CSC 546: Client/Server Fundamentals

Fall 2000

Management aspects

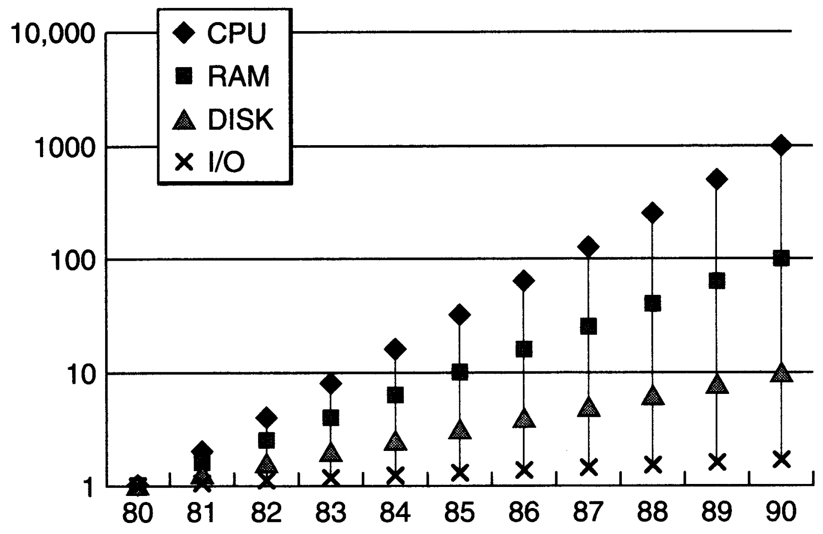
- economics
  - specialized architectures
  - relative cost, marginal cost
  - cost factors
- priorities for implementing client/server
- transitioning an enterprise into client/server
- operational challenges

Economics of client/server

from previous discussions:

- desktop machines + network architectures made client/server feasible
- faster/smaller components made client/server cost effective
- process reengineering, info-centric economy made client/server desirable

technical forces are converging on client/server technology



relative growth rates in the 1980's

- CPU power doubled every year
- RAM density doubled every 1.5 years
- DISK density doubled every 3 years
- I/O access speed doubled every 10 years

Note: logarithmic scale

➔ performance gap is significant, growing

Specialized architectures

the technology performance gap has led to specialized architectures

- not cost-effective to optimize both processing and storage

- instead, optimize one architecture for CPU and memory (client)
- optimize separate architecture for disk storage and I/O (server)

➔ client/server model is natural result of technology performance gap

high cost of I/O relative to CPU also leads to client/server

- optimize a few servers for disk storage and I/O
- amortize the high cost over many lower-cost clients

Economy of scale

few large, specialized servers + many clients ➔ cost-effective

- doubling server performance more than doubles system performance

```
responseTime = 1/(serviceRate – arrivalRate)
```

Suppose 1 server, can process 30 I/O operations per sec
            20 clients, each can generate 1 I/O request per sec

```
responseTime = 1/(30 – 20) = 1/10 = 100 msec
```

If double server performance:

```
responseTime = 1/(60 – 20) = 1/40 = 25 msec
```

If double again:

```
responseTime = 1/(120 – 20) = 1/100 = 10 msec
```

Economy of scale (cont.)

What happens if you double server performance & double # of clients?

response time cut in half!

```
responseTime = 1/(30 – 20) = 1/10 = 100 msec

responseTime = 1/(60 – 40) = 1/20 = 50 msec

responseTime = 1/(120 – 80) = 1/40 = 25 msec

responseTime = 1/(240 – 160) = 1/80 = 12.5 msec
```

in general: if increase server performance & client #'s by a factor of N,
then decrease system response time by factor of N as well!

```
initResponse = 1/(initService – initArrival)

doubResponse = 1/(N*initService – N*initArrival)
             = 1/(N*(initService – initArrival))
             = 1/N * (1/(initService – initArrival))
             = 1/N * initResponse
```

