Streaming in a Connected World: Querying and Tracking Distributed Data Streams

Graham Cormode
AT&T Labs—Research
180 Park Avenue
Florham Park NJ
graham@research.att.com

Minos Garofalakis
Yahoo! Research and UC Berkeley
2821 Mission College Blvd
Santa Clara CA
minos@yahoo-inc.com

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications

General Terms

Algorithms, Measurement, Reliability

Keywords

Data Streams, Sensor Networks, Distributed Data

1. INTRODUCTION

Today, a majority of data is fundamentally distributed in nature. Data for almost any task is collected over a broad area, and streams in at a much greater rate than ever before. In particular, advances in sensor technology and miniaturization have led to the concept of the *sensor network*: a (typically wireless) collection of sensing devices collecting detailed data about their surroundings. A fundamental question arises: how to query and monitor this rich new source of data? A similar scenario emerges within more traditional, wired networks: if data is collected over remote sites, either about observed external conditions or about the network itself (e.g. in IP network monitoring), how to process this data in order to answer certain queries? Additionally, other emerging models of distributed computation, such as peer-to-peer (P2P) networks and grid-based computing face the same problems of managing and interrogating data streams observed by distributed participants.

The prevailing paradigm in database systems has been understanding management of *centralized* data: how to organize, index, access, and query data that is held centrally on a single machine or a small number of closely linked machines. In these distributed scenarios, the axiom is overturned: now, data typically streams into remote sites at high rates. Here, *it is not feasible* to collect the data in one place: the volume of data collection is too high, and the capacity for data communication relatively low. For example, in battery-powered wireless sensor networks, the main drain on battery life is communication, which is orders of magnitude more expensive than computation or sensing. This establishes a fundamental concept for distributed stream monitoring: if we can perform more computational work *within* the network to reduce the communication needed, then we can significantly improve the value of our network, by increasing its useful life and extending the range of computation possible over the network.

We consider two broad classes of approaches to such in-network query processing, by analogy to types of queries in traditional DBMSs. In the *one shot* model, a query is issued by a user at some site, and must be answered based on the current state of data in the network. We identify several possible approaches to this problem. For simple queries, partial computation of the result over a spanning tree can reduce the data transferred significantly. For 'holistic' queries, such as medians, count distinct and so on, simple combination of partial results is insufficient, and instead clever composable summaries give a compact way to accurately approximate query answers. Lastly, careful modeling of correlations between measurements and other trends in the data can further reduce the number of sensors probed.

In the *continuous* model, a query is placed by a user which requires the answer to be available continuously. This yields yet further challenges, since even using tree computation, summarization and modeling, we cannot afford to communicate every time new data is received by one of the remote sites. Instead, the result of work on this problem has been a new tradeoff of reduced accuracy in the query answer for reduced communication cost. This has led to a variety of techniques for different query types to apportion the available "uncertainty" in the query answer between different sites, and to model the evolution of measured values to anticipate future values and so reduce communication further.

There are many open problems relating to distributed data stream monitoring still to be studied, and systems addressing these issues are just starting to appear. There is great potential for work in this area to have impact on the next generation of data management systems in a world that is inherently distributed and constantly changing. Our objective in this tutorial is to discuss the algorithmic foundations of this new world, illustrate some of the powerful techniques that have been developed to address these challenges, and outline interesting directions for future work in the area. This tutorial complements and builds on some of the first attempts at collecting work in this area [18] and tutorials which have emphasized the hardware and systems aspects [29]. The aim is to inspire attendees to contribute to this exciting, growing area of research.

Copyright is held by the author/owner(s).

SIGMOD'07, June 12–14, 2007, Beijing, China.
ACM 978-1-59593-686-8/07/0006.

2. TUTORIAL OUTLINE

2.1 Introduction and Motivation

Large-scale event-monitoring systems require fast or continuous query answering in a world where the data is streaming and inherently distributed. The key challenge is to minimize both communication and processing burden while ensuring accuracy and timeliness of answers. We discuss example application domains, including sensor networks, network monitoring, and P2P networks. We also cover basic (centralized) data-streaming models and results, and outline the key dimensions of distributed data-streaming problems:
(1) Querying Model: One-shot vs. continuous, exact vs. approximate, deterministic vs. randomized;
(2) Communication Model: Single-level, hierarchical, or fully-distributed (e.g., DHT-based P2P systems), other communication constraints (e.g., network loss, intermittent connectivity); and,
(3) Class of Queries: Holistic vs. non-holistic aggregates, duplicate sensitive vs. insensitive aggregates, more complex queries (e.g., inference models, set-valued results).

2.2 One-Shot Distributed-Stream Querying

The one shot approach requires a query to be distributed to relevant sites in the network, and the answer computed and relayed back to the user. We discuss various approaches to aggregate computation:

- **Tree-based Aggregation.** Computing simple, non-holistic aggregates (e.g., MIN, MAX, SUM, AVG) over a hierarchical architecture using 'in-network' aggregation of fixed-size partial aggregates (e.g., TAG [30]). For holistic aggregates, we discuss techniques for avoiding linear communication costs through effective approximation using *composable data synopses* for count distinct, quantiles, heavy hitters, join size, and so on [17, 3, 2, 11, 19, 33].

