NoSQL Database
Hands-On: R Flow Control
Database Design Project

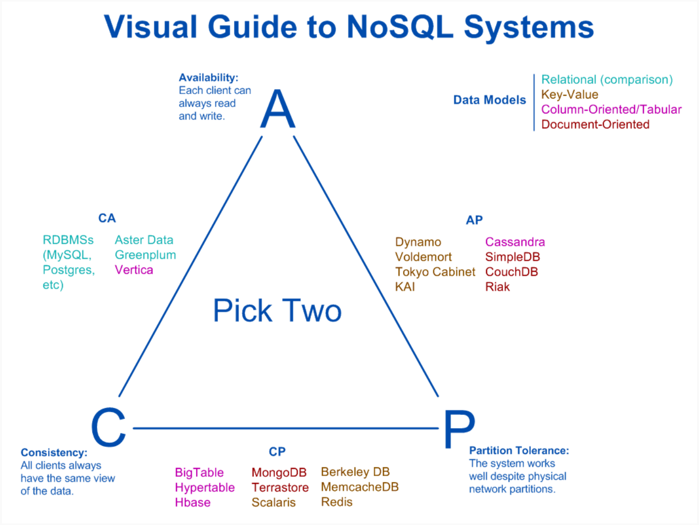
BMI701 Introduction of Biomedical Informatics
Lab Session 4

Wei-Hung Weng

September 26, 2016

HMS DBMI — MGH LCS

Database Selection



Courtesy by digbigdata.com

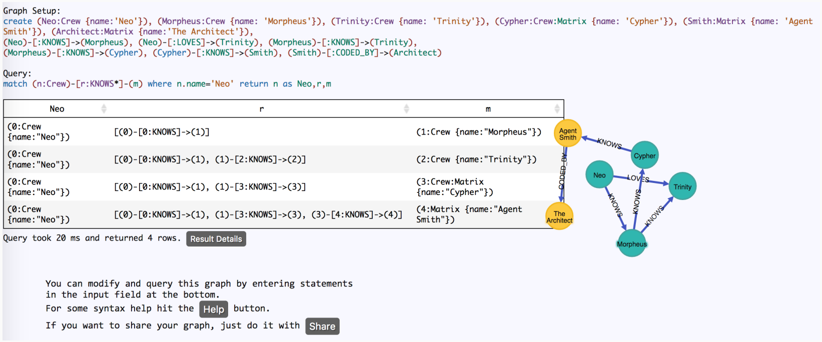
NoSQL

- Key-value Stores
  - Hash table: key: value
  - Redis: data in RAM, not HD (so limited to RAM)
- Column Family Stores
  - Also key-value hash table, but one key may map to different columns (not the single value)
  - Cassandra: super fast write, slow read: e.g. facebook LIKE)
  - HBase: best to run MapReduce with Hadoop/HDFS stack, good for realtime queries (logs)
- Document Databases
  - JSON like document (BSON: binary JSON)
  - Nested: {`Key:`{`Key:Value, Key:Value`}}
  - **MongoDB**: very good for general use
  - CouchDB: can use RESTful HTTP API, use request
- Graph Databases
  - Node / Relation / Property
  - **Neo4J**

MongoDB

- Download and install MongoDB
- Open your terminal
  - `brew update`
  - `brew install mongodb`
  - `mkdir -p  /mongo/db`
  - `mongod --dbpath  /mongo/db`
  - Play with the official sample
- Running MondoDB in R
  - `install.packages("rmongodb")`
  - `library(rmongodb)`
  - Then we can follow the cheat sheet

Neo4J



```
Graph Setup:
create (Neo:Crew {name:'Neo'}), (Morpheus:Crew {name: 'Morpheus'}), (Trinity:Crew {name: 'Trinity'}), (Cypher:Crew:Matrix {name: 'Cypher'}), (Smith:Matrix {name: 'Agent
Smith'}), (Architect:Matrix {name:'The Architect'}),
(Neo)-[:KNOWS]->(Morpheus), (Neo)-[:LOVES]->(Trinity), (Morpheus)-[:KNOWS]->(Trinity),
(Morpheus)-[:KNOWS]->(Cypher), (Cypher)-[:KNOWS]->(Smith), (Smith)-[:CODED_BY]->(Architect)

Query:
match (n:Crew)-[r:KNOWS*]-(m) where n.name='Neo' return n as Neo,r,m
```

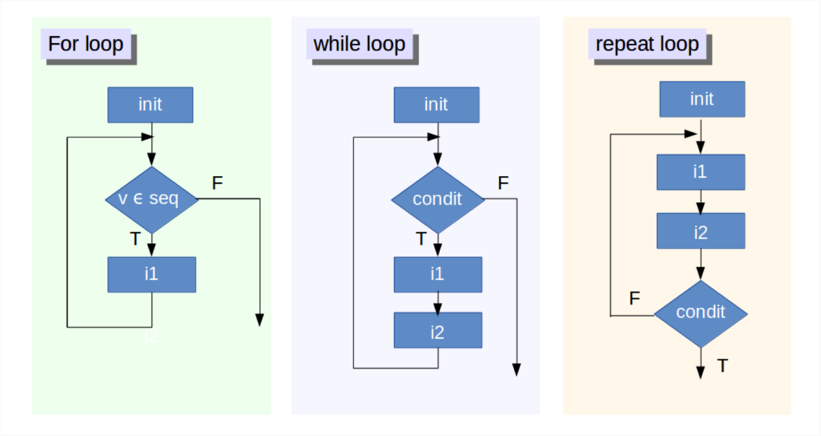
Neo	r	m
(0:Crew {name:"Neo"})	[(0)-[0:KNOWS]->(1)]	(1:Crew {name:"Morpheus"})
(0:Crew {name:"Neo"})	[(0)-[0:KNOWS]->(1), (1)-[2:KNOWS]->(2)]	(2:Crew {name:"Trinity"})
(0:Crew {name:"Neo"})	[(0)-[0:KNOWS]->(1), (1)-[3:KNOWS]->(3)]	(3:Crew:Matrix {name:"Cypher"})
(0:Crew {name:"Neo"})	[(0)-[0:KNOWS]->(1), (1)-[3:KNOWS]->(3), (3)-[4:KNOWS]->(4)]	(4:Matrix {name:"Agent Smith"})

Query took 20 ms and returned 4 rows. Result Details

You can modify and query this graph by entering statements
in the input field at the bottom.
For some syntax help hit the Help button.
If you want to share your graph, just do it with Share

- Download Neo4J
- Neo4J cheat sheet
- Use Neo4J in R

Flow Control



Courtesy by Dr. Mujeeb Basit

- `for, while, if/else, next/break, stop`
- Sample codes

R: Apply Function

- `apply`
  - apply a given function to the rows (index 1) or columns (index 2) of a matrix
- `lapply`
  - apply a given function to every element of a list and obtain a list
- `sapply`
  - apply a given function to every element of a list and obtain a vector
- `tapply`
  - function to process vector subsets

R: More Apply

- `Map()` and `mapply()`
    - iterate over multiple input data structures in parallel
- `mclapply()` and `mcMap()`
    - parallel versions of `lapply()` and `Map()`
    - Only for Linux or Mac
- `clusterApply()` and `clusterApplyLB()`
    - Linux, Mac, Windows

R: Other Useful Packages/Functions

- `foreach`, `parallel`, `doParallel`
- `sqldf`
- `system()`
- Sample codes

Take Home Message

- NoSQL summary
- Flow control / R apply family
- Good luck on your presentation!
- Contact
    - Github repository
    - ckbjimmy@gmail.com
    - Linkedin: Wei-Hung Weng