Cloud & Cluster Data Management

Project Overview

Note: Consult individual handouts for details

Course Project

- The course will be accompanied by a project that is based on a data management scenario, done in groups of 4-5 students
  - **Task 1:** Study question that will take you through a "dry run" of mapping the application to a NoSQL data model and make you think about how to answer some queries using the data.
  - **Task 2:** Pick a specific NoSQL system and compile a systems profile, based on papers and documentation.
  - **Task 3:** Design a management and processing system based on the given application and the previously chosen NoSQL system. This time for real! Plus load a bit of data.
  - **Task 4:** Implement a prototype of the data management and processing system, plus individually do a 2-page paper on the project.
- Not too soon to start forming teams – need 8 teams.

  Four with 4, Four with 5

Data Source: DBLP

DBLP Computer Science bibliography:
http://dblp.uni-trier.de/

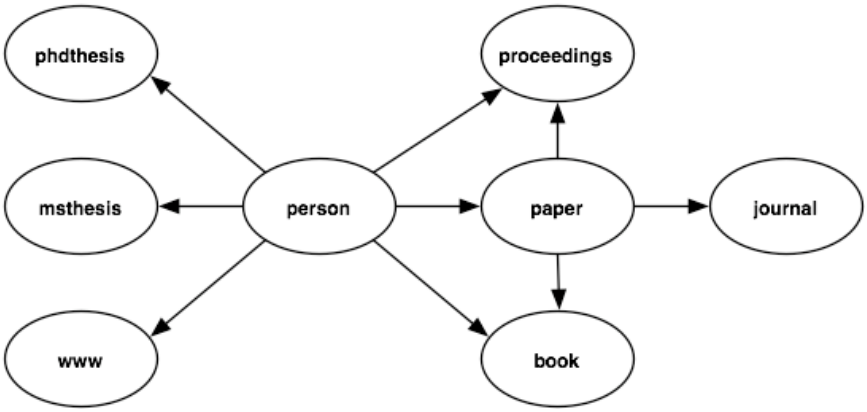
We will actually be using a graph (nodes & links) form of the data, from:
https://kdl.cs.umass.edu/display/public/DBLP

There is an excerpt of the UMass data on SQLShare

- DBLP_OBJECTS_SMALL
- DBLP_LINKS_SMALL
- DBLP_ATTRIBUTES_SMALL

3

Structure of DBLP Graph



Plus attributes of both nodes and links, for example, title of a paper

4

Information is Highly Disaggregated

Info is broken down into small pieces.

- Objects (nodes)
- Links
- Attributes (including object-type and link-type)

You will probably want to group the data for faster and easier access.

You might also want to store information redundantly:

- Papers for authors
- Authors for papers

5

Task 1: Application Design "Dry Run"

- The goal of this project is to complete the following tasks
  - Pick a NoSQL data model and map DBLP data into that model
  - There will be six queries. In words or pseudo-code, describe how you would do the queries listed in the handout for Task 1 (coming soon)
- Deliverable is a written report

6

Task 2: Systems Profile

- Horizontally scalable data management systems
  - Riak (http://wiki.basho.com/)
  - Project Voldemort (http://project-voldemort.com/)
  - CouchDB (http://couchdb.apache.org/)
  - SimpleDB (http://www.amazon.com/simpledb/)
  - HBase (http://hbase.apache.org/)
  - Cassandra (http://cassandra.apache.org/)
  - OrientDB (http://www.orientechnologies.com/)
  - Accumulo (https://accumulo.apache.org/)
- Other options require approval – can,t be a system covered in detail in class.

7

Task 2: System Profile

- Data Model
  - Precise description of the data model, especially in terms of differences from the "standard" models presented in the lecture
  - Detailed summary of the basic data manipulation API, i.e. features to create, retrieve, update and delete data items.
- Query Support
  - Supported query types, i.e. point, range, navigation, arbitrary?
  - What is the query language of the system? Is it declarative, functional, algebraic or imperative?
  - Are queries automatically optimized?
- Indexes
  - What index structures are available?
  - What can be indexed? What can be a key? What can be a value?
  - How are indexes managed, i.e. manually or automatically?

8

Task 2: Systems Profile

- Storage Options
  - Disk or file storage
  - In-memory (RAM)
  - Flash or SSD
  - Traditional database
  - Cloud Storage (GFS, HDFS, S3)
- Transactions and Concurrency Control
  - Does the system support transactions?
  - How are transactions implemented, i.e. locks, OCC, MVCC, etc.?
  - What consistency guarantees are given?

9

Task 2: Systems Profile

- Scalability and Replication
  - What types of replication are supported, i.e. synchronous or asynchronous?
- Platform/Deployment
  - What cloud infrastructures are supported?
  - What deployment scenarios are supported, i.e. embedded, client-server, multi-core CPU, cloud, etc.?
  - Language bindings?
  - Communication protocols, i.e. JSON, REST, etc.?

10

Task 3: Application Design

- Design the example data management problem in the previously profiled system
  - similar to Task 1, but more technical as it is based on a concrete system
  - consider queries specified in the scenario
- Deliverable is a ten-minute presentation in class
  - discuss final design and motivate the design choices made w.r.t. the requirements of the application and the capabilities of the system
  - give details on the mapping of data structures, planned indexes, and query implementation strategies
- Also, load a small sample of the data into the structure you plan

Task 4: Prototype Implementation

- Implement a prototype based on previous design
  - data model (with data loading capabilities)
  - at least four of the queries given
- The goal of this part of the project is to realize a small application and experience its performance in practice
- Deliverable is the developed source code plus a ten-minute presentation during the final slot
- Also, each group member will write a 2-page paper individually on the experience.