Lecture #29

storage systems
1. Disks – bw, capacity improving more quickly than seek
2. disk performance models
3. RAID – reliability, performance
4. tertiary – much more capacity, much worse seek

Outline - 1 min

Networks:
   background, motivation
   performance metrics
   ♦ bandwidth, latency, overhead
   hardware issues
   ♦ media
   ♦ network interface
   ♦ switches
   ♦ network topology
   connections v. connectionless
   ♦ congestion control

Preview - 1 min

<see above>

Lecture - 20 min

Networks: Background, Motivation

------------------------------------------
Impact of networks on architecture is manifest

80’s LANs → workstations, PCs; move away from mainframes
90’s WWW → ?computing → communication?;
  ?more mainframes (huge data/information centers)
  (unexpected, at least to me, but internet means that
  millions of people can access a popular server,
  so distribution has actually increased the need for
  big servers
90’s “killer network” – cheap switches → interconnect hundreds of
PCs in a building

- Goal: Communication between computers
- Eventual goal: treat collection of computers as if one big computer;
distributed resource sharing
- Theme: different computers must agree on many things
  - overriding importance of standards and protocols

Many networks: Ethernet, modem, wireless, ATM, FDDI, X.25, T1, T3, …
All basically the same:
Facets of networks people talk about a lot

- Direct (point-to-point) v. indirect (multi-hop)
- topology (bus, ring, hypercube)
- routing algorithms
- switching
- wiring (copper, coax, fiber)

What really matters

- latency
- bandwidth
- cost
- reliability

3 communities

- MPP – performance, latency, bandwidth
- LAN – workstations, cost
- WAN – telecommunications, reliability, phone call revenue

→ we’ll try to pull together into single terminology

ABC’s of Networks

(Motivate complexity incrementally)
Starting point – send bits between 2 computers
  • queue (FIFO) on each end
  • can send both ways “Full duplex”
  • Rules for communication: protocol
    • simple protocol – once computer can read data from the other
    • Request(address); Response(data)
    • \[\rightarrow\] need request, response signaling

Messages: headers, trailers

<table>
<thead>
<tr>
<th>Request/Response</th>
<th>Address/data</th>
<th>&lt;none&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Header”</td>
<td>“Body”</td>
<td>trailer</td>
</tr>
</tbody>
</table>

header/trailer - information to deliver a message
header:
  0: please send data from address
  1: packet contains data corresponding to request
body/payload: data in message

Physical Reality: Packets
Abstraction: messages
limited size arbitrary size
unordered ordered
unreliable reliable
machine-to-machine process-to-process
only on LAN routed anywhere
asynchronous synchronous
not secure secure

QUESTION: how do you build these abstractions?

What if more than 2 computers want to communicate?
    → address field
What if packet garbled?
    → add error detection (e.g. CRC)
What if packet lost?
    → more elaborate protocols to detect loss
e.g. NAK, timeouts
What if multiple processes/machine
    per-process queue

These questions lead to more complex protocols

Manage complexity via layering
e.g. TCP/ip
    IP: get data from one machine to another
        (addressing, routing)
    TCP: unlimited size, multiple processes, in-order, reliable, congestion control

******************************************************************************

Admin - 3 min
******************************************************************************
Sermon: simplicity
project checkpoints today

**************************
Lecture - 24 min
**************************

Latency
Overhead
Bandwidth

Overhead v. Latency
overhead – time used to insert a message (CPU busy)
latency – time spent waiting for a message

difference is analogous to pipelining in CPU
  overhead ~ pipeline stage
  latency ~ pipeline depth
(a little different because BANDWIDTH also limits rate of inserting
new packets into pipeline; pipelining analogy holds for small, fixed-size packets)

\[ \rightarrow \text{you can hide latency, but you always pay for overhead} \]

Link bandwidth v. bisection bandwidth

link bandwidth – how fast can one machine insert bits onto the wire

bisection bandwidth – accounts for interference among different streams of communication

bisection bandwidth definition – min cut of network

---

Performance Metrics: another view

Example performance measures

----------------------------------------

MPP (CM-5)  LAN (Ethernet)
<table>
<thead>
<tr>
<th></th>
<th>Bisection BW</th>
<th>Link BW</th>
<th>Latency</th>
<th>HW Overhead send/recv</th>
<th>SW overhead send/recv</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N * 5 MB/s</td>
<td>20 MB/s</td>
<td>5 us</td>
<td>.5us/.5us</td>
<td>1.6us/12.4us</td>
</tr>
<tr>
<td></td>
<td>1.125 MB/s</td>
<td>1.125 MB/s</td>
<td>15 us</td>
<td>6us/6us</td>
<td>200us/241us</td>
</tr>
</tbody>
</table>

Example of importance of overhead

---

100 MB/s network with 100us overhead v. 10MB/s network with 10us overhead:

![Latency v. Bandwidth](attachment:image.png)

Example – intel paragon had a high bandwidth network (40 MB/s?), but software overhead was > 100us → needed to send huge amounts of data to get within 10% of peak overhead
**Importance of Latency and Overhead**

Study: NSF trace over 1 week: 95 msgs < 200 bytes
Ethernet: 9Mbit/s BW; 456us overhead
ATM Synoptics: 78Mbit/s BW; 626us overhead
→ ATM’s 8x better bandwidth → 20% better performance
(latency predicts performance better than BW!)

Moral: bandwidth is not correct measure of network performance (like MIPS)

*****************************************************************************

Summary - 1 min
*****************************************************************************

Networks: huge impact on architecture
   ♦ standards, protocol -driven
   ♦ performance – not just bandwidth!!