Input/Output

Forecast

- Motivation
- Disks
- Networks
- Buses
- Interfaces
- Examples

Motivation

I/O needed

- To/from users (e.g., display)
- To/from non-volatile media (disk)
- To/from other computers (networks)

Key questions

- How fast?
- Getting faster?

Examples

<table>
<thead>
<tr>
<th>Device</th>
<th>I or O?</th>
<th>Partner</th>
<th>Data Rate KB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>mouse</td>
<td>I</td>
<td>human</td>
<td>0.01</td>
</tr>
<tr>
<td>graphics display</td>
<td>O</td>
<td>human</td>
<td>60,000</td>
</tr>
<tr>
<td>modem</td>
<td>I/O</td>
<td>machine</td>
<td>2-8</td>
</tr>
<tr>
<td>LAN</td>
<td>I/O</td>
<td>machine</td>
<td>500-6000</td>
</tr>
<tr>
<td>tape</td>
<td>storage</td>
<td>machine</td>
<td>2000</td>
</tr>
<tr>
<td>disk</td>
<td>storage</td>
<td>machine</td>
<td>2000-10,000</td>
</tr>
</tbody>
</table>

I/O Performance

What is performance

Supercomputers write and read 1G of data
- want high bandwidth to vast data (bytes/sec)

Transaction processing does many independent small I/Os
- want high I/O rates (I/Os /sec)
- sometimes fast response times

File systems
- want fast response time first
- lots of locality
Magnetic Disks

Stack of platters
two surfaces per platter
tracks
heads move together
sectors

Disk access:
- queuing + seek
- + rotation + transfer

seek = 10-20 ms but smaller with locality
rotation = 1/2 rotation/3600 rpm = 8.3 ms (5400 rpm - 5.6 ms)
transfer = x/2-4MB/s (4kB/4MB/s = 1 ms)

(Remember: mechanical == ms)

Disk Trends

- $/MB down ($100/GB = 10 cents/MB)
- disk diameter 14" --> 1.8" --> 1"
- seek time down
- rotation unchanged
- transfer rates up

optical
- CD ROMS good for read only
- Write once read-write less good!

RAID

What if we want to store data on 100 disks
MTTF = 5 years/100 = 18 days!
RAID 1 = mirror= stored twice = 100% overhead
RAID 3 = bit-wise parity = small overhead
RAID 5 = block-wise parity = small overhead and small writes
Local Area Network (LAN) = Ethernet

Original Ethernet
- one-write bus with collisions and exponential backoff
- within building
- 10Mb/s (~= 1MB/s)

Now Ethernet is
- point to point to clients (switched network)
- with hubs
- client s/w unchanged
- 100Mb/s --> 1Gb/s

LAN

Ethernet is no longer technically optimal

Nevertheless, many standards have failed to displace it (e.g., token rings, ATM)

Emerging Approach: System Area Network (SAN)
- Reduce SW stack (TCP/IP)
- Reduce HW stack (e.g., interface on memory bus)
- New Standard: Infiniband (http://www.infinibandta.org)

WAN

E.g., ARPANET, Internet
arranged as a DAG
backbones now 1Gb/s; 100Gb/s in the future
TCP/IP - protocol stack
- Transmission control protocol, Internet protocol

Key issues:
- Top-to-bottom systems issues
- getting net into homes
- digital subscriber loop (DSL), cable modem, ??

Buses in a Computer System
Buses

A bunch of wires
- arbitration
- control
- data (address)
+ flexible, low cost
  - bandwidth bottleneck

Types
- processor-memory
  - short, fast, custom
- I/O
  - long, slow, standard
- Backplane
  - medium, medium, standard

Synchronous - has clock
- everyone watches clock and latches at appropriate phase
- transactions take fixed or variable number of clocks
- faster but clock limits length
  - Processor-memory

Asynchronous - requires handshakes
- more flexible
- I/O

Asynchronous Handshake

(1) Request made & (2) request seen
(3) Request deasserted & (4) ack deasserted
(5) Data sent & (6) Data received & (7) ack deasserted
Asynchronous Control

