ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

Distributed Socialite: A Datalog-Based Language for Large-Scale Graph Analysis

C.Dhivena[1], C. Sumithradevi [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

2. Asst.Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

Abstract : With the rise of social networks, large-scale graph analysis becomes increasingly important. Because SQL lacks the expressiveness and performance needed for graph algorithms, lower-level, general-purpose languages are often used instead. For greater ease of use and efficiency, we propose SociaLite, a high-level graph query language based on Datalog. As a logic programming language, Datalog allows many graph algorithms to be expressed succinctly. However, its performance has not been competitive when compared to low-level languages. With SociaLite, users can provide high-level hints on the data layout and evaluation order; they can also define recursive aggregate functions which, as long as they are meet operations, can be evaluated incrementally and efficiently. Moreover, recursive aggregate functions make it possible to implement more graph algorithms that cannot be implemented in Datalog. We evaluated SociaLite by running nine graph algorithms in total; eight for social network analysis (shortest paths, PageRank, hubs and authorities, mutual neighbors, connected components, triangles, clustering coefficients, and betweenness centrality) and one for biological network analysis (Eulerian cycles). We use two real-life social graphs, LiveJournal and Last.fm, for the evaluation as well as one synthetic graph. The optimizations proposed in this paper speed up almost all the algorithms by 3 to 22 times. SociaLite even outperforms typical Java implementations by an average of 50 percent for the graph algorithms tested. When compared to highly optimized Java implementations, SociaLite programs are an order of magnitude more succinct and easier to write. Its performance is competitive, with only 16 percent overhead for the largest benchmark, and 25 percent overhead for the worst case benchmark. Most importantly, being a query language, SociaLite enables many more users who are not proficient in software engineering to perform network analysis easily and efficiently.

I. INTRODUCTION

We have witnessed the rise of a large number of online social networks, many of which have attracted hundreds of millions of users. Embedded in these databases of social networks is a wealth of information, useful for a wide range of applications. Social network analysis encompasses topics such as ranking the nodes of a graph, community detection, link prediction, as well as computation of general graph metrics. These analyses are often built on top of fundamental graph algorithms such as computing shortest paths and finding connected components. In a recent NSF-sponsored workshop on Social Networks and Mobility in the Cloud, many researchers expressed the need for a better computational model or query language to eventually achieve the goal of letting consumers express queries on their personal social graphs. Datalog is an excellent candidate for achieving this vision because of its high-level declarative semantics and support for recursion. The high-level semantics makes possible many optimizations including parallelization and time bounded approximations. However, the relational representation in Datalog is not a good match for graph analysis. Users are unable to control the data Representation or the evaluation. Consequently, the performance of Datalog is not competitive when compared with other languages. For this reason,

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

developers resort to using general-purpose languages, such as Java, for social network analysis. Not only is it more difficult to write analysis programs in generalpurpose languages, these programs cannot be parallelized or optimized automatically. This paper presents SociaLite, an extension of Datalog that delivers performance similar to that of highly optimized Java programs. Our proposed extensions include data layout declarations, hints of evaluation order, and recursive aggregate functions.

II.RELATED WORK

Extending the semantics of Datalog. Various attempts have been made in the past to allow incremental analysis of aggregate functions in Datalog [29], [30]. Ganguly et al. showed how a non-recursive minimum or maximum function can be rewritten with a set of recursive rules involving negation, and proved that incremental analysis will yield the same result [29]. In Section 3 we described in detail the two well-known Datalog extensions: Datalog for greedy algorithms and Datalog with monotonic aggregate functions. In short, recursive aggregate functions in SociaLite extends Datalog with more expressive power than the two extensions. Recently, Mazuran et al. proposed DatalogFS that extends Datalog with frequency support goals, which allow counting the distinct occurrences satisfying given goals in rules [31]. The counting operation in DatalogFS is a meet operation, hence it can be implemented in recursive aggregate functions in SociaLite. The semantics of the function predicates in SociaLite is similar to the previously proposed external predicates [32]. Other Datalog research. Recently Datalog research has been revived in many domains including security [33], programming analysis [34], and network/distributed systems [35], [36]. Datalog is used in the domain of network and distributed systems to implement, for example, network protocols like distributed consensus. Datalog engines for those domains are extended with features for network programming. Dedalus, for example, has incorporated the notion of time as a language primitive, which helps reasoning with distributed states [37]. In contrast, SociaLite has different goals. It aims to make graph analysis easy and efficient. The extensions of SociaLite are designed and implemented to help programmers write efficient analysis programs easily. Data layout. Various projects in the past have explored nested data structures. NESL is a data-parallel programming language with nested data structures [38]. More recently, nested structures have been adopted in Pig Latin, a high-level language that allows users to supply an imperative program that is similar to a SQL query execution plan [39]. The language then translates the plan into map-reduce operations. In contrast, nested tables in SociaLite are strictly layout hints. The SociaLite rules are oblivious to the nesting in the representation. Users can treat elements in a nested table just like data in any other columns. Graph analysis. A number of query languages have been proposed for graph databases, including GraphLog [40] and GOQL [41]. These query languages support functionalities that are useful for graph analysis, such as subgraph matching and node traversal. SociaLite is as expressive as, if not more, than these query languages, with its recursive aggregate functions and user-defined functions. In terms of distributed frameworks for graph analysis, the popular MapReduce model does not support graph analysis