- **Robustness/Loss Issues.** Robustness is a key concern for hierarchical aggregation schemes, as a single loss near the root can dramatically affect accuracy. We cover possible remedies for both holistic and non-holistic aggregates, including *gossip-based techniques* [26, 25, 24], *duplicate-insensitive aggregation schemes* [7, 34, 21, 12], and *hybrid schemes* combining elements of both approaches [32].

- **Probabilistic Approximation.** Cover recent work on using *probabilistic models* of site values and their correlations to reduce communication overheads for approximate answers. [16, 6].

2.3 Continuous Distributed-Stream Tracking

The continuous, distributed setting puts a much more stringent demand on the system: a continuous query is distributed to the participating sites, and they must collaborate to ensure that the answer to the query is continuously provided to the user that is accurate (e.g., within specified error bounds) compared to the exact current state. We cover various fundamental ideas that are applicable in this setting:

- **Adaptive Slack Allocation.** A first cut is to take the allowable "slack" in answering the query within allowable bounds, and distribute it between different participants for different query types, e.g., top-k (most frequent) items [4], item values [35], set expressions [15], and duplicate resilient aggregates [13]. In such settings, communication is necessary to adjust the slacks, plus some global communication is needed when a large rebalancing of slacks takes place.

- **Predictive Models of Site Behavior.** We discuss recent work that extends the idea of local-slack allocation by incorporating simple *models* of the data evolution to "predict" site behavior. Combined with intelligent summarization techniques, these approaches only require concise communication exchanges when prediction models are no longer accurate [10, 9, 8]

- **Distributed Triggering.** An important common feature of many distributed continuous monitoring problems is evaluating a condition over distributed data and *triggering* when it is met. We will show recent work that has provided several solutions to this problem based on a variety of techniques, both deterministic and randomized (where the probability of triggering increases with the amount by which a threshold is exceeded) [27, 38, 22].

2.4 Distributed Data-Stream Systems

We look at different design choices, algorithms, and assumptions in state-of-the-art systems and research prototypes, including Borealis/Medusa [39, 1], Telegraph/TelegraphCQ and TinyDB/TAG [5, 30, 31]. Other relevant systems include the Gigascope streaming database (actually deployed for monitoring AT&T's ISP network [14]), and the P2 parallel dataflow engine [28].

2.5 Future Research Directions and Open Problems

We discuss several challenging directions for future work in the distributed data-streaming arena, including:

- Extensions to other application areas and more complex communication models, e.g., monitoring P2P services over shared infrastructure (OpenDHT [36] over PlanetLab), and dealing with constrained communication models (e.g., intermittent-connectivity and delay-tolerant networks (DTNs) [23]).

- Richer classes of distributed queries, e.g., set-valued query answers, machine-learning inference models [20].

- Developing a theoretical/algorithmic foundation of distributed data-streaming models: what are fundamental lower bounds, how to apply/extend information theory, communication complexity, and distributed coding?

- Richer prediction models for stream tracking: can models effectively capture site correlations rather than just local site behavior? More generally, understand the model complexity/expressiveness tradeoff, and come up with principled techniques for capturing it in practice (e.g., using the MDL principle [37]).

- Stream computations over an *untrusted* distributed infrastructure: coping with privacy and authentication issues in a communication/computation-efficient manner.

3. INTENDED AUDIENCE

This target audience for this tutorial comprises SIGMOD attendees who are interested in understanding the rapidly growing area of distributed data-stream management and in-network query processing. We will not assume any background in the area, but will attempt to give broad coverage of many of the key ideas, making it appropriate for graduate students seeking new areas to study and researchers active in the field alike.

4. ABOUT THE PRESENTERS

Graham Cormode is a Principal Member of Technical Staff in the Database Management Group at AT&T Shannon Laboratories in New Jersey. Previously, he was a researcher at Bell Labs, after postdoctoral study at the DIMACS center in Rutgers University from 2002-2004. He was granted his PhD by the University of Warwick in 2002. He works on data stream algorithms, large-scale data mining, and applied algorithms, with applications to databases, networks, and fundamentals of communications and computation.

Minos Garofalakis is a Principal Research Scientist at Yahoo! Research, and an Adjunct Associate Professor at the University of California, Berkeley. He obtained his Ph.D. from the University of Wisconsin-Madison in 1998. He joined Yahoo! 2006, after spending 6 years as a Member of Technical Staff with Bell Labs, and 2 years with Intel Research. His current research interests include data streaming, approximate query processing, network-data management, and XML databases. He has presented tutorials on data streaming and approximate query processing at several of the top database forums (including, VLDB'2001, ACM SIGMOD'2002, VLDB'2002, and ACM SIGKDD'2002), and is the co-editor for an upcoming Springer-Verlag volume on Data-Stream Management.

5. REFERENCES

[1] D. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J. Hwang, W. Lindner, A. Rasin, N. Tatbul, Y. Xing, and S. Zdonik. Distributed operation in the borealis stream processing engine. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2005.

