Trends and Challenges in Database Development

Ellen Munthe-Kaas
Database Trends

1. The object/relation battle
2. Web Services
3. Queues and workflows
4. Cubes and online analytic processing
5. Data mining and machine learning
6. Column stores
7. Approximate answers
8. Semi-structured data
9. The Semantic Web
10. Stream and sensor processing
11. Smart objects: Databases everywhere
12. Publish-subscribe
13. Massive memory, massive latency
14. Self managing and always up

A selection of these subjects (and some additional ones) are discussed more thoroughly in the rest of the course.
1. The Object/Relation Battle

- Objects? Relations? **Object-Relational!**
- The Object-Relational world:
  Marry programming languages and DBMSs = ORDBMSs
  - Stored procedures evolve to "real" languages
    Java, C#,... with real object models
  - Encapsulated data: a class with methods
  - Tables are enumerable and indexable record sets with foreign keys
  - Records are vectors of objects
  - Opaque or transparent types
  - Set operators on transparent classes
  - Ends Inside-DB Outside-DB dichotomy
Example: Skyserver

Astronomy data online

- Maps half of the northern sky
- For research on astronomical observations

```sql
SELECT TOP 1000
g.run, f.field, p.objID
FROM
TARGDR4..PhotoObj p,
TARGDR4..Field f,
TARGDR4..Segment g
WHERE
f.fieldid = p.fieldid
and f.segmentid = g.segmentid
and f.psfWidth_r > 1.2
and p.colc > 400.0
```
Skyserver
Data Release 5 (DR5)

- Imaging catalog:
  - Footprint area: 8000 sq. deg.
  - 215 mill. unique objects
  - Data volume:
    - images: 9.0 TB
    - catalogs (data archive server): 1.8 TB
    - catalogs (SQL database): 3.6 TB

- Spectroscopic catalog: 1,048,960 spectra
  - 5740 sq. deg.
  - 674,749 galaxies
  - 79,394 quasars (redshift < 2.3), 11,217 quasars (redshift > 2.3)
  - 154,925 stars
Skyserver Home Page

http://skyserver.sdss.org/

Projects

In our SkyServer Projects, you will learn science by studying the 120 million stars and galaxies of the Sloan Digital Sky Survey (SDSS) - the same objects that professional astronomers study. Most of these objects have never been seen before by human eyes.

STUDENTS:

- Register as a SkyServer student user
- Request answers to SkyServer projects
- View answers you have requested
- Evaluate a project you have finished

TEACHERS:

- Learn how you can use SkyServer in your classroom
- See the teacher’s guide for SkyServer projects
- Register as a SkyServer teacher

Work through these projects at your own pace. Each consists of several questions and exercises. If you get stuck, try reading our About Astronomy or About SDSS pages. Each project ends with a research challenge, which lets you do real astronomy research, just like thousands of professional astronomers around the world. When you finish the research challenge for each project, e-mail it to us. We’ll look over all the results we get, and we’ll put the best up on these pages!

Click on one of the following project categories to get started:
2. Web Services

- **Web service**: SW system designed to support interoperable machine-to-machine interaction over a network
  - Web servers and runtime (Apache, IIS, J2EE, .NET) displaced TP monitors and ORBS
  - Web services (soap, wsd1, xml) are displacing current brokers
  - DBMS listening to port 80
    - Publishing WSDL, DISCO, WS-Security
    - Servicing SOAP calls
    - **DBMS is a web service**
  - Basis for distributed systems
  - A consequence of ORDBMS
Example: OpenSkyQuery

• Cross-matching astronomical catalogs
  – 29 archives
  – spatial data search
  – Raw Pixel data live in file servers
  – Catalog data (derived objects) in database
  – Online SQL
  – Based on Web Services

• Also used for education
  – 150 hours of online astronomy
  – implicitly teaches data analysis
Welcome to Open SkyQuery!

OpenSkyQuery allows you to cross-match astronomical catalogs and select subsets of catalogs with a general and powerful query language. You can also import a personal catalog of objects and cross-match it against selected catalogs.

To get started, go to the query page and look at some of the samples. Please note there are some limitations to this system.

