CSC 546: Client/Server Fundamentals

Fall 2000

See online syllabus at:

www.creighton.edu/~davereed/csc546

Course goals:

- introduce client/server concepts
- basic understanding of issues involved in design, implementation, and evaluation of client/server systems
- working knowledge of existing applications and implementation techniques

What is client/server?

client/server describes a *logical* relationship between *processes*

- an entity (client) requests a service
- another entity (server) provides that service

client: professor with bad spelling skills
server: student with dictionary

classic example: relational database management system

- server maintains and searches database (Oracle, Ingres, Informix, Sybase)
- database front-ends receive and process queries from users,
  request data from server, format and display reports

Pervasive examples

print server

- server maintains print queue, file processing
- client applications request a print job, transmit file

World Wide Web

- server maintains Web pages, links to other servers
- browsers present links to users, request pages from server, display pages

e-mail

- server stores messages and routing/timing information
- apps allows users to compose message, requests transmission from server
  apps requests access to incoming messages, displays to user

Client/server involves processes, not machines

- the same machine can run multiple processes, both server & client

  client: student asks for knowledge, guidance
  server: professor lectures, answers questions

  same machine might run Web server, mail server, printer server, …

- server/client relationship can be internal to a machine

  client/server: professor looks up his own spellings

  can run browser, access mail, print from server machine

Distributed processing

client/server is an example of a distributed processing architecture

Basic terminology:

- *multiprogramming*: multiple programs/processes in memory
  but only 1 executes at a time (e.g., DOS)
- *multitasking*: can execute multiple processes pseudo-concurrently
  preemptive: OS can interrupt process (e.g., OS/2, UNIX)
  cooperative: process can only be interrupted if designed (Mac, Windows)
- *multiprocessing*: multiple CPU's execute processes concurrently
  tightly coupled: single OS controls CPU sharing (Compaq Systempro)
  loosely coupled: multiple OS's, must coordinate execution (LAN/WAN)
- *Interprocess Communication (IPC):* message passing facilities
  provided by OS to allow concurrent processes to communicate/coordinate

Distributed architectures

among networks, the computation/logical load can vary

mainframe architecture:

- all intelligence resides within central host computer
- can be accessed via dumb terminals (transmit & display keystrokes)

file-sharing architecture (LAN):

- central host computer contains files & applications
- files/applications are downloaded over network, executed locally

client/server architecture:

- more even distribution of work, less data transmission
- can introduce multiple tiers of servers, hierarchical network

Client/server attributes

client:

- proactive
- intermittent
- maintains and processes the user dialog

  e.g., screen handling, menu/command interpretation, data validation, help processing, error recovery, GUI controls
- fat vs. thin

server:

- reactive
- continuous
- provides functional service
- processes multiple requests concurrently

e.g., shared resource servers: backup server, print server, mail server
display server: X windows
access server: firewall

client/server communications are: *transactional, cooperative, many-to-few*

Advantages of client/server

flexibility

- mainframes provide only one interface (usually textual)

- with client/server, can create different interfaces for different modes
  e.g., marketing interface, inventory interface, …
- each client can select/customize interface
  e.g., PC GUI already familiar/intuitive to many users

reduced network traffic

- file-sharing systems require transmitting entire files/databases

- with client/server, files/databases reside on server
  only response to client's request needs to be transmitted
- interface details, query verification, response formatting handled by client

Advantages (from a business perspective)

reduced system development and maintenance time

- mainframes are expensive, software is nonstandard
- packaged client/server applications are usually cheaper and simpler
- centralized updates are easier and less disruptive

Motorola

old system: maintained in-house database system on mainframes

$30K per month on coding & software updates

new system: UNIX clients running X windows, off-the-shelf applications

computing costs reduced from 3.2% of annual revenue to < 1% ($150M over 2 yrs)

Panalpina World Transport (U.K.)

old system: 200 offices with local PC software, upload data to mainframe

IT support had to travel to remote sites for upgrades & repairs

new system: single server, clients in each office

no data duplication, reduced support staff (450 → centralized staff of 3)

Hyatt Corporation

client/server reservation system: update times reduced from 6 months to 2-3 weeks

operating costs reduced by 30%, staff reduced from 23 to 6

More advantages (from a business perspective)

data is readily accessible to decision makers

- information is centralized, always up-to-date
- fast, accurate information can increase organizational responsiveness

Lutheran Hospitals (Chicago)

patient data is stored in a database on a central server
can be updated by doctor/nurse/administrator, immediately accessible

Lexmark International (KY)

client/server system links production, marketing, finance, and HR
manufacturing cycle reduced from 4 hours to 24 minutes
inventory turns over 59% faster, volume per shift increased 33%

National Gallery of Art (D.C.)

client/server system maintains > 600 pieces of info on each object in collection
each department has access to relevant information about objects
movement subsystem helps plan, monitor & coordinate shipments
acquisitions subsystem helps integrate new objects, plan & arrange shows

Disadvantages of client/server

resistance to change

- new technology requires new expertise, change in operations

high training costs

- since work is distributed, more workers need expertise

extensive planning required

- distributed computation can lead to chaos

inadequate standards

- system management, maintenance, and security not standardized

susceptibility to security violations

- access to centralized data must be controlled

Next week…

Technology & business perspectives

- business pressures leading to client/server
- technology backdrop
- enterprise computing

Read Chapters 1, 2 and 3

Be prepared for quiz on

- today's lecture (moderately thorough)
- the reading (superficial)