

Daedalus: Statistical Aggregation for Large-Scale Dynamic Networks

Evangelia Kalyvianaki and Yu-En Lu

Computer Laboratory

University of Cambridge

15 JJ Thomson Avenue, Cambridge CB3 0FD, UK

Email: {firstname.lastname}@cl.cam.ac.uk

I. INTRODUCTION

In the last few years, there has been a dramatic growth of global, distributed applications such as Skype for VoIP telephony, P2P file sharing systems, and software/content distribution networks. Almost in parallel, a number of different monitoring overlay networks have been proposed to monitor the health of such systems for facilitating tasks like performance planning and problem solving.

An important task of any monitoring overlay, is the computation of aggregation functions such as MEAN over a set of nodes. However, in the face of multi-million nodes networks, the computation of any aggregation function over the whole network or even a large subset of it is challenging. Every query scheme has to be *robust* against churn – nodes join and leave the system in arbitrary rates – and *scalable* up to millions of nodes.

Consider the simple approach to aggregation. We may first do a broadcast to the network with an aggregate query and then have each node return its local value. Clearly, this would take significant time to complete in a large network. Moreover, without suitable coordination, the responses may collectively become a DDoS attack to the querying node.

The lack of scalability in this approach has led to in-network computation, where an overlay is constructed to disseminate and compute the query in a distributed manner [2]. Several variations to computing aggregates have been proposed for sensor networks, Grids and cluster-based applications, with each approach being focused on the constraints imposed by the network under study. Considine et al. [3] for example propose the use of duplicate insensitive sketches to compute SUM aggregates for networks with resource constraints and node failures such as sensor overlays. Other systems like Astrolabe [5] target smaller and less dynamic networks by constructing a fault resilient monitoring network to cope with network partitioning. In this paper, we address the problem of computing aggregate functions for large-scale networks with high churn.

We propose **Daedalus**, a monitoring overlay designed for dynamic networks of multi-million of nodes. As the network in consideration is so large that any aggregated result may be obsolete by the time it has been evaluated, we argue that having statistical confidence semantics will enable a *timely* and

confident monitoring infrastructure.

Daedalus, is based on overlay sampling and database aggregation techniques. For achieving scalability we compute any aggregate function over a representative subset of nodes. With the overlay sampling, Daedalus provides statistically confident estimations of the aggregation continuously over the query evaluation. For example, consider the query: *What is the avg session duration?* Whilst the complete aggregation result may be calculated, Daedalus samples the network and gives an estimation: 10 mins with 1 min variation and 95% confidence. The confidence is the probability of the estimation falling within the given interval.

In the next section we give new query semantics that incorporate both temporal properties of the overlay and statistical confidence level. Via the two techniques, our system delivers timely monitoring information with confidence.

II. SYSTEM OVERVIEW

Daedalus borrows from statistical analysis and DBMS for building a scalable monitoring infrastructure; its architecture is described below.

A. Architecture and Scalability

The goal of Daedalus is two fold, achieving scalability and efficient performance for near real-time monitoring. An overlay network is formed to monitor and disseminate the query and information produced by the client application. Figure 1 gives an overview of Daedalus' architecture.

The Daedalus monitoring overlay network is composed of the most stable nodes and is used to disseminate the query and compute the answer, as in step 1. When connected to the overlay network a regular node initialises a list of predefined variables to keep statistics upon. Every some timeslots t , also initialised upon connection, each node sends the computed values to its connecting Daedalus node. Daedalus nodes gather information from all regular nodes and compute summaries over them. When the Querying Node submits a query a broadcast message is sent to all Daedalus nodes. The final aggregation function is computed using the Daedalus nodes summaries. We further discuss the aggregation computation scheme in the next section.

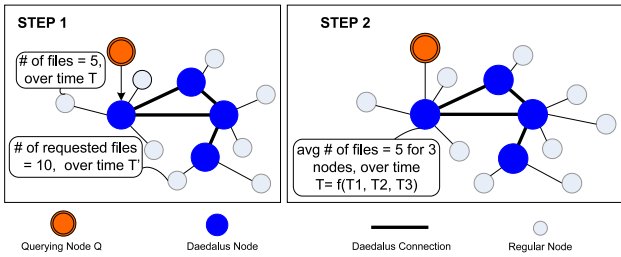


Fig. 1. Daedalus uses a subset of the monitored system’s nodes to compute global aggregate functions. A step by step procedure is used to select the nodes that contribute towards a query result. In step (1), Daedalus network is created by the most stable nodes of the network. The final aggregation function is computed over the selected nodes, step (2).

