Object Oriented & Component based Software Paradigm
rograming Style, classes, methods, UML
Interfaces, polymorphism, encapsulation, inheritance, ADT
Frameworks
Design Patterns
Softwareengg-I
Component based
Java beans, EJB components
Web services components

4. Multithreading, i/o, n/w
6. Typing & Memory
8. Software

Fundamentals of Computing and Computer Systems
Digital Logic, Arithmetic
Sets, Theory of Computation
Data Structures
Algorithms
. Computer Architecture - instruction sets, CPU, Risc, cisc
~~Superscalar~~
. OS - threads, processes
. OS - concurrency, memory management
. OS - I/o, File Systems

ADVANCES IN DATA MANAGEMENT

1. Relational model, SQL

→ Database: collection of data; shared
→ dbms: set of prg for managing database
→ database system: database + dbms
→ 3 schema architure: physical design + conceptual design (data model)
+ views
→ physical & logical independence
→ DDL + DML
→ objectives: - controlling redundancy
- data structuring & efficient access
- simultaneous access
- interfaces
- security / integrity / backup
↓
(ref, val, structural)

→ relational model: - table, tuple, column, domain (schema)
- key: rows
- relational algebra

→ Key - value stores
- store data indexed by key
- ins / del / look up ops

→ doc stores
- Complex, nested data models
- P, 2°
- data more flexibly partitioned

→ wide col stores
- records extended by attrs
- parallel & horizontal partitioning of data

→ graph dbs
- relations == entities
- graph specific indexes, QL etc

→ NewSQL
- data fits into MM
- logging on persistent db

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

→ Process of consolidation in order to provide -
- correctness & completeness
- single data model & schema
- 1 entity rep
- no conflicts

→ Heterogeneity
- Tech: means to access data
- Syn: encoding

- data preprocessing - normalize
- gather data, normalize values, apply sim measures, combine scores, decide match / non-match

→ blocking
- partition records into buckets
- sort :- generate key, sort by key

→ evaluation
- ground truth
  TP, FP, FN, TN
- precision, recall, F-1

→ learning matching models
- training data & feature generation
- convert each training ex to pair (v, c) & apply learning alg.

6. Data cleaning, quality & fusion

→ Providers have diff levels of knowledge, views, intensions
→ info is wrong, biased, outdated, incomplete, inconsistent
→ depends on consumer on how to use it
→ Data Provenance
- info on how data was produced
- simple v/s full prov chains
  who, process
- publish prov info on web
  - vocabs
  - timestamp
  - metadata

→ Data Quality
- = fitness for use
- enterprise / web context
  (control) (no control)
- dimensions
  content, tech, intellectual, formatting

- significance level
- Errors

type I (reject null when true) | type II (fail to reject null when false)

- Tests

two-tailed (true mean is diff from claimed) | upper-tailed (true mean higher) | lower-tailed (true mean lower)

- F-score
- ANOVA: analysis of variance
  - two or more sample
  - between / within group variance
  - independence of observations / normality

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

→ process:- understand application domain
select sources, collect data
preprocess
data mining
post processing
incorporate them

→ applications:- marketing
engineering
text
web

- distances of 2 clusters

Single Complete avg centroids

- distance functions
  - euclidian ~~functi~~
  - manhattan
  - chebychev
  - document
  - cosine
- data standardization
  - attrs have a comon range value
  - interval-scaled
  - ratio-scaled
- trial / error
- evaluation
  - confusion matrix, precision, recall, f-1
  - entropy, purity
  - intra cluster cohesion / inter cluster separation

7. classification

- learn a target function that can predict values
- training / testing
- distance of train & test data ues same
- decision tree
  - NP hard best tree
  - convert to set of rules
  - recursive partitioning based on infomation gain
  - overfitting

Prepruning Postpruning

- evaluation of classifiers
  - accuracy
  - holdout set, n-fold cross validation
  - precision, recall, f-1
  - score (prob that ex belongs to +ve class)

2. Search strategies

→ Concepts
- State space
- path
- frontier (reachable)
- solution (goal test)

→ Algos
- tree search (exp)
- graph search (avoids repeated states
  detected queue
  (memory itself)

→ Uninformed Search
- strategy for expansion
- evaluation - completeness, space/time complexity, optimality (branching factor, depth)

- Algos
  - BFS → FIFO
  - DFS → LIFO
  - Depth limited (iterative deepening) (DFS with limit)
  - Uniform cost (dijkstra with no goal)
  - Bidirectional

→ Informed Search
- node expansion based on some estimate of distance to goal (best first search)
  ↓
  heuristic fn to distance to goal
- greedy bfs → heuristic fn is used for expansion
- A^* → $g(n) + h(n)$

- Local search & optimization
  - hill climbing (gradient ascent or descent) (local maxima)
  - Simulated annealing
    - accept bad moves with prob
  - genetic algorithms (random, mutate, crossover, fitness)

→ Game playing AI
- Games as search
  - game trees
  - evaluation function (for board position)
- Adversarial search
  - min max (best move with ease)
  - alpha-beta pruning — Improve

ates)
complexity
tor, depth)
Bidirectional

4 Knowledge representation and Logic

- knowledge based agent
  - knowledge base
  - rep
    - propositional logic
    - first order logic
  - reasoning via inferencing
- Logic
  - sentence
    - syntax (well formed)
    - semantics (meaning)
  - model: true/false
  - entailment
  - inference — alg is sound, complete
- Propositional logic
  - Syntax
    ¬, ∨, ∧, →, ↔, T, F