[2] N. Alon, P. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. In *Proceedings of ACM Principles of Database Systems*, pages 10–20, 1999.

[3] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 20–29, 1996. Journal version in *Journal of Computer and System Sciences*, 58:137–147, 1999.

[4] B. Babcock and C. Olston. Distributed top-k monitoring. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2003.

[5] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. R. Madden, F. Reiss, and M. A. Shah. TelegraphCQ: continuous dataflow processing. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, page 668, 2003.

[6] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. In *IEEE International Conference on Data Engineering*, 2006.

[7] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *IEEE International Conference on Data Engineering*, 2004.

[8] G. Cormode and M. Garofalakis. Efficient strategies for continuous distributed tracking tasks. *IEEE Data Engineering Bulletin*, 28(1):33–39, March 2005.

[9] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In *Proceedings of the International Conference on Very Large Data Bases*, 2005.

[10] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2005.

[11] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[12] G. Cormode and S. Muthukrishnan. Space efficient mining of multigraph streams. In *Proceedings of ACM Principles of Database Systems*, 2005.

[13] G. Cormode, S. Muthukrishnan, and W. Zhuang. What's different: Distributed, continuous monitoring of duplicate resilient aggregates on data streams. In *IEEE International Conference on Data Engineering*, 2006.

[14] C. Cranor, T. Johnson, O. Spatscheck, and V. Shkapenyuk. Gigascope: A stream database for network applications. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 647–651, 2003.

[15] A. Das, S. Ganguly, M. Garofalakis, and R. Rastogi. Distributed set-expression cardinality estimation. In *Proceedings of the International Conference on Very Large Data Bases*, 2004.

[16] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-drive data acquisition in sensor networks. In *Proceedings of the International Conference on Very Large Data Bases*, 2004.

[17] P. Flajolet and G. N. Martin. Probabilistic counting. In *IEEE Conference on Foundations of Computer Science*, pages 76–82, 1983. Journal version in *Journal of Computer and System Sciences*, 31:182–209, 1985.

[18] M. Garofalakis. Special issue on in-network query processing. *IEEE Data Engineering Bulletin*, 28(1), March 2005.

[19] M. Greenwald and S. Khanna. Power-conserving computation of order-statistics over sensor networks. In *Proceedings of ACM Principles of Database Systems*, pages 275–285, 2004.

[20] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Information Processing in Sensor Networks*, 2004.

[21] M. Hadjieleftheriou, J. W. Byers, and G. Kollios. Robust sketching and aggregation of distributed data streams. Technical Report 2005-11, Boston University Computer Science Department, 2005.

[22] A. Jain, J. Hellerstein, S. Ratnasamy, and D. Wetherall. A wakeup call for internet monitoring systems: The case for distributed triggers. In *Proceedings of the 3rd Workshop on Hot Topics in Networks (Hotnets)*, 2004.

[23] S. Jain, K. Fall, and P. Rabin. Routing in a delay tolerant network. In *ACM SIGCOMM*, 2005.

[24] S. Kashyap, S. Deb, K. V. M. Naidu, R. Rastogi, and A. Srinivasan. Efficient gossip-based aggregate computation. In *Proceedings of ACM Principles of Database Systems*, 2006.

[25] D. Kempe, A. Dobra, and J. Gehrke. Computing aggregates using gossip. In *IEEE Conference on Foundations of Computer Science*, 2003.

[26] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. In *Proceedings of the ACM Symposium on Theory of Computing*, 2001.

[27] R. Keralapura, G. Cormode, and J. Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2006.

[28] T. Loo, J. Hellerstein, I. Stoica, and R. Ramakrishnan. Declarative routing: Extensible routing with declarative queries. In *ACM SIGCOMM*, 2005.

[29] S. Madden. Data management in sensor networks. In *Proceedings of European Workshop on Sensor Networks*, 2006.

[30] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: a Tiny AGgregation service for ad-hoc sensor networks. In *Proceedings of Symposium on Operating System Design and Implementation*, 2002.

[31] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TinyDB: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173, 2005.

[32] A. Manjhi, S. Nath, and P. Gibbons. Tributaries and deltas: Efficient and robust aggregation in sensor network streams. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2005.

[33] A. Manjhi, V. Shkapenyuk, K. Dhamdhere, and C. Olston. Finding (recently) frequent items in distributed data streams. In *IEEE International Conference on Data Engineering*, pages 767–778, 2005.

[34] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. Synopsis diffusion for robust aggrgation in sensor networks. In *ACM SenSys*, 2004.

[35] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2003.

[36] S. Rhea, G. Brighten, B. Karp, J.Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and Y. Harlan. OpenDHT: A public DHT service and its uses. In *ACM SIGCOMM*, 2005.

[37] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[38] I. Sharfman, A. Schuster, and D. Keren. A geometric approach to monitoring threshold functions over distribtuted data streams. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2006.

[39] S. Zdonik, M. Stonebraker, M. Cherniack, and U. Cetintemel. The aurora and medusa projects. *Bulletin of the Technical Committee on Data Engineering*, pages 3–10, Mar. 2003.