This site is best viewed at 1024x768 or greater with JavaScript enabled.
Example Query: Brown Dwarf Search

```
SELECT o.objId, o.ra, o.dec, o.type, t.objId, t.j_m, o.z
FROM SDSSDR2:PhotoPrimary o, TWOMASS:PhotoPrimary t
WHERE XMATCH(o, t) < 2.5 AND Region('CIRCLE J2000 16.031 -0.891 30') AND (o.z - t.j_m) > 2
```
3. Queues and Workflows

• Applications loosely connected via queued messages

• Queues:
  – Supported in all major database systems
    • defining queues
    • queueing and dequeueing messages
    • attaching triggers to queues
  – Basis for publish-subscribe and workflow

• Challenges: How to structure workflows and notifications; characterize design patterns
4. Cubes and Online Analytical Processing

- OLAP: Approach to quickly provide the answer to analytical queries that are dimensional in nature
  - sales
  - marketing
  - management reporting
  - ...
- Databases for OLAP contain data cubes
  - Data cubes now standard
  - MDX is very powerful (Multi-Dimensional eXpressions)
  - Cube stores cohabit with row stores
    - ROLAP + MOLAP + xOLAP
      (relational + multidimensional ++ ...
       OnLine Analytical Processing)
    - Very sophisticated algorithms
- Challenge: Better ways to query and visualize cubes

SELECT <axis_spec>
FROM    <cube_spec>
WHERE <slicer_spec>
5. Data Mining and Machine Learning

- Tasks: Classification, association, prediction
- Tools: Decision trees, Bayesian networks, a priori clustering, regression, neural nets, ...
- Now unified with DBs
  - Create table T(x,y,z,a,b,c)
    Learn ”a,b,c” from ”x,y,z” using <algorithm>
  - Train T with data
  - Then can ask:
    - Probability (?x,?y,?z,?a,?b,?c)
    - Probability (x,y,z,?a,?b,?c)
  - Example: Learn height from age
- Challenge: Better learning algorithms
Data ....

- **Data mining**
  - Process of automatically searching large volumes of data for patterns
  - Applies computational techniques from statistics, information retrieval, machine learning and pattern recognition

- **Data farming**
  - Process of using a high performance computer or computing grid to run a simulation thousands or millions of times across a large parameter and value space
  - Result is a “landscape” of output that can be analyzed for trends, anomalies, and insights in multiple parameter dimensions.

- **Data warehouse**
  - Collection of computerised data organised to most optimally support reporting and analysis activity

- **Data mart**
  - Specialized version of a data warehouse for specific user groups or needs
Data Mining – Database Synergy

• Create the model: Learn height from Gender + Age
  CREATE MINING MODEL HeightFromAgeSex
    (ID long key,
     Gender text discrete,
     Age long continuous,
     Height long continuous PREDICT)
     USING Decision_Trees

• Train a data mining model: Database verbs to drive Modeler
  INSERT INTO Height
    SELECT ID, Gender, Age, Height
    FROM People

• Predict height from model: Probabilistic reasoning
  SELECT height,
     PredictProbability(height)
  FROM Height PREDICTION JOIN New
    ON New.Gender = Height.Gender
    AND New.Age = Height.Age
6. Column Stores

- Universal relations: Users see fat base tables
- Ex. LDAP
  - 7 required, thousands of measured attributes
- Conceptually simple, but use only some columns
- To avoid reading useless data,
  - do vertical partitions
  - define 10% popular columns index
  - make many skinny indices
  - query engines uses covering index
  - much faster read, slower insert/update
- Column stores automate this
- Challenge: Automate design
7. Approximate Answers

- "Messy" data types: Text, time, space
  - Integrating programming languages with DBMS allows adding data types and libraries for indexing and accessing such data
  - Approximate answers
  - Probabilistic reasoning
  - No clear algebra
8. Semi-Structured Data

"Cyberspace is a giant XML document: xQuery for manipulation"

- Not all data fits into the relational model
  - XML – eXtensible Markup Language
- File directories are becoming databases
  - Pivot on any attribute
  - Folders are standing queries
  - Freetext + schema search (better precision/recall)

"Structure: YES! Semi-structured: NO!"
9. The Semantic Web

- Today’s World Wide Web content:
  - Designed for humans to read
  - Can be parsed for layout and routine processing
  - Data hidden in HTML files:
    Useful in some context, but useless in others
  - Consequence: Low precision
    - Ex.: Search for birds of family Panurus, only knowing its English name....