Scalability is achieved via forming the overlay with *supernodes*, an idea proliferated by Gnutella and Skype. Gnutella selects nodes with characteristics such as long session duration and sufficient resource capabilities as supernodes, to form a fast and scalable search subnetwork. As supernodes are stable and have faster connectivity, we expect low maintenance cost and thus reducing the overall network traffic produced.

Whilst scalability is our main goal, the correctness of the result is also a prerequisite to our design. We discuss the accuracy of our aggregation scheme in section II-B.

B. Online Aggregation

In Daedalus, we are interested in the temporal behaviour of network measurement values as well as their distribution over the entire network. Here, we propose the online aggregation method for large scale aggregation queries in such context. We consider the following cases of aggregation¹: MAX, MIN, AVG, MEAN, DIST.

As the scale of aggregation may be large, evaluating the final result over all Daedalus nodes can be time consuming and impractical. Therefore, we propose to do online aggregation [4] in which we can have an approximate result *during* the full aggregation evaluation in addition to the final result.

Online aggregation adopts the notion of confidence interval in statistics to provide probabilistic semantics during the aggregation process. As random sampling is the core to all statistical treatments in aggregation, we briefly discuss our approach for achieving large scale random sampling.

Random sampling has been available in indexing data structures [4], however, it is not clear how this can be applied in P2P networks. Notice that random sampling in distributed context is not easy since it has to be insensitive to network conditions such as latency and load and scalable for large networks. For example, an overlay network may be a random graph; a naive sampling along the graph paths would be biased towards the querying node’s neighbours and nodes with low latency. Therefore, a global randomisation and query processing technique are required to ascertain the randomness and completeness of sampling.

¹An Aggregation function is a function f such that $f(x_1, x_2, \dots, x_n) = f(f(x_1, x_2, \dots, x_t), \dots, f(x_{n-t+1}, \dots, x_n))$.

We evaluate the online aggregation over the Daedalus overlay network as follows. Emphasis is given on randomly selecting nodes to compute online aggregations in a distributed manner. The key idea is to create a broadcast tree with edges of randomly assigned numbers. Each node can thus have a random order by concatenating the numbers along the edges from root to them. Firstly, the querying node, for example the operation centre, issues the query to some Daedalus nodes which are given a random number. These nodes propagate the query to their neighbours, assigning new random numbers to them, and so on. Upon having no neighbours to forward onwards or receiving identical queries, each node acknowledges the query result to their parent nodes. Acknowledges are only forwarded upwards along the tree when all messages with smaller random numbers have been acknowledged and sent.

Thus, every node obtains a global random order without user posing constraints on the size of network and network latency. Also, this approach assumes no topology and therefore can be deployed in both structured and unstructured P2P networks.

C. Evaluation Plan

As any other application designed for large-scale networks, we are facing the challenges of testing Daedalus against a real, global application. Therefore, we initially plan to evaluate the sampling algorithm and the aggregation processing on top of a DHT-based file sharing system and a Gnutella-like network using the Emulab network testbed [1], with real Gnutella workload traces and other related published measurements. We can thereafter test our approach against globally deployed applications.

III. SUMMARY

In this paper we propose Daedalus, a monitoring overlay network designed to compute approximate aggregate functions over multi-million of nodes, dynamic networks. Our design combines techniques from DBs and P2P systems. A small subset of related work is briefly described throughout the paper². Daedalus further extends these ideas to networks of much larger sizes, which we believe is a unique challenge.

REFERENCES

- [1] <http://www.emulab.net>.
- [2] M. Bawa, A. Giones, H. Garcia-Molina, and R. Motwani. The price of validity in dynamic networks. In *Proceedings of the ACM SIGMOD*, 2004.
- [3] Jeffrey Considine, Feifei Li, George Kollios, and John Byers. Approximate aggregation techniques for sensor databases. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*, 2004.
- [4] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *Proceedings of the 1997 ACM SIGMOD*, 1997.
- [5] R. V. Renesse, K. Birman, and W. Vogels. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Transactions on Computer Systems*, 21(2):164–206, 2003.

²Aggregation has been studied for many years, resulting in a number of significant works with both theoretical and practical contexts. Due to space limitations, we do not further elaborate these contributions.