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

very well [1], so a number of languages have been proposed to simplify the processing of large-scale graphs in parallel. HaLoop provides programming support to iterate map-reduce operations until they converge [42]. Christo Ananth et al. [22] proposed a system in which FASTRA downloads and data transfers can be carried over a high speed internet network. On enhancement of the algorithm, the new algorithm holds the key for many new frontiers to be explored in case of congestion control. The congestion control algorithm is currently running on Linux platform. The Windows platform is the widely used one. By proper Simulation applications, in Windows we can implement the same congestion control algorithm for Windows platform also. The Torrents application which we are currently using can achieve speeds similar to or better than ―Rapid share (premium user) application.

2.1 EXISTING SYSTEM

In the existing system data log implementations are relatively slow. Due to unnecessary computation of sub-optimal distances, as well as inefficient data structures. Data log not competitive for solving fundamental graph algorithms. More generally, join operations defined over relational databases do not seem to be a good match for graph algorithms.

2.1.1 Disadvantages

Lacks of the expressiveness and performance. Double Counting Problem. Not a good match for the datalog.

2.2 PROPOSED SYSTEM

In the proposed extensions include data layout declarations, hints of evaluation order, and recursive aggregate functions. SociaLite, a high-level graph query language based on Data log. SociaLite, users can provide high-level hints on the data layout and evaluation order. These query languages support functionalities that are useful for graph analysis, such as sub graph matching and node traversal. SociaLite is as expressive as, if not more, than these query languages,with its recursive aggregate functions and user-defined functions.

2.2.1 Advantages

Easily Finding the Shortest path to quickly eliminate unnecessary multiple parts It avoids redundant computation Improves data locality Interleaving rule evaluation.

III SYSTEM DESIGN

With the rise of social networks, large-scale graph analysis becomes increasingly important. Because SQL lacks the expressiveness and performance needed for graph algorithms, lower-level, general-purpose languages are often used instead. For greater ease of use and efficiency, we propose SociaLite, a high-level graph query language based on Datalog. As a logic programming language, Datalog allows many graph algorithms to be expressed succinctly. However, its performance has not been competitive when compared to low-level languages. With SociaLite, users can provide high-level hints on the data layout and evaluation order; they can also define recursive aggregate functions which, as long as they are meet operations, can be evaluated

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

incrementally and efficiently. Moreover, recursive aggregate functions make it possible to implement more graph algorithms that cannot be implemented in Datalog.

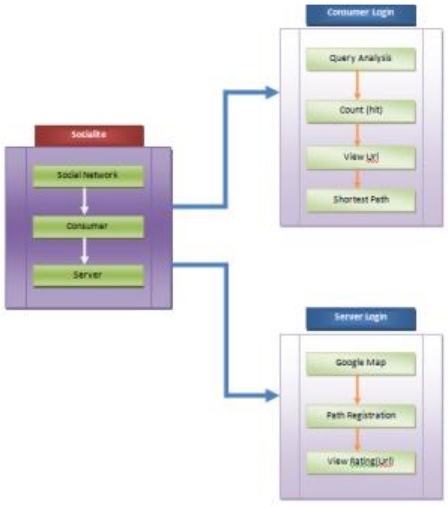


Fig 1. Architecture Diagram

3.1 Socialite

SociaLite, a high-level graph query language based on Datalog,SociaLite supports recursive aggregate functions, which greatly improve the language's expressiveness.A socialite is a person (usually from a privileged, wealthy or aristocratic background), who has a largely known reputation and a high social position in upper class society. A socialite spends a significant amount of time participating in popular social activities and continually attends various social gatherings such as charity events, private parties, fashion shows, fine arts fundraisers, luncheons, festivals and other exclusive events; as well as engaging in conversations with guests or entertaining members who are socially connected or share a similar upper-class social standing.

