G53DBC

Distributed Data Processing

Introduction to Distributed Data Processing (DDP)

- Movement and structure of data around organisations
- Range of data processing approaches
  - Classified according to degree of 'Centralisation'
  - Currently, most businesses employ hybrid approaches
  - Client/Server architectures

Centralized Data Processing (CDP)

- Centralised computers, processing, data, control, support
- What are the advantages?
  - Economy for equipment and personnel
  - Lack of duplication
  - Ease in enforcing standards, security

Distributed Data Processing (DDP)

- Computers are dispersed throughout organisation
- Allows greater flexibility in structure
- More redundancy
- More autonomy

Why is DDP Increasing?

- Dramatically reduced hardware costs
- Increased desktop power
- Improved user interfaces (!)
- Ability to share data across multiple servers

DDP Pros & Cons

- There are no complete solutions
- Key issues
  - How does it affect end-users?
  - How does it affect management?
  - How does it affect productivity?

Benefits of DDP (1)

- Responsiveness
- Availability
- Organisational Patterns
- Resource Sharing
- Incremental Growth
- Increased User Involvement & Control

Benefits of DDP (2)

- End-user Productivity
- Distance & location independence
- Privacy and security
- Vendor independence

Drawbacks of DDP

- Difficulties in failure diagnosis
- More components and dependence on communication means more points of failure
- Incompatibility of components
- Incompatibility of data
- More complex management & control
- Difficulty controlling information resources
- Suboptimal procurement
- Duplication of effort

Reasons for DDP

- Need for new applications
  - On large centralised systems, development can take years
  - On small distributed systems, development can be component-based and very fast
- Need for short response time
  - Centralised systems result in contention among users and processes
  - Distributed systems provide dedicated resources

Hybrid processing approaches

- Centralized systems (mainframes, etc)
- Distributed systems (PCs)
- Networked systems
- Client-Server computing

Client/Server Architecture

- Combines advantages of distributed and centralized computing
- Cost-effective, achieves economies of scale
- Flexible, scalable approach

Intranets

- Often use Internet standards
- Content is accessible only to internal users
- A specialized form of client/server architecture

Internet

- Provides access to outside users
  - Customers
  - Vendors / suppliers
- 'Extranets'
  - extensive, controlled, secure access
  - usually mapped to a particular WAN

Distributed applications

- Horizontal partitioning
  - Different applications on different systems
  - One application replicated on systems
  - Example: Office automation
- Vertical partitioning
  - One application dispersed among systems
  - Example: Retail chain

Keeping data

- Tailored to size of business
  - Small businesses can often rely on a collection of files (e.g. text and numerical data)
  - Large businesses will often rely on one or more databases
  - Distributed organisations will often need to distribute databases
    - I/O
    - Store

Distributed databases (1)

- Centralised databases
  - No duplication of data
  - Good for security and consistency
  - Contention for access
  - Response times poor
  - Breakdowns propagate to local sites

Distributed databases (2)

- Replicated databases
  - Less data contention
  - High/replicated storage & data update costs
  - Three variants of replication
    - Real-time (two-phase commit, on entry)
    - Near Real-time (batch backups, e.g. every 30 mins.)
    - Deferred (bulk transfer, once or twice per day)

Distributed databases (3)

- Partitioned databases
  - No duplication, limited contention
  - Disperses load
  - Graceful degradation?
  - Ad hoc reports more difficult to assemble
  - Maintaining separation as data evolves difficult

Networking Requirements of DDP

- Connectivity requirements
  - What links between components are necessary?
- Availability requirements
  - Percentage of time application or data is available to users
- Performance requirements
  - Response time requirements

Summary

- CDP and DDP
- Benefits and pitfalls of DDP
- Architectures for hybrid approaches to DP
  - Client/Server architecture
- Distributed applications
- Distributed data
- Networking requirements
- Next time: Intranets and Internet