, do operation
  - If error, take action

- **Common in small or low-performance real-time embedded systems**
  - Predictable timing
  - Low hardware cost

- **In other systems, wastes CPU time**
Interrupts

- **When a device is ready or error occurs**
  - Controller interrupts CPU

- **Interrupt is like an exception**
  - But not synchronized to instruction execution
  - Can invoke handler between instructions
  - Cause information often identifies the interrupting device

- **Priority interrupts**
  - Devices needing more urgent attention get higher priority
  - Can interrupt handler for a lower priority interrupt
I/O Data Transfer

- **Polling and interrupt-driven I/O**
  - CPU transfers data between memory and I/O data registers
  - Time consuming for high-speed devices

- **Direct memory access (DMA)**
  - OS provides starting address in memory
  - I/O controller transfers to/from memory autonomously
  - Controller interrupts on completion or error
DMA/Cache Interaction

- If DMA writes to a memory block that is cached
  - Cached copy becomes stale

- If write-back cache has dirty block, and DMA reads memory block
  - Reads stale data

- Need to ensure cache coherence
  - Flush blocks from cache if they will be used for DMA
  - Or use non-cacheable memory locations for I/O
DMA/VM Interaction

- **OS uses virtual addresses for memory**
  - DMA blocks may not be contiguous in physical memory

- **Should DMA use virtual addresses?**
  - Would require controller to do translation

- **If DMA uses physical addresses**
  - May need to break transfers into page-sized chunks
  - Or chain multiple transfers
  - Or allocate contiguous physical pages for DMA
Measuring I/O Performance

- I/O performance depends on
  - Hardware: CPU, memory, controllers, buses
  - Software: operating system, database management system, application
  - Workload: request rates and patterns

- I/O system design can trade-off between response time and throughput
  - Measurements of throughput often done with constrained response-time
TPC Benchmarks

- **Transactions**
  - Small data accesses to a DBMS
  - Interested in I/O rate, not data rate
  - Measure throughput
    - Subject to response time limits and failure handling
    - ACID (Atomicity, Consistency, Isolation, Durability)
    - Overall cost per transaction

- **Transaction Processing Performance Council**
  - TPC-APP: B2B application server and web services
  - TPC-C: on-line order entry environment
  - TPC-E: OLTP for brokerage firm
  - TPC-H: decision support system
File System & Web Benchmarks

- SPEC System File System (SPEC SFS)
  - Synthetic workload for NFS server, based on monitoring real systems
  - Results
    - Throughput (operations/sec)
    - Response time (average ms/operation)

- SPEC Web Server benchmark (SPECweb)
  - Measures simultaneous user sessions, subject to required throughput/session
  - Three workloads: Banking, Ecommerce, and Support
I/O vs. CPU Performance

- **Amdahl’s Law**
  - Don’t neglect I/O performance as parallelism increases compute performance

- **Example**
  - Benchmark takes 90s CPU time, 10s I/O time
  - Double the number of CPUs/2 years
    - I/O unchanged

<table>
<thead>
<tr>
<th>Year</th>
<th>CPU time</th>
<th>I/O time</th>
<th>Elapsed time</th>
<th>% I/O time</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
<td>90s</td>
<td>10s</td>
<td>100s</td>
<td>10%</td>
</tr>
<tr>
<td>+2</td>
<td>45s</td>
<td>10s</td>
<td>55s</td>
<td>18%</td>
</tr>
<tr>
<td>+4</td>
<td>23s</td>
<td>10s</td>
<td>33s</td>
<td>31%</td>
</tr>
<tr>
<td>+6</td>
<td>11s</td>
<td>10s</td>
<td>21s</td>
<td>47%</td>
</tr>
</tbody>
</table>
I/O System Design

- **Satisfying latency requirements**
  - For time-critical operations
  - If system is unloaded
    - Add up latency of components

- **Maximizing throughput**
  - Find “weakest link” (lowest-bandwidth component)
  - Configure to operate at its maximum bandwidth
  - Balance remaining components in the system

- **If system is loaded, simple analysis is insufficient**
  - Need to use queuing models or simulation
Applications are increasingly run on servers
  • Web search, office apps, virtual worlds, ...