3.2 Social Networks

Social networks, large-scale graph analysis becomes increasingly important, Its lower-level, general-purpose languages, In the proposed system SociaLite, a high-level graph query language based on Data log In the Recursive aggregate functions make it possible to implement more graph algorithms that cannot be implemented in Data log.

3.3 Consumer

It buys products or services for personal us e and not for manufacture or resale. A consumer is someone who can make the decision whether or not to purchase an item at the store, and someone who can be influenced by marketing and advertisements. Any time someone goes to a store and purchases a toy, shirt, beverage, or anything else, they are making that decision as a consumer.

3.4 Server

Server is the process of managing a business or non-profit organization, so that it remains stable and continues to grow. The administration of a business includes the performance or management of business operations and decision making, as well as the efficient organization of people and other resources, to direct activities toward common goals and objectives. In general, administration refers to the broader management function, including the associated finance, personnel and MIS services. In some analyses, management is viewed as a subset of administration, specifically associated with the technical and operational aspects of an organization, distinct from executive or strategic

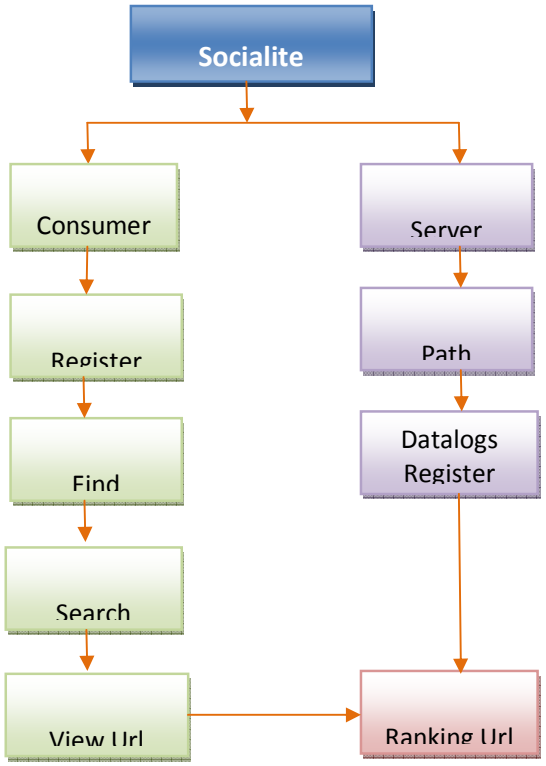
ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

functions. Alternatively, administration can refer to the bureaucratic or operational performance of routine office tasks, usually internally oriented and reactive rather than proactive. Administrators, broadly speaking, engage in a common set of functions to meet the organization's goals. These "functions" of the administrator were described by Henri Fayol Sometimes creating output, which includes all of the processes that create the product that the business sells, is added as a sixth element.

IV. DATA FLOW DIAGRAM

Social networks, large-scale graph analysis becomes increasingly important, Its lower-level, general-purpose languages, In the proposed system SociaLite, a high-level graph query language based on Data log In the Recursive aggregate functions make it possible to implement more graph algorithms that cannot be implemented in Data log.