- How to obtain more precision?
  - Adding semantics to the World Wide Web
Adding Semantics to WWW

• Documents "marked up" with semantic information
  – Extension of HTML `<meta>` tags
    • Machine-readable information (metadata) about human-readable content of the document
    • "Pure" metadata representing a set of facts

• Common metadata vocabularies (ontologies)
  – For marking up documents in an agreed way

• Automated agents
  – Perform tasks for users of the Semantic Web
  – Use provided metadata

• Web-based services
  – To supply information specifically to agents
    • E.g., a Trust service: Has an online store a history of poor service or spamming?

• Database support needed!
Standards and Tools

- **URI** – Uniform Resource Identifier
  - For identifying resources uniquely
- **XML** – eXtensible Markup Language
  - Surface syntax for structured documents
  - No semantic constraints on the documents
- **XMLS** – XML Schema
  - Language for restricting the structure of XML documents
- **RDF** – Resource Description Framework
  - Simple data model for referring to resources and how they are related
  - An RDF-based model can be represented in XML syntax
- **RDFS** – RDF Schema
  - Vocabulary for describing properties and classes of RDF resources
  - Data model for class hierarchies
- **OWL** – Web Ontology Language
  - Vocabulary for describing further class and relationship properties
Example: Museo Suomi

- The Portal MuseumFinland: Finnish museums on the Semantic Web
  - Making cultural collections available and semantically interoperable through WWW

http://www.museosuomi.fi/
Topic Maps

• Standard for representation and interchange of information
  – Provides a model and grammar for representing the structure of information resources
• Emphasis on findability of the information
• XTM – XML Topic maps
  – XML-based interchange syntax
• Not a language for providing formal ontologies like RDF and OWL
  – Deliberately supports inconsistencies!
Example: **Apollon**

- University of Oslo’s popular science magazine
  - Paper version four times a year
  - Web resource
    - Semantic portal using Topic Map technology
    - Associative links for easy cross-article, topic-based browsing and search
Apollon Portal

http://www.apollon.uio.no/

INF5100 Autumn 2006
©Ellen Munthe-Kaas
10. Stream and Sensor Processing

• Data generated by instruments that monitor the environment
  – Need to process/analyze streams of data
  – Traditionally:
    • Query large amounts of facts
  – Streams:
    • Large amounts of queries on each new fact
Streams

• Implications:
  – New aggregation operators
  – New programming style
  – Streams in products:
    • Queries represented as records
    • New query optimizations

• Lots of challenges
  – Data structures, query operators, execution environments are qualitatively different from classical DBMS architectures
Sensor Networks

Base station (gateway)

Motes (sensors)
Sensor Network Characteristics

• Autonomous nodes
  – Small, low-cost, low-power, multifunctional
  – Sensing, data processing, and communicating components

• Sensor network is composed of large number of sensor nodes (motes, smart dust)
  – Proximity to physical phenomena
    • Deployed inside the phenomenon or very close to it

• Monitoring and collecting physical data
  – Streams of data

• No human interaction for weeks at a time
  – Long-term, low-power nature
Sensor Data Harvesting

- Optimize wrt. power and bandwidth
  - Push queries out to sensors
    - Moving intelligence to the periphery of the network
    - Every mote and smart dust a small database in itself
  - Aggregate results during data collection
- Much more dynamic query optimization strategies needed
11. Smart Objects: Databases Everywhere

- Phones, PDAs, Cameras, ... have small DBs
  - even motes
  - and smart dust?
- Disk drives have enough cpu, memory to run a full-blown DBMS
- All these devices want/need to share data
- Need a simple-but-complete DBMS
  - They need an "Esperanto": a data exchange language and paradigm
- Billions of clients, million of servers
12. Publish-Subscribe

• Data with many users
  – Data warehouses collect vast data archives and publish subsets to special interest group data-marts
  – Replicas for availability and/or performance
  – Mobile users do local updates, synchronize later

• Publish-subscribe model:
  – Custom subscriptions installed at the warehouse
  – Real-time notification
Publish-Subscribe and Stream Processing

• Compare publish-subscribe & stream processing systems:
  – Millions of standing queries (subscriptions) compiled into dataflow graph
  – At arrival of new data, incrementally evaluate dataflow graph

• Challenge:
  – Support more sophisticated standing queries
  – Better optimization techniques
13. Massive Memory, Massive Latency

- 2005: RAM costs $\sim 100k\text{€} - 300k\text{€} per TByte
- Main-memory databases!
- Latency a problem
  - TByte ram memory scan $\sim$ minutes
  - TByte disk scan $\sim$ hours
- NUMA latency a problem
- Challenge: Algorithms for massive main memory
  - Database engines need to overhaul their algorithms
14. Self Managing and Always Up

- No DBAs for cell phones or cameras
  - nor for panel heaters, washing machines,...
- Self* is the key
  - Self-managing
  - Self-configuring
  - Self-organizing
  - Self-healing
  - Self-...
- Requires a modular software architecture
  - Clear and simple knobs on modules
  - Software manages these knobs
But What Happened to the Classical Databases?