Requires large data center servers
  • Multiple processors, networks connections, massive storage, ...
  • Space and power constraints

Server equipment built for 19” racks
  • Multiples of 1.75” (1U) high
Rack-Mounted Servers

Sun Fire x4150 1U server

- 2 Redundant power Supplies
- 3 PCI Express Slots
- System Status LEDs
- Management NIC
- 2 USB Ports
- Management Serial
- 4 Gigabit NICs
- Video
Sun Fire x4150 1U Server

- 4 cores each
- 16 x 4GB = 64GB DRAM
Given a Sun Fire x4150 system with

- Workload: 64KB disk reads
  - Each I/O op requires 200,000 user-code instructions and 100,000 OS instructions
- Each CPU: \(10^9\) instructions/sec
- FSB: 10.6 GB/sec peak
- DRAM DDR2 667MHz: 5.336 GB/sec
- PCI-E 8x bus: 8 x 250MB/sec = 2GB/sec
- Disks: 15,000rpm, 2.9ms avg. seek time, 112MB/sec transfer rate

What I/O rate can be sustained?

- For random reads, and for sequential reads
I/O System Design Example (2)

- **I/O rate for CPUs**
  - Per core: $10^9 / (100,000 + 200,000) = 3,333$
  - 8 cores: 26,667 ops/sec

- **Random reads, I/O rate for disks**
  - Assume actual seek time is average/4
  - Time/op = seek + latency + transfer
    - $= 2.9\text{ms}/4 + 4\text{ms}/2 + 64\text{KB}/(112\text{MB/s}) = 3.3\text{ms}$
  - 303ops/sec per disk, 2424 ops/sec for 8 disks

- **Sequential reads**
  - $112\text{MB/s} / 64\text{KB} = 1750$ ops/sec per disk
  - 14,000 ops/sec for 8 disks
I/O System Design Example (3)

- **PCI-E I/O rate**
  - 2GB/sec / 64KB = 31,250 ops/sec

- **DRAM I/O rate**
  - 5.336 GB/sec / 64KB = 83,375 ops/sec

- **FSB I/O rate**
  - Assume we can sustain half the peak rate
  - 5.3GB/sec / 64KB = 81.540 ops/sec per FSB
  - 163,080 ops/sec for 2 FSBs

- **Weakest link: disks**
  - 2424 ops/sec random, 14,000 ops/sec sequential
Pitfalls (1)

- Offloading to I/O processors
  - Overhead of managing I/O processor request may dominate
    - Quicker to do small operation on the CPU
    - But I/O architecture may prevent that
  - I/O processor may be slower
    - Since it’s supposed to be simpler
  - Making it faster makes it into a major system component
    - Might need its own coprocessors!
Pitfalls (2)

- Backing up to tape
  - Magnetic tape used to have advantages
    - Removable, high capacity
  - Advantages eroded by disk technology developments
  - Makes better sense to replicate date
    - E.g., RAID, remote mirroring
Pitfalls (3)

- **Peak performance**
  - Peak I/O rates are nearly impossible to achieve
  - Usually some other system components limits performance
  - E.g., transfers to memory over a bus
    - Collision with DRAM refresh
    - Arbitration contention with other bus masters
  - E.g., PCI bus: peak bandwidth $\sim 133$MB/sec
    - In practice, max 80MB/sec sustainable
Concluding Remarks

- **I/O performance measures**
  - Throughput, response time
  - Dependability and cost also important

- **Buses used to connect CPU, memory, I/O controllers**
  - Polling, interrupts, DMA

- **I/O benchmarks**
  - TPC, SPEC SFS, SPECweb