V. MODULES DESCRIPTION

- Social Network Analysis
- Data logs
- Query Languages
- Evaluation Of Socialite

4.1 Social Network Analysis

In the Social Network Analysis, it an Social network analysis (SNA) is a strategy for investigating social structures. Examples of social structures commonly visualized through social network analysis include social media networks, friendship acquaintance networks, kinship, disease transmission, and sexual relationships. These networks are often visualized through sociograms in which nodes are represented as points and ties are represented as lines. Social network analysis has emerged as a key technique in modern sociology. It has also gained a significant following in anthropology, biology, communication studies, economics, geography, history, information science, organizational studies, political science, social psychology, development studies, and sociolinguistics and is now commonly available as a consumer tool.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

4.2 Data Logs

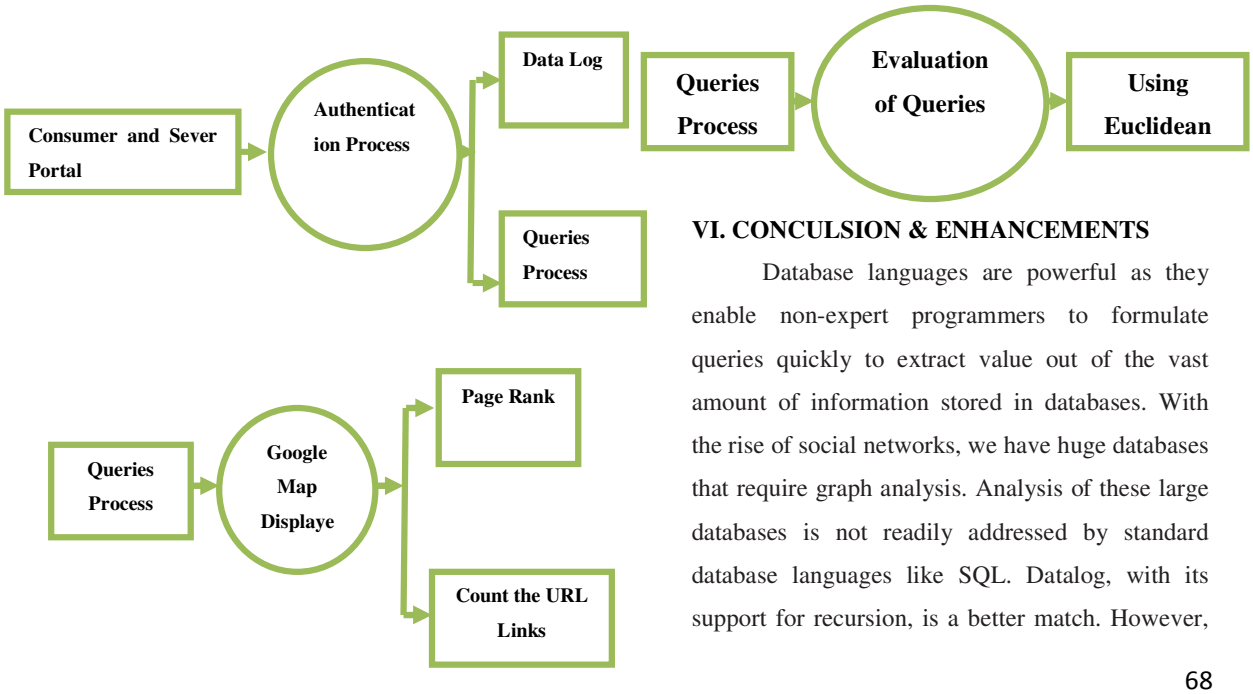
Data log is a truly declarative logic programming language that syntactically is a subset of Prolog. It is often used as a query language for deductive databases. In recent years, Data log has found new application in data integration, information extraction, networking, program, analysis, security, and cloud computing. Data log is an excellent candidate for achieving this vision because of its high-level declarative semantics and support for recursion. Data log is not a good match for graph analysis. Users are unable to control the data representation or the evaluation.

4.3 Query Languages

A number of query languages have been proposed for graph databases, including Graph Log. These query languages support functionalities that are useful for graph analysis, such as sub graph matching and node traversal. It is a proprietary object-oriented query language for querying relational databases; successor of Datalog.

4.4 Evaluation of Socialite

A socialite is a person (usually from a privileged, wealthy or aristocratic background), who has a largely known reputation and a high social position in upper class society. A socialite spends a significant amount of time participating in popular social activities and continually attends various social gatherings such as charity events, private parties, fashion shows, fine arts fundraisers, luncheons, festivals and other exclusive events; as well as engaging in conversations with guests or entertaining members who are socially connected or share a similar upper-class social standing. All the optimizations presented in the system have been implemented in a SociaLite compiler. In the large collection of popular graph algorithms can be expressed succinctly in SociaLite, including Page Rank, hubs and authorities, clustering coefficients, as well as Eulerian cycle, an important algorithm used in DNA sequence assembly.