Degrees of specialization

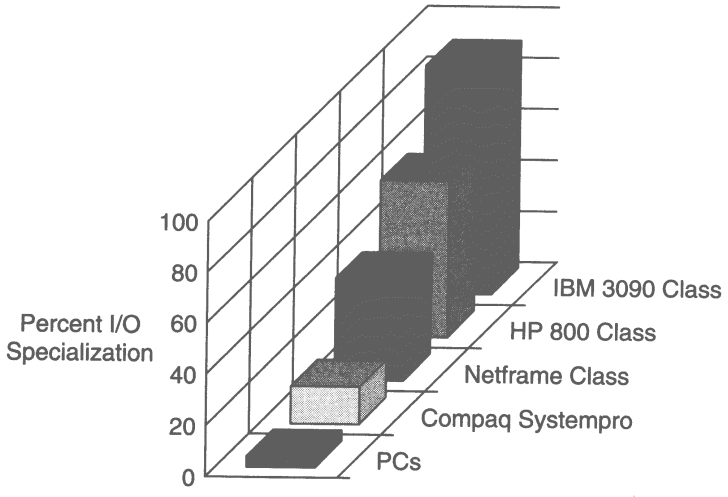
specialized server architectures are becoming commonplace

e.g., Sun SparcCenter, HP 9000/800, Pyramid, Sequent

can determine degree of specialization by comparing MIPS

Suppose 1 65-MIP CPU
2 16-MIP disk I/O controllers
1 4-MIP network I/O controller

% CPU specialization = (CPU MIPS/Total MIPS) = 65/101 = 64%
% I/O specialization = (I/O MIPS/Total MIPS) = 36/101 = 36%



continuum of
I/O specialization

Optimized client/server configurations

most client systems are optimized for

- CPU
- RAM
- display components

low price more important than reliability – failure only affects 1 person

most server systems are optimized for

- I/O throughput
- DISK storage

reliability essential – failure affects many people (in mission-critical ways)

e.g., Redundant Array of Independent Disks (RAID)

Economic factors

relative computational costs

- it is less expensive to do computation on a workstation than on a mainframe

typical (1996) numbers:

mainframe = $30M / 500 MIPS = $60K per MIP

workstation = $3K / 65 MIPS = $46 per MIP

marginal CPU costs

- *marginal utility* is the increase in total benefit resulting from the new process

for mainframe, 90% of processing power may be dedicated to I/O

$60K per MIP / 0.1 = $600K per delivered CPU MIP

if 200 users, (500 MIPS * 0.1)/200 = 0.25 CPU MIPS per user

→ $600K / 0.25 = $2.4M for delivered CPU MIP

for workstation, 90% of processing power may be dedicated to CPU

$46 per MIP / 0.9 = $51 per delivered CPU MIP

if 200 users, each client delivers dedicated MIPS to user

→ $51 * 200 machines = $10.2K for delivered CPU MIP

Economic factors (cont.)

marginal I/O costs

for mainframe, 90% of processing power may be dedicated to I/O

$60K per MIP / 0.9 = $67K per delivered I/O MIP

if 200 users, (500 MIPS * 0.9)/200 = 2.25 I/O MIPS per user

→ $67K / 2.25 = $30K for delivered CPU MIP

for workstation, 90% of processing power may be dedicated to CPU

$46 per MIP / 0.1 = $460 per delivered I/O MIP

if 200 users, each client delivers dedicated MIPS to user

→ $460 * 200 machines = $92K for delivered I/O MIP

it makes economic sense to downsize mainframes to specialized servers

- consider 250 MIP server, 75% dedicated to I/O, cost of $50K

$50K / 250 MIPS = $200 per MIP

$200 per MIP / 0.75 = $150 per delivered I/O MIP

if 200 users, (250 MIPS * 0.75)/200 = 0.9 MIPS per user

→ $150 / 1.1 = $165 for delivered CPU MIP

Client/server cost factors

absolute computational costs

- by moving computation to clients, can downsize to cheap, specialized servers

commodity cost advantages

- by moving to client/server using standard components, can save

depreciation

- mainframes may have ~5 yr lifetime , but not useful for other tasks
- workstations have shorter lifetime (~3 yrs), but can be trickled down

reduced development costs

- developing software on mainframes is costly since CPU time is expensive
- can develop code on cheaper workstations using standard development tools

reduced opportunity costs

- slow development yields backlogged project, postpones benefits

improved user efficiency

- application response time is improved since interface is local, server optimized for I/O
- hopefully, translates into improved productivity
- danger: must provide desktop support – don’t expect workers to install/maintain

