IMS1907 Database Systems

Week 5
Database Systems Architecture

Systems Architecture
- The 'blueprints' of the system
- Focus on the individual 'building blocks' of the system and how they are put together
  - hardware, software, network, database
  - encourages independence between components
- Often considers logical and physical views of the system and its components
- Match 'user needs' to architecture
- 'Art' vs 'Engineering'?

Database Systems Architecture
- You have considered several different 'views' of databases
  - enterprise view, ER models, tables, datasheets, forms, reports, queries
  - different aspects of the logical and physical views
- A schema
  - a representation, model or specification of a view of a database
- Database systems are based on the ANSI/SPARC standard three-schema architecture

Three-schema Architecture
- The ANSI/SPARC standard for describing the structure of data (1978) consists of three schema
  - external schema
    - user views
    - conceptual schema
      - single, coherent definition of enterprise data
  - internal schema
    - physical storage structures

External Schema
- Combination of the enterprise data model (top-down) and a collection of detailed (bottom-up) user views
- User view is a logical description of some portion of the database required to perform a task
  - guided by user requirements
- Represent data access and authorisation at the individual users' level
- Conceptually a relation, but not actually stored in DBMS
  - records in a view are computed as needed
Conceptual Schema

- Detailed specification of the overall structure of organisational data
- Complete logical view
  - independent of any DBMS technology
- Usually depicted graphically - ER, OO modelling
- Schema specifications stored as metadata
- Scope is entire organisation or major business area

Conceptual Schema

- Includes all entity types and subtypes
- All relationships are documented
- All attributes are documented - keys specified
- Data types, formats, domains, and business rules are specified and stored in repository
- Ideally data model is fully normalised
  - acceptable and common to normalise in the logical schema

Internal Schema

- Physical storage structure details
- Representation of the conceptual schema as it is physically stored on particular DBMS technologies
- A conceptual schema can have many internal schemas
- Consists of a logical and physical schema
- Good design relies on understanding of how data is accessed and used

Internal Schema

- Logical schema
  - representation of data for particular DBMS
    - relational, OO, dimensional
  - tables, data types, formats, keys, …
  - derived by transforming elements of conceptual schema to DBMS structures

Internal Schema

- Physical schema
  - set of specifications describing how data from a logical schema are stored in a computer’s secondary memory for a specific DBMS
  - ideally one physical schema for each logical schema
  - describes organisation of physical records, file organisations, access paths to data, usage of indexes, clusters, …

Database System Development

- The conceptual schema and external schema are typically developed iteratively until both are fully defined
- Logical schema is developed by transforming conceptual schema (or parts of it) to implementation model constructs
- Associated physical schema is specified taking into account the software, hardware and network characteristics along with users’ performance expectations
- Inconsistencies discovered in physical schema may require iteration back to design of conceptual schema
Data Independence

- Ability to change the schema at one level of a DB without having to change the schema at the next higher level
- Logical data independence
  - the capacity to change the conceptual schema without having to change external schema or application
- Physical data independence
  - the capacity to change the internal schema without having to change conceptual or external schema
- Only mappings between levels should change

Logical data independence
- changes in the way data is defined should not affect what the user sees
- only metadata should change

Physical data independence
- changes in the way data is stored should not affect what the user sees, nor the conceptual view of the data

This means that application programs are insulated from:
- changes in the way the data is structured - logical
  - changes in the way data is defined should not affect what the user sees
- changes in the way the data is stored - physical
  - changes in the data storage method should not affect what the user sees, nor the conceptual view of the data

A major decision in database systems design relates to where the data is physically stored and processed.

Many different types of database systems in use in enterprises:
- need to balance organisational, technical and usage issues
- data for a given IS may reside in multiple locations on many machines
- different types of processing occur at different locations

Significant factors in the growth of client/server architecture:
- increasing systems complexity
- the proliferation of web-enabled systems

Need for an application level of ‘servers’ that handle transactions from ‘client’ machines against some back-end database.

Data can exist on many different types of server:
- database server, application server, client server, web server

We commonly consider the following three architectural layers or tiers:
- Client tier
- Application or Web server tier
- Enterprise or Data Services server tier

Sometimes this view is limited to the client tier and a general server tier.
Client/server Architecture

- Application or Web server tier
  - sometimes called process services tier
  - houses applications
    - A/P, A/R, Orders, Sales, Inventory, ...
  - houses web services
    - processes HTTP protocols, scripting tasks, dynamic web pages, session management, calculations
  - provides data access
    - access and connectivity to DBMS

Enterprise server tier

- sometimes called the data services or database tier
- sometimes stored on a mainframe or minicomputer
- transaction databases containing all organisational data, summarised data on departmental databases
- performs sophisticated calculations
- manages merging of data from multiple sources
- manages multiple requests for data from multiple sources

Database Systems Architecture

- in a client/server architecture
  - DBMS software on a server (database server or database engine) performs database commands sent to it directly from client workstations or via application servers
  - client concentrates mainly on user interface functions
  - application servers concentrate on application-related processing functions
  - allows distribution of database across all types of user groups and one central server, as a single distributed database or as a set of physically related databases

Implications for database development

- ease of separation of development of database and modules that maintain it, from the IS applications that access and present database contents to users
- many programming languages provide easy-to-use GUIs
  - Powerbuilder, Java, VB.NET, ...
- middleware greatly facilitates application access to data across large, complex systems
- opportunities for reuse of software components

References