VI. CONCULSION & ENHANCEMENTS

Database languages are powerful as they enable non-expert programmers to formulate queries quickly to extract value out of the vast amount of information stored in databases. With the rise of social networks, we have huge databases that require graph analysis. Analysis of these large databases is not readily addressed by standard database languages like SQL. Datalog, with its support for recursion, is a better match. However,

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

current implementations of Datalog are significantly slower than programs written in conventional languages. Our proposed language, SociaLite, is based on Datalog and thus can succinctly express a variety of graph algorithms in just a few lines of code. SociaLite supports recursive aggregate functions, which greatly improve the language's expressiveness. More importantly, the convenience of our high-level query language comes with a relatively small overhead. Semi-naive evaluation and prioritized computation can be applied to recursive aggregate functions that are meet operations. The integration with imperative languages enables wider class of applications to be implemented in SociaLite.

REFERENCES

[1] C. Yu, "**Beyond simple parallelism: Challenges for scalable complex analysis over social data,**" in Proc. NSF Workshop Soc. Netw. Mobility Cloud, 2012, pp. 47–48.

[2] C. Beeri, S. Naqvi, R. Ramakrishnan, O. Shmueli, and S. Tsur, "**Sets and negation in a logic database language (LDL1),**" in Proc. ACM Symp. Principles Database Syst., 1987, pp. 21–37.

[3] I. S. Mumick, H. Pirahesh, and R. Ramakrishnan, "**The magic of duplicates and aggregates,**" in Proc. 16th Int. Conf. Very Large Data Bases, 1990, pp. 264–277.

[4] Logicblox inc [Online]. Available: http://www.logicblox.com/, 2013.

[5] E. W. Dijkstra, "**A note on two problems in connexion with graphs**," Num. Math., vol. 1, no. 1, pp. 269–271, 1959.

[6] S. Greco and C. Zaniolo, "**Greedy algorithms in datalog**," Theory Practice Logic Programm., vol. 1, pp. 381–407, 2001.

[7] K. A. Ross and Y. Sagiv, "**Monotonic aggregation in deductive databases**," J. Comput. Syst. Sci., vol. 54, no. 1, pp. 79–97, 1997.

[8] J. W. Lloyd, Foundations of Logic Programming. New York, NY, USA: Springer, 1987.

[9] A. Tarski, "**A lattice-theoretical fixpoint theorem and its applications**," Pacific J. Math., vol. 5, no. 2, pp. 285–309, 1955.

[10] S. Greco, C. Zaniolo, and S. Ganguly, "**Greedy by choice**," in Proc. ACM Symp. Principles Database Syst., 1992, pp. 105–113.

[11] F. Giannotti, D. Pedreschi, D. Sacca, and C. Zaniolo, "Non-determinism in deductive databases," in Proc. 3rd Int. Conf. Deductive Object-Oriented Databases, 1991, pp. 129–146.

[12] R. Krishnamurthy and S. A. Naqvi, "Non-deterministic choice in datalog," in Proc. 2nd Int. Conf. Data Knowl. Bases, 1988, pp. 416– 424.

[13] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2010, pp. 135–146.

[14] L. C. Freeman, "A set of measures of centrality based on betweenness," Sociometry, vol. 40, no. 1, pp. 35–41, 1977.

[15] T. Condie, D. Chu, J. M. Hellerstein, and P. Maniatis, "Evita raced: Metacompilation for declarative networks," Proc. VLDB Endowment, vol. 1, pp. 1153–1165, 2008.

[16] Iris, an open-source datalog engine [Online]. Available: http:// www.iris-reasoner.org/, 2013.

[17] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," Nature, vol. 435, pp. 814–818, 2005.

[18] J. W. Raymond, E. J. Gardiner, and P. Willett, "Rascal: Calculation of graph similarity using maximum common edge subgraphs," Comput. J., vol. 45, pp. 631–644, 2002.

[19] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in Proc. Int. Conf. Companion World Wide Web, 1998, pp. 107–117.

[20] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," J. ACM, vol. 46, pp. 604–632, 1999.

[21] Livejournal [Online]. Available: http://www.livejournal.com/, 2013.