Buses

Synchronous w.r.t. asynchronous
- must distribute clock and deal with skew
  + simple handshake
- hard to backward compatible with slow devices
  + no metastability problems

For memory buses
- pipelined in-order responses
- out-of-order responses

Buses

Improving bandwidth
- wider bus
- separate/multiplexed address/data lines
- block transfer

Bus Arbitration

one or more potential bus masters; others slaves
- bus request
- bus grant
- priority
- fairness

Implementations
- Centralized (also logical central but distributed FSMs)
- Distributed (e.g., original Ethernet)
Buses

Bus Standards
Also PCI
  • 32-64 bit data
  • synchronous 33 MHz clock
  • multiple masters
  • 111 MB/s peak bandwidth

Tape

Revolution caused by helical scan tapes
  • 8mm video tape + more ECC (as did CD-ROM)
  • 2 GB/tape at $10/tape = 0.5cent/MB in 1993 -cheaper now!
  • tape robots that hold many tapes per reader
  • library of congress without pictures is 10TB
  • 5000 tapes
  • $ 50,000
  • not that simple!

Frame Buffer

e.g., 1560 x 1290 pixels x 24 bits/color pixel - 5.7 MB
refresh whole screen 30 times/sec = 170MB/s > PCI!
on memory bus
use 24 video DRAMs (dual ported)
  • refresh display and allow image change by CPU
  • DRAM port
  • serial port to video
See AGP (Accelerated Graphics Port)

Interfacing

Three characteristics:
  • multiple users share I/O resources
  • I/O often use interrupts to communicate to CPU
  • low-level details of I/O devices complex

Three functions:
  • virtualize resources - protection, scheduling, etc
  • interrupts similar to exceptions
  • device drivers
Interfacing

How do you give I/O device a command?
- Memory-mapped
  - special addresses not for memory
  - send commands as data
- I/O commands
  - special opcodes
  - send over I/O bus

Interfacing

How do I/O devices communicate with CPU
- poll on devices
  - waste CPU cycles
  - poll only when device active
- interrupts
  - different from exceptions although similar!
  - info in cause register
  - vectored interrupts

Interfacing

Transfer data
- polling and interrupts - by CPU
- OS transfers data

Too many interrupts?
- use DMA so interrupt only when done
- use I/O channel - extra smart DMA

Interfacing

DMA
- CPU sets up
  - device id, operation, memory address, #bytes
- DMA
  - performs actual transfer (arbitrates, buffers, etc)
  - interrupt CPU

Typically I/O bus with devices (e.g., hard drive) uses DMA
Interfacing

DMA virtual or physical addresses?

Cross page boundaries in DMA?
  • virtual
    • translation map entries
    • translations provided by OS
  • physical
    • one page per transfer
    • OS chains the physical addresses

no page faults in between - nail down pages

Interfacing

Multiprogramming
  • I/O through OS
  • syscall interface between program and OS
  • OS checks protection, etc
  • OS runs device drivers
  • suspends current process and switches process
  • I/O interrupt fielded by OS
  • OS completes I/O and makes process runnable
  • after interrupt, run next ready process

Interfacing

Caches and I/O
  • I/O in front of cache - slows CPU
  • I/O behind cache - cache coherence?
  • OS invalidate/flush cache first before I/O

Apple Mac 7200 (Fig. 8.16)
'99 Compaq SP700

PC I/O [Section 7.2, Hill et al., 2000]

PCI -- Peripheral Component Interface -- ‘93 133MB/s ‘93
AGP -- specialized PCI to graphics accelerators
PCMIA -- 2-4 high credit-card size slot
USB -- 12Mb/s for many low-performance devices
FireWire/IEEE 1394 “plug & play” at 800Mb/s
IDE -- ‘83 8MB/s for disk in PC enclosure
SCSI -- ‘86 for higher performance disks at 1.5x cost premium