Object-Oriented & Component-based Software Development

Programming Style, classes, methods, UML

Interface, polymorphism, encapsulation, inheritance, AOT

Frameworks

1. Multi-threading, I/O, N/w

Design Patterns

6. Typing & Memory management

Software engineering

Component-based programming

Java beans, EJB components

Web services components

Fundamentals of Computing and Computer Systems

Digital Logic, Arithmetic

Sets, Theory of Computation

Data Structures

Algorithms

Computer Architecture - Instruction sets, CPU, RISC, CISC

Operating System - Threads, processes

OS: concurrency, memory management

OS: I/O, File Systems

ADVERTISE IN DATA MANAGEMENT

1. Relational model, SQL

→ Databases: collection of data; shared

→ DBMS: set of prog. for managing database

→ Database system: database + dbms

→ 3 schema architecture: physical design + conceptual design + views

→ Physical & logical independence

→ DDL + DML

→ Objectives:

- Controlling redundancy
- Data structuring + efficient access
- Simultaneous access
- Indexing
- Security / integrity / backup (ref, req, str)

→ Relational model: tuple, column, domain
- Key: rows
- Relational algebra
DATA INTEGRATION & WAREHOUSING

1. Introduction & basics, web data models
   - why? apps need to work with data from diff sources
   - differences: data models, schema, entities, duplication, conflicts
   - Proven & consolidation in order to provide:
     - correctness & completeness
     - Single data model & schema
     - 1 entity rep
     - no conflicts

2. Heterogeneity
   - Tech: means to access data
   - syn: encoding

3. Structured data
   - SQL
   - NoSQL

4. Key-Value stores
   - store data indexed by keys
   - update/lookups ops
   - doc store
     - complex, nested data models
     - primary index
     - data can be more flexibly partitioned

5. Wide (rel) stores
   - records extended by attrs
   - parallel & horizontal partitioning of data

6. Graphs
   - relations = entities
   - graph specific indexes, etc

7. NewSQL
   - data fits into MS
   - logging as persistent DB
5. Data preprocessing - normalize
gather data, normalize values, apply sim measures
combine scores, decide match/non-match

6. Data cleaning, quality & fusion
- records have diff levels of knowledge,
  views, intentions
- info is wrong, biased, outdated, incomplete,
  inconsistent
- depends on context on how to use it
- data provenance
  - info on how data was produced
  - simple vs full prov chains
    who
    - publish prov info on web
      - vocab
    - timestamp
      - metadata
- Data Quality
  - fitness func
    - enterprise web contact
    (control) / (no control)
  - dimension
    - content tech intellecual property

7. Learning matching models
- training data & feature generation
- convert each training ex to pair (v,x)
  & apply learning alg.
DATA MINING

3. Data Mining Process, Preprocessing

- Extraction of useful patterns from data source, which are valid, novel etc.

- Basic tasks:
  - Classification, clustering, association rule mining, regression

- Processes:
  - Understand application domain
  - Select sources, collect data
  - Preprocess data
  - Data mining
  - Post-processing
  - Incorporate them

- Application:
  - Marketing
  - Engineering
  - Web

- F-Score

- ANOVA: Analysis of Variance
  - Two or more samples
  - Between/within group variance
  - Independence of observations
  - Normally
- distance of 2 clusters
  - single
  - complete
  - average unweighted
  - average weighted
  - centroid
- distance function
  - euclidean:
  - manhattan
  - chebychev
- document
- cosine
- data standardization
  - z-score: mean range value
  - interval-scaled
  - ratio-scaled
- trial/ error
- evaluation
  - confusion matrix, precision, recall, f-1
  - entropy, purity
  - intra-cluster cohesion / inter-cluster separation
- classification
  - learn a target function that can predict values
  - training / testing
  - distance of training test data are same
  - decision tree
    - NP hard: best tree
    - convert data into rules
    - recursive partitioning based on information gain
    - overfitting
    - pruning
- evaluation of classifiers
  - accuracy
  - holdout set, n-fold cross validation
  - precision, recall, f-1
  - score (prob that x belongs to two class)
4. Knowledge Representation and Logic

- Knowledge-based agent
  - Knowledge-base
    - KB

- Propositional Logic
  - Connectives
    - AND, OR, NOT

- Inference
  - Direct method
  - Model checking

- First Order Logic
  - Objects, relations (arity), concepts, properties, functions
  - Quantifiers
    - ∀ (for all), ∃ (there exists)

- Equivalence
  - Corresponds to propositional logic (complete, consistent)
  - Generalized resolution
  - Horn clauses
  - Resolution (convert to CNF, sound if # literals in head = # quantifiers)
Map Reduce & Hadoop

- Divide & Conquer
- Scale-out, hide sys details from programmers
- Sync co-workers
  (comm + shared resource access)
- Prag models lower (mpi, threads)
- DF: master/slave, prod-ones
  Shared worker queues

