Big Query Processor (BQP)

A Project for CSG516 Adv. Databases Systems
Second Semester, 2022-2023 AY.

Brief Description:

In the present era, scientific publications are increasing rapidly, making the network of collaborations, topics, papers, etc., more complex than ever. Not surprisingly, the term Big Scholarly Data has been recently coined to refer to the rapidly growing scholarly source of information (e,g., extensive collections of scholarly data with million authors, papers, citations, figures, tables, as well as massive scale related data such as scholarly networks). The analysis of such data and network is helpful for researchers to identify colleagues working on similar topics, make a profile of a researcher for understanding its research interests based on its academic records and scores, and identify experts on a specific research area.

The DBLP server provides bibliographic information of Scholarly Data on major ***computer science journals and proceedings.*** It is a high-quality digital library with complete coverage of computer science literature. DBLP data if modelled in a graph format would allow several outcomes such as finding experts in the community, community detection, community mining, keyword extraction, etc. Researchers in academia are categorized into communities and characterized by topics, interests, geographical influence, etc. In the DBLP, the community is a significant object of interest. Generally, a community is a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the graph. Relation among the entities can be represented in a graph format using Neo4j. Specifically, Neo4j is a graph database. It models attributes, labels, and directed multi-graphs. Neo4j makes use of the declarative Cypher language for querying the graph-store.

Recent studies state a growing interest in studying and understanding the network of scholars/researchers to find research experts, trending topics, influencing scholars, etc., by searching scholarly data in the graph format. In view of providing a tool to researchers for querying the DBLP bibliography in a graph format, students are required to build an application through a Python shell interface or a Web GUI. Any graph-based queries on Neo4j can be performed using the Cypher query language with the below specified model.

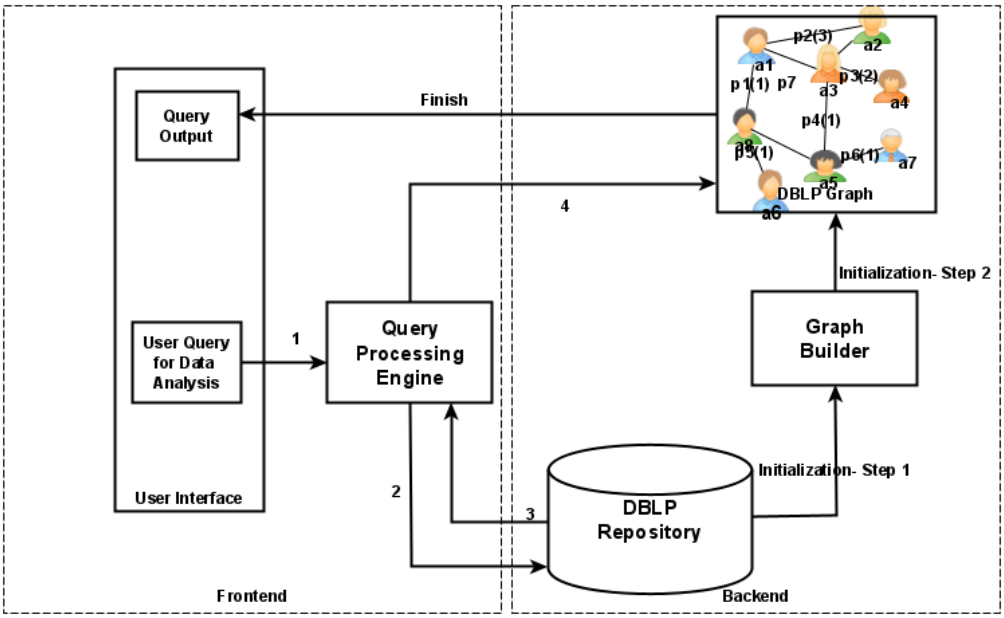

Figure 1 Data Model of DBLP

Functionalities:

Backend:

The data model of DBLP dataset is depicted in Figure 1 using the two building blocks of tool which are Frontend and Backend. Download DBLP dataset version v10 in json format from *https://www.aminer.org/citation.* Backend stores the information that can be directly inferred from the DBLP dataset includes *Authors* and *Publications*. Since the Neo4j model supports entity hierarchies (in contrast with the relational model), a publication can either be an *Article* or an *InProc* or both. As part of Initial Step-1, create CO_AUTHOR relationship between authors that have collaborated, even if they have collaborated on multiple articles. In initial Step-2, build co-authorship graph.

Frontend:

In this block, user enters the keyword related to their search. That keyword based Query will be processed in the DBLP model Backend. The relevant search results will be displayed on the user interface. The queries that this implementation should support are given below as follows:

Q1: ***Keyword Discovery:*** User enters any one research topic displayed on the user interface. Tool returns a list of authors working on that topic (see Table 1). The ***relevance*** estimates the prolificacy of the author within the whole DBLP community that has been working on that topic, while the ***score*** estimates the weight of that keyword among all the author's publication records. This query is useful to perform expert finding on a given research topic and similar research fields.

Table 1 Keyword Discovery

Author	**Relevance**	**Score**
Murray E. Jennex	0.63	46.15
Stefan Smolnik	0.42	15.84
David T. Croasdell	0.39	41.67
Petter Gottschalk	0.39	17.86
Henry Linger	0.34	38.24

Q2: ***Researcher Profiling***: User enters name of a researcher displayed on the user interface. Tool extracts all the topics on which she/he has been working along her/his career (Table 2).

This query is useful to profile researchers, and to discover other researchers working on similar or related topics. To this end, a list of keyword similarities is returned for each topic with the similarity value.

Table 2 Researcher Profiling

Topic	**Author**	**Relevance**	**Score**	Other researchers
Database	Subbarao K	0.64	26.85	Ismael Caballero (0.41 20.51)

Software Quality	Eva Onaindia Dana S.	0.41	52.11	Mario Piattini (0.28 48.28)
Automata	Dana S. Nau	0.37	27.42	Angelica Caro (0.25 19.7)

Q3: ***Influencing Author***: User enters research topic displayed on the user interface. Tool extracts the more influential list of authors working on that topic (see Table 3).

The page rank algorithm computes a score that indicates the transitive influence of an author. The higher the score, authors are the more influential.

Table 3 Influencing Author

Author	**Page Rank**
Edmund Clarke	1.33
Moshe Vardi	1.16
E. Allen Emerson	0.95

Important Note:

- Students to work in groups. A group may consist of 2-3 students only. Larger groups will not be allowed. You can form your own groups. The evaluation will be done for the group and all the members will get the same score. It will be up to the members to have fair share of contribution for the successful implementation/development of the project. Arguments or feedback on individual member's contribution will not be entertained. If someone is not giving valuable contribution, then it depends upon other members to decide how to engage with the non-performing member.
- You are free to design the GUI for the above project as per your creativity.
- Above mentioned basic functionalities are mandatory for mini-project evaluation. Each of the mandatory functionality (Query implementation and display of results) carries 8 marks. Hence, if any group has implemented the three functionalities will be evaluated from 24 marks. An innovative GUI will carry 6 marks, and the backend will carry 10 marks.
- The marking will purely depend upon the evaluators and their evaluation will be final. No arguments in this regard will be entertained and will attract negative marking as per my discretion.
- A working application is expected, if your implementation will not execute then it will not be evaluated and some default mark will be awarded. In case of unethical practices, if observed or brought to my notice, then without any discussion all the groups involved will be awarded 0 marks.
- Do a good work so that you will be able to reflect in your resume.

considered as breach of trust and academic integrity. Every student working on this project means he/she agrees with the policy.

Best Wishes