



Figure 3: DREAM’s GUI-based front-end tier running on a client machine. The GUI features DREAM’s execution of a SPARQL query, Q_9 , derived from the standard LUBM benchmark[10].

planner to generate a near-optimal graph plan, G_P , of any G (see Section 2.2 for details).

3.2.2 Pane 2: Query Planning and Processing

After a participant selects or writes a SPARQL query, Q , she/he can submit Q to DREAM, wherein its query planner will be subsequently activated. Pane 2 illustrates the internal processing performed by the query planner. If the serial execution mode was selected, color-coded graph plans, generated by the query planner, will be displayed in real-time. The lowest-cost (or near-optimal) graph plan, G_P , will be then chosen using a novel cost model. Afterwards, G_P ’s constituent sub-graph(s) will be placed at one or many slave machine(s) (i.e., run as either centralized or distributed), depending on the complexity of G_P . If DREAM is run as distributed, the participant will be able to observe and validate the communication pattern(s) between them, which should at least respect the directionalities of edges in G_P . If the batch execution mode was selected, the mechanics of the specified job scheduler will be demonstrated, whereby the participant can view the query list, with queries getting enqueued and dequeued in real-time based on the scheduler’s policy (e.g., greedy).

3.2.3 Pane 3: Output

This pane displays the final result set(s), coupled with a runtime breakdown. The runtime breakdown encompasses the time spent by DREAM on each of its major tasks: query planning, execution, and communication. This will enable the audience to thoughtfully assess the performance and network results of DREAM.

3.3 Comparisons with Related Schemes

Finally, the audience will be able to execute multiple state-of-the-art centralized and distributed RDF systems, namely RDF-3X [12], Huang *et al.* [11], and H2RDF+ [13]. All these systems will be deployed on the same 11-VM cluster of DREAM (for centralized RDF-3X, only one machine will be utilized), allowing participants

to compare and contrast all the schemes using the same SPARQL queries and RDF datasets.

Acknowledgements

This publication was made possible by NPRP grant # 7-1330-2-483 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

4. REFERENCES

- [1] Dream project - distributed rdf query engine. <http://www.qatar.cmu.edu/~mhhammadou/DREAM/index.html>.
- [2] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach. Scalable semantic web data management using vertical partitioning. In *VLDB*, 2007.
- [3] R. Angles and C. Gutierrez. Querying rdf data from a graph database perspective. In *The Semantic Web: Research and Applications*, 2005.
- [4] M. Atre, J. Srinivasan, and J. A. Hendler. Bitmat: A main-memory bit matrix of rdf triples for conjunctive triple pattern queries. In *ISWC (Posters & Demos)*, 2008.
- [5] V. Bonstrom, A. Hinze, and H. Scheppe. Storing rdf as a graph. In *First Latin American of Web Congress*, 2003.
- [6] M. A. Bornea, J. Dolby, A. Kementsietsidis, K. Srinivas, P. Dantressangle, O. Udrea, and B. Bhattacharjee. Building an efficient rdf store over a relational database. In *SIGMOD*, 2013.
- [7] M. Hammoud, D. A. Rabbou, R. Nouri, S. Beheshti, and S. Sakr. DREAM: Distributed RDF Engine with Adaptive Query Planner and Minimal Communication. *PVLDB*, 8(6), 2015.
- [8] J. Hayes and C. Gutierrez. Bipartite graphs as intermediate model for rdf. In *ISWC*, 2004.
- [9] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. De Melo, and G. Weikum. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *WWW*, 2011.
- [10] <http://swat.cse.lehigh.edu/projects/lubm/>. The LUBM Benchmark.
- [11] J. Huang, D. J. Abadi, and K. Ren. Scalable sparql querying of large rdf graphs. *PVLDB*, 4(11), 2011.
- [12] T. Neumann and G. Weikum. The rdf-3x engine for scalable management of rdf data. *The VLDB Journal*, 19(1), 2010.
- [13] N. Papailiou, D. Tsoumakos, I. Konstantinou, P. Karras, and N. Koziris. H₂RDF+: an efficient data management system for big RDF graphs. In *SIGMOD*, 2014.
- [14] K. Zeng, J. Yang, H. Wang, B. Shao, and Z. Wang. A distributed graph engine for web scale RDF data. *PVLDB*, 6(4), 2013.