Database Systems: Achievements and Opportunities

Avi Silberschatz
Michael Stonebraker
Jeff Ullman

History and Vocabulary

- **Relational Databases hugely successful**
- Represent all data by simple tabular data structures (relations)
- Access data thru high-level query language
  - Specify a predicate
  - Query optimizer translates predicate into algorithm to perform DB access
- **Transaction: sequence of operations that appear "atomic" when executed**
  - Either all or none of the actions in the sequence happen
  - Transform database between consistent states

- **Concurrency control: each transaction appears to execute in isolation**
  - Synchronize and serialize transactions
- Lock data items to prohibit conflicting accesses
- Two-phase locking: requires transactions to obtain all locks it needs before releasing any locks

- **Recovery: assures that hardware or software failures do not result in inconsistent state**
- Write-ahead logging: summary of effects of transaction stored to log (sequential file)
- **Commit a transaction: actually change database**
- Don't commit until record written to log
- If transaction doesn't complete, log restores state

Challenges for Modern Databases

- Application examples:

1. NASA data of satellite images
2. Computer-aided design applications
   - maintain multiple versions of data
3. Human genome project
   - gene sequence is several billion elements long
4. Data mining
   - e.g., stores have huge databases on customers
   - "mine" this data for hidden patterns
5. Multimedia databases for insurance companies

New Kinds of Data

- Databases optimized for delivering small records
- Paradigm breaks down for large objects

- **Type management**
- Traditional type-checking is dynamic
- Errors show up at run-time
- Need to combine
  - static type discipline
  - persistent data
  - database structure that changes over time

- **.....Move toward object-oriented databases**

More new challenges

- **Rule processing**
  - Declarative (if A is true, then B is true)
  - Imperative (if A is true, then do C)
- Elaborate constraints that must be enforced
- Triggered actions that require processing in response to specific events
- Provide logical declaration of what the user wants, not detailed specification of actions
- Tools for validating and debugging a large collection of rules

Challenges

- **New Data Models**
- Spatial data: multidimentsional access
  - 2-dimensional points, lines, polygons
  - Find 10 closest neighbors to some data element
- Time: retrieve/explore past state
- Uncertainty: attach likelihood that data represents a certain phenomenon

- **Scaling**
- Larger than a terabyte will be common
- Building traditional indexes hard
- Backup and recovery hard

- **Parallelism**

- **Tertiary Storage**
- Placement of data to minimize retrieval times

- **Maintaining multiple versions**
- Example: design databases

- **Heterogeneous, distributed databases**
- Want them to inter-operate
- Individual databases under local control
- Connected by networks
- Semantically inconsistent (e.g., PhD database)

- **Need to fuse data returned by different DBs**
- **Need to browse query responses to correct inconsistencies**

- **Mediators and Name Services**
- Stand between user and heterogeneous databases
- National name service to find databases, mediators
- **Search engines**

- **Security: authentication**
- In heterogeneous, distributed world, multiple authenticators of varying trustworthiness

- **Scaling to hundreds, thousands of sites**