- inferencing
  with rules
  like {a → b, a} ⊨ {b}
  - |KB|²
  - linear with horn clauses
    (almost 1 +ve literal)
- First Order Logic
  - objects, relations (arity), constants
    properties, functions
  - quantifiers
    - ∀ (small) universal
    - ∃ (reverse) existential
  - equality
  - closed world assumption
  - inferencing
    - convert to propositional logic (complete, inefficient)
    - generalized modus ponens (complete (semidecidable))
    - Fwd / Bckwd chaining
    - resolution
      (convert to CNF, same as PL but without quantifiers)
      by Skolemization

name node in main memory

- data node
  - block server, block reports
  - pipelining
    - block placement, data coredness, failures,

- distributed access structures
  - indexing

tree hash

- consistent hashing - distributed hash directories
- Amazon dynamo - BTree
- Bigtable - BTree + non-dense index files

↓

data model: row + key (tablets)

5. Map Reduce & Hadoop

→ divide & conquer
→ scale-out, hide sys details from programer
  - sync workers (comm + shared resources access)
  - prg models lower (mpi, threads)
  - DP :- master/slave, prod-cons, shared worker queues

→ map - shuffle & sort - reduce

extract group aggregate

→ MR - runtime handles
  - scheduling
  - data distribution
  - sync
  - error & faults
  - coordination & dealing with failures

→ DFS + LFS

→ value-key inversion

- sort by value

→ Beyond MR:

- BSP

- processors + communication

- supersteps - local comp + comm + barrier sync

- pregel, giraph, hama

7. Data Management, Link Analysis, information retrieval in the cloud

→ RDBMS v/s MR

RDBMS → multipurpose, ACID, SQL

MR → fault tolerant, native data

→ DB workloads

- OLTP

real-time, small queries, random reads & writes

- OLAP

batch, less concurrency, complex query, large scans for query

- hadoop : cheaper than parallel dbms

- rel alg ops: prj, sel, join, cartesian, union, diff, groupby, rename

- prj : map over tuples

sel : map over tuples

groupby : reduce

join : map-side, reduce-side, in-memory (hash)

- HIVE, PIG

→ Index construction

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

→ Retrieval

- MR optimized for throughput
  latency, mr start is exp, not for real time
- partitioning of terms/docs
- replication, caching
- routing.

→ Link analysis

key: node n
value: d adjacency list
map: adjacency list
sort/shuffle: group distances by reachable nodes
reduce: sel min distance for each reachable node

SEMANTIC WEB TECHNOLOGIES

1. Introduction, Layers

→ goals & foundations of semantic web

→ today: syntactic web
computers do presentation, humans do linking & interpretation
impossible tasks
- complex queries involving background knowledge
- web service
- agents
- location info in data repositories

→ make implicit knowledge to explicit

→ add extra info in a std way to make it m/c readable

→ SW is a set of technologies to realize web of data

→ formal m/c understandable languages, rules, ontologies

ntology: explicit specification of a conceptualization

ayers: unicode + uri
xml + namespace + xmlschema
rdf + rdfs
ontology vocab
logic
proof
trust

HTML: is about presentation
XML: more structure (nesting)
content / formatting (data exchange)
tree model
DTD / XMLs
namespace
XSL
XSLT xpath XSL-FO

not powerful enough for sw

2. RDF

→ subject, predicate, object (mathematical logic)
↓
domain, range (unary relation)
(n-ary)

→ resource description framework

→ S, P, O all are uri's

→ triples, xml syntax

→ blank node

→ container elements
- bag
- seq
- alt

→ reification
- stmt abt stmt

→ adapter
- resolve incompatible interfaces

→ visitor
- define new op without changing class

→ singleton
- only 1 instance of a class created

→ proxy
- for an object to control references to it

→ command
- encapsulate request to it

→ factory
- Create objects without exposing instantiation logic to clients

6. Typing, Memory management

→ Typed v/s untyped Languages
- for every operation, types of data for 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 prg.)

→ Activation records : allocated when proc is entered & dealloc when proc is exited
(info needed by single exec of proc)

- interchangeable
- high composibility
- goals:
  - manage complexity
  - change
  - reuse
- specification, implementation, installed component → object
  1-many
- principles: decomposition, abstraction, reusable, S/w dependable, productivity, Standardization
- C = < P, M, E, I >
  - P → id
  - M → module (visible, type, access, parameters)
  - E → event
  - I → interface
- infrastructure: Component model
  Connection model
  deployment model
- contracts

10. Java beans, EJB

Java beans

- software component model for java
- properties
  - simple (appearence & behavior)
  - bound (value change, notified)
  - constrained (value changes can be vetoed)
  - indexed (null-value)
- events
  - introspection
  - bean info
- persistence
  via serialization

- deadlock
  - conditions
    - mutual exclusion
    - hold & wait
    - no preemption
    - circular wait
  - resource allocation graph
  - deadlock prevention / avoidance
    (future process requests)
    - banker algorithm
      (claim matrix, resource matrix)

- Memory Management
  - prg relative addres → actual addres
    → translation
  - registers: base & bound
  - partitioning, placement
    - best fit
    - first fit
    - next fit
    - fixed (internal fragmentation)
    - dynamic
  - virtual memory
  - paging (replacement LRU, tables (TLB), locality of ref, thrashing)

8. Operating Systems - I/O, File systems

- devices
  - data rate
  - unit of transfer
  - data rep
  - error conditions
- device controllers
- I/O
  - prg
  - interrupt
  - DMA
- device drivers
- buffering
  - block
  - stream
- disk performance parameters
  - seek time
  - rot latency
  - access time
- disk scheduling
  - FIFO, SSTF, SCAN
- RAID
- disk cache, LRU