- Classical databases are alive and kicking!
- Classical requirements are still valid!
  - Persistent data management
  - Concurrency control
  - Availability / fault tolerance
  - Ad-hoc queries
  - Data integration
  - Logical and physical data independence
  - Data consistency
  - Data security
  - Distribution
  - Performance
  - Extensibility
  - Cost effectiveness
  - Simple manageability
- Classical database application domains and classical database functionality do NOT disappear
Classical Application Domains

• Traditional database technology emerged from bookkeeping and banking
  – Administrative/business data processing
  – Production management
  – CASE
  – CAD/CAM
  – ERP
  – CRM
  – ...

• Relational databases stand firm
Emerging Application Domains

- Data warehousing
- OLAP
- Data mining
- GIS
- eCommerce
- Multimedia databases
- Mobile databases
- Scientific databases
  - Medicine
  - Astronomy
  - Biology
  - Seismology
  - Meteorology
  - Music
Database Trend Summary
According to Subject

1. Database system implementation
2. Database model
3. Interaction model
4. Data accessibility
5. Data yellowpaging
6. Accessing the data
7. Data processing
8. Database integration
1. Database System Implementation

More efficient DBMSs

- Column stores
  - Dealing with sparse data in an efficient way
- Stream and sensor processing
  - Dealing with severe power, bandwidth, and memory restrictions
  - Dealing with evolving data
- Massive memory, massive latency
  - Dealing with new memory and disk technologies
2. Database Model

Data model paradigm. Interaction model

• The object/relation battle
  – The object-relational model
  – ”Real” programming languages
• Semi-structured data
  – Liberation from the O/R model
• Smart objects: Databases everywhere
  – Common data model
• Approximate and probabilistic reasoning
  – Models for data that cannot be expected to provide exact answers
• Stream and sensor processing
  – Model for evolving data
• Cubes and OLAP
  – Multidimensional data model
  – Data organized for fast complex analytical and ad-hoc queries
• Data warehouses and data marts
  – Multidimensional model
  – Data organized for optimally supporting reporting and analysis
• Queues and workflows
  – Workflows rather than RPC style interaction model
• Publish-subscribe
  – Data dispatched according to a push model
3. Data Accessibility

Preparing data for access
- making sure relevant data is near by

• Web services
  – Realizing federated heterogeneous systems

• Queues and workflows
  – Using queues to obtain more loosely coupled systems

• Data warehouses and data marts
  – Collections of derived data

• The Semantic Web
  – ”Explaining” data through metadata
4. Data Yellowpaging

What kinds of data are available
How to obtain it

• Web services
  – Announcing available data
  – Describing how to obtain it

• The Semantic Web
  – Ontologies describing data semantically
5. Accessing the Data

Getting hold of the data

• Queues and workflows
  – Asynchronous communication
  – Realizing delay-tolerant networks

• The Semantic Web
  – Automated agents performing tasks for users
  – Web-based services supplying info to agents

• Publish-subscribe
  – Dispatching evolving data
6. Data Processing

Climbing the value chain
- from data to information to knowledge to wisdom

- Cubes and OLAP
  - Fast data analysis
  - For better business decisions

- Data mining and machine learning
  - Knowledge discovery in databases

- Data farming
  - Simulations for better analysis

- Approximate answers
  - Text retrieval, spatio-temporal data analysis
  - Approximate and probabilistic reasoning

- The Semantic Web
  - Obtaining higher precision and quality on data retrieval
7. Database Integration

Utilizing database functionality in larger systems

• Queues and workflows
  – Supporting business processes
  – Part of ERP systems

• Data warehouses and data marts

• Self managing and always up
  – Embedded databases
  – Simplifying life for the uninformed user
Literature

• Jim Gray: *The Next Database Revolution*, *Proc. 2004 ACM SIGMOD International Conference on Management of Data*

(Available through the ACM Digital Library; cf. http://x-port.uio.no)