Additional costs

operational costs

- not all costs are reduced by client/server (administration 10-30% higher) due to management of desktop machines and attendant logistics
- relative operations cost is reduced since client/server delivers significantly more processing power to users

```
relativeOpCost = totalOpCost / totalTansactionsDelivered
```

- desktop machines require as much support as other computers

  in 1994, estimated that 73% of client/server operating cost was personnel

  3-year desktop lifecycle: 30% acquisition, 5% maintenance, 65% support

maintenance costs

- standard equipment and software is easier, cheaper to maintain

facility costs

- mainframes require large, special purpose facilities
- can be power intensive

  e.g., in one case, 200 servers used same power as one mainframe disk drive

Strategic priorities for implementing client/server

refocus IT

- introducing new ways of doing business can meet cultural resistance
- must redirect significant IT support to desktop systems
- may have to rethink billing (# desktops? # network drops? # transactions?)

use technology for competitive advantage

- focus on productivity (more, better, faster, less workers, …)
- data integration (i.e., shared, central database) can be a significant factor

enterprise-wide architecture

- enterprise computing requires a well-planned, integrated architecture
- application frameworks can provide a structure for integrating applications

open systems

- need standards so that components fit together and interact effectively

closed: proprietary, tied to a single vendor (e.g., Mac OS)
semi-open: vendor provides API or licensed source (e.g., Windows)
open: industry concensus (e.g., Motif), formal standards (e.g., POSIX),
public domain (e.g., TCP/IP, HTML)

Transitioning into client/server

success factors

- manage the transition, develop a roadmap to the end system
- move forward in phases, demonstrate paybacks at each step

risk areas

- technical risk: inadequate capacity planning, poor integration, scalability
  must fully understand technology, develop prototypes, plan big
- management risk: cultural resistance, dual support during transition
  prepare a detailed deployment plan, communicate, balance old & new systems
- operations risk: expertise with desktops, new OS's may be scarce
  establish standards and conventions, retraining

transition strategies

- Green Field: start replacing old systems, all new systems should be client/server
  fastest transition, minimizes dual support; costly, requires rapid shift
- Incremental: build client/server prototypes of old systems, gradually flesh out
  lowest design risk, cost spread out over time; slower, throwaway development
- Evolutionary: migrate task to client (terminal, interface, validation, processing, …)
  moderate cost & design risk; very slow, lots of throwaway, can lose functionality

Operational strategies

1. Use service-oriented metrics that are user-centric, people-cost sensitive, and cycle-time oriented.
2. Set configuration standards for hardware platforms, peripherals, OS, network software, …
3. Set up every client system identically, allowing variations only for role of desktop (admin, marketing, engineering), type (mobile, home, office), and OS (Windows, Mac, UNIX).
4. Keep your servers and communications equipment safe (both physically and environmentally) and clearly identified.
5. Use common directory services to reduce application administration, and improve security and data consistency.
6. Proactively administer all of your systems (client, server, & network) to prevent outages.
7. Avoid the need to backup desktop machines keeping data on fileservers & performing automated backups on those servers.
8. Organize your network printing environment by monitoring print queues, establishing consistent naming conventions, and planning for sufficient supplies
9. Reduce the hidden cost of user support by providing help desks, onsite help and training.
10. Organize your software into user-oriented bundles and plan for periodic updates.
11. Plan your hardware maintenance around standard configurations by paying for higher reliability and swapping out entire client systems to get users up and running faster.
12. Get control over your desktop assets by taking an inventory of existing machines, and centralizing hardware and software purchasing.

Next week…

Communicating objects

- remote method invocation
- Open Distributed Processing (ODP) object model
- CORBA vs. COM/OLE

Read Chapter 14

As always, be prepared for a short quiz