[22] Christo Ananth, A. Ramalakshmi, S. Velammal,B. Rajalakshmi Chmizh, M. Esakki Deepana, "FASTRA –SAFE AND SECURE", International Journal For Technological Research

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Special Issue 11, March 2016

In Engineering (IJTRE), Volume 1, Issue 12, August-2014,pp: 1433-1438.

[23] U. Brandes, "A faster algorithm for betweenness centrality," J. Math. Soc., vol. 25, no. 2, pp. 163–177, 2001.

[24] J. D. L. Rivas and C. Fontanillo, "Protein-protein interactions essentials: Key concepts to building and analyzing interactome networks," PLoS Comput. Biol., vol. 6, no. 6, pp. e1000807, 2010.

[25] N. G. de Bruijn, "A combinatorial problem," Koninklijke Nederlandse Akademie v. Wetenschappen, vol. 49, pp. 758–764, 1946.

[26] R. M. Idury and M. S. Waterman, "A new algorithm for DNA sequence assembly," J. Comput. Biol., vol. 2, pp. 291–306, 1995.

[27] C. Hierholzer and C. Wiener, "Ueber die mglichkeit, einen linienzug ohne wiederholung und ohne unterbrechung zu umfahren," Math. Annalen, vol. 6, pp. 30–32, 1873.

[28] P. G. Kolaitis and C. H. Papadimitriou, "Why not negation by fixpoint?" J. Comput. Syst. Sci., vol. 43, pp. 125–144, 1991.

[29] S. Ganguly, S. Greco, and C. Zaniolo, "Minimum and maximum predicates in logic programming," in Proc. ACM Symp. Principles Database Syst., 1991, pp. 154–163.

[30] S. Sudarshan and R. Ramakrishnan, "Aggregation and relevance in deductive databases," in Proc. 17th Int. Conf. Very Large Data Bases, 1991, pp. 501–511.

[31] M. Mazuran, E. Serra, and C. Zaniolo, "Extending the power of datalog recursion," VLDB J., vol. 22, no. 4, pp. 471–493, 2013.

[32] R. G. Chimenti, R. Gamboa, and R. Krishnamurthy, "Towards on open architecture for LDL," in Proc. 15th Int. Conf. Very Large Data Bases, 1989, pp. 195–203.

[33] M. Sherr, A. Mao, W. R. Marczak, W. Zhou, B. T. Loo, and M. Blaze, "A3: An extensible platform for application-aware anonymity," in Proc. Netw. Distrib. Syst. Security Symp., 2010, pp. 247–266.

[34] J. Whaley and M. S. Lam, "Cloning-based context-sensitive pointer alias analyses using binary decision diagrams," in Proc. ACM SIGPLAN Conf. Programm. Language Design Implementation, 2004, pp. 131–144.

[35] P. Alvaro, T. Condie, N. Conway, K. Elmeleegy, J. M. Hellerstein, and R. C. Sears, "Boom analytics: Exploring data-centric, declarative programming for the cloud," in Proc. 5th Eur. Conf. Comput. Syst., 2010, pp. 223–236.

[36] B. T. Loo, T. Condie, M. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica, "Declarative networking," Commun. ACM, vol. 52, no. 11, pp. 87–95, 2009.

[37] P. Alvaro, W. R. Marczak, N. Conway, J. M. Hellerstein, D. Maier, and R. Sears, "Dedalus: Datalog in time and space," in Proc. 1st Int. Conf. Datalog Reloaded, 2010, pp. 262–281.

[38] G. E. Blelloch, "Programming parallel algorithms," Commun. ACM, vol. 39, pp. 85–97, 1996.

[39] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig latin: A not-so-foreign language for data processing," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2008, pp. 1099–1110.

[40] M. P. Consens and A. O. Mendelzon, "Expressing structural hypertext queries in graphlog," in Proc. 2nd Annu. ACM Conf. Hypertext, 1989, pp. 269–292.

[41] L. Sheng, Z. M. €Ozsoyoglu, and G. € Ozsoyoglu, "A graph query language and its query processing," in Proc. Int. Conf. Data Eng., 1999, pp. 572–581.

[42] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "HaLoop: Efficient iterative data processing on large clusters," Proc. VLDB Endowment, vol. 3, nos. 1/2, pp. 285–296, 2010.