- Map - shuffle & sort - reduce
  Extract | Group | Aggregate

- MR - combine handles
  - scheduling
  - data distribution
  - sync
  - concurrency faults
  - coordination & dealing with failures

- DFS + LFS
7. Data Management, Link Analysis, information retrieved in the cloud

- gekoDBms v/s MR
- multipurpose
- fault tolerant
- native data

Beyond MR:
- BSP
- processors + cores
- supersteps - local map + count
- barrier sync
- pregel, graph, hama

DB workloads
- OLTP
  - real-time, small queries, random reads, write
- OLAP
  - batch, less concurrency, complex query,
  - large scan for query
  - hadoop: cheaper than parallel files
  - rel alg ops: proj, set, join, cartesian, union,
    - diff, group by, rename
    - proj: map over tuples
    - set: map over tuples
    - group by: reduce
    - join: map-side, reduce-side, in-memory (hash)
    - Hive, Pig
SEMANTIC WEB TECHNOLOGIES

1. Introduction, Layers
   - goble & foundations of semantic web
   - today: social web
   - computers do present, humans do linking & interpretation
   - complex queries involving background knowledge
   - web service
   - agents
   - location info in data repositories
   - make implicit knowledge explicit
   - add context info in a std way to make it w/e readable
   - set is a set of technologies to realize web of data
   - formal w/e understandable language, rules, ontologies

-> Index construction
   - map over all docs, emit term as key (doc, tf) as value
   - sort/shuffle - group postings
   - reduce - gather, sort, emit postings

-> Retrieval
   - MR optimized for throughput, low latency, most stat is exp, not on real-time
   - partition of terms/docs
   - replication, caching
   - routing

-> Link analysis
   - key: node
   - value: adjacency list
   - map: adjacency list
   - sort/shuffle: group distances by reachable nodes
   - reduce: set min distance for each reachable node
ontology: explicit specification of a conceptualization

syntax: uri
xml + namespace + xsd + saxon
rdf + rdfs
ontology: uses
logic
proof
trust

HTML: is about presentation

XML: more structure (nesting)
    content/formation (data exchange)
    free model
    DTD/XMLSchema
namespace
    XSL
    XSLT XSL-P
not powerful enough for SW

2. RDF

- subject, predicate, object (mathematical logic)
  ↓
  domain, range (n-ary relation)
  ↓
  resource description framework
  ↓
  s, p, o all are units
  ↓
  triples, xml syntax
  ↓
  blank node
  ↓
  container elements
  - bag
  - seq
  - alt
  ↓
  validation
  - strict, alt, strict
- adapter
  - resolve incompatible interfaces

- visitor
  - define new p w/o changing class
    - only 1 instance of a class is created

- singleton
  - for an object to control references to it

- command
  - encapsulate requests to it

- factory
  - create objects w/o exposing instantiation logic to clients

6. Typing, memory management

- Types vs. untyped languages
  - for every operation, types of data
    - which it is applicable
  - weak/strong typing
    - type-safe
    - prevent value of one type to be treated as another
  - static/dynamic typing
    - compile-time
    - run-time
      - (no-type for variable) during prog

- Activation records: allocated when proc is entered & deallocated when proc is exited
  - info needed by single proc
- interchangeable
- high composability
- goals:
  - manage complexity
  - change
  - re-use
- specification, interface, implementation
- install a component -> object
  - one
- many
- principles: decomposition, abstraction, reusability, software dependable, productivity, standardization
- $C = \{E, M, M, E, I\}$
  - id
  - module
  - event
  - interface
    (visible, type, access, parameter)
- infrastructure: component model
  - component model
  - configuration model
  - deployment model
- contracts

10. Java beans, EJB
- Java beans
- software component model per java
- properties
  - simple (appearance & behavior)
  - bound (value change, notifications)
  - constrained (value changes can be vetoed)
- indexed (null-value)
- events
  - introspection
  - bean info
  - persistence
    via serialization
- deadlock
  - conditions
    - mutual exclusion
    - hold & wait
    - no pre-emption
    - circular wait
  - resource allocation graph
  - deadlock prevention/avoidance
    - banker's algorithm
      - (claim matrix, resource matrix)

- Memory Management
  - program-relative address - actual address
    - register: base & band
  - partitioning, placement
    - fixed, dynamic, next-fit
      - internal fragmentation
  - virtual memory
    - paging (replacement, locality, thrashing)

8. Operating Systems - I/O, File Systems
- devices
  - data rate, unit data error rate, conditions
  - device controllers
  - I/O
    - msg/interrupt, DMA
  - device drivers
    - buffering
      - block, stream
      - disk performance parameters
        - seek, rot, access, time, latency, time
  - disk scheduling
    - FIFO, SSTF, SCAN
- RAID
- disk cache, LRU