Gossip-based Distribution Estimation in Peer-to-Peer Networks

Maya Haridasan, Robbert van Renesse
Computer Science Department, Cornell University
{maya, rvr}@cs.cornell.edu

***Abstract*— We propose a novel gossip-based technique that allows each node in a system to estimate the distribution of values held by other nodes. We observe that the presence of duplicate values does not significantly affect the distribution of values in samples collected through gossip, and based on that explore different data synopsis techniques that optimize space and time while allowing nodes to accumulate information. Unlike previous aggregation schemes, our approach focuses on allowing all nodes in the system to compute an estimate of the entire distribution in a decentralized and efficient manner. We evaluate our approach through simulation, showing that it is simple and scalable, and that it allows all nodes in the system to converge to a satisfactory estimate of the distribution in a small number of rounds.**

I. Introduction

The knowledge of how one ranks relative to peers can be put to a variety of uses. It allows outliers to be detected, overall trends to be observed, and informed predictions to be made. A tool that allows nodes to maintain an estimate of what values other peers hold for particular properties can help systems be more resilient and self-adapt in sophisticated ways. In this work, we propose an approach where nodes build such estimates in a timely and scalable manner. Our approach relies on gossip-style exchange of data, and uses data synopsis techniques for minimizing the amount of data exchanged between pairs of nodes. Nodes maintain a fixed-size array of entries and periodically exchange and accumulate information obtained from other peers.

Previous work has focused on the diagnosis of individual aggregate values, such as averages, sums, minimum and maximum values of distributions. Tree-based approaches compute aggregates hierarchically and under no-failure scenarios allow exact values to be computed [9], [10]. In the presence of node failures or nodes joining and leaving the system, decentralized gossip-based techniques present a more resilient model, even though computation of exact aggregates may not always be possible [6], [5]. We do not attempt to present nodes with exact distribution models since that would lead to high costs without adding significant benefits, but instead focus on providing a more expressive model that provides nodes with an approximation of the entire distribution rather than just individual aggregates.

To restrict the storage and communication costs, we explore previous work on data synopsis from the database community, originally for approximate query answering in the context of large repositories. Typically, the goal is that within a single pass through all the data a concise representation be created which allows queries to be answered within short delays of time. Two main differences in using these techniques within our gossiping context are that: (a) a large number of duplicates are present in the sample; and (b) not all data is available to every node. We will argue in the paper that the presence of duplicates does not significantly bias the estimated distributions, and that it is therefore simpler and more efficient to leverage their presence in the samples than attempting to remove them.

We evaluated our proposed gossip-based approach through simulation when coupled with four data synopsis techniques. We compared these in terms of quality of the estimation, storage and bandwidth requirements, and convergence time. All techniques were evaluated with a diverse set of distributions, including uniform, normal, heavy-tailed and bimodal distributions. By experimenting with up to 100 K nodes, we have empirically validated that a limited number of rounds and a constant message throughput per node in each round is sufficient to achieve an efficient and lightweight protocol.

II. Design

A. Problem Statement

We assume a system consisting of N nodes with identifiers 1 to N. Each node i holds a numerical value

The authors were supported by AFRL award FA8750-06-2-0060 (CASTOR), NSF award 0424422 (TRUST), AFOSR award FA9550-06-1-0244 (AF-TRUST), DHS award 2006-CS-001-000001 (I3P), and Intel Corporation. The views and conclusions herein are those of the authors.

x_i that measures some variable of interest to the system. The set of values X held by all nodes may follow any arbitrary distribution. The main goal of our protocol is that within a predetermined interval of time, every node is able to produce a satisfactory estimate of the distribution of values x_i held by all nodes in the system.

The set of values held by nodes may vary with time. The protocol executes in phases, which are in turn subdivided into rounds. Within each phase, each node converges to an estimate of the distribution. The estimates produced at the end of each phase are an approximation of the distribution of values held at the beginning of the phase. When a new phase is started, old values are discarded in favor of newer ones.

The notion of what is a satisfactory estimate of the distribution is subjective, and may vary depending on the purpose of the application using the estimated distribution. Instead of attempting to achieve perfect accuracy, we focus on the best balance between space overhead and accuracy, also taking into account the time complexity of the proposed solution.

B. Basic Protocol

Nodes execute in rounds and phases. A phase is the larger time interval in which each node produces an estimate of the distribution of values. Each phase is composed of rounds of approximately fixed duration δ (e.g. 1 second). Even though rounds have a fixed duration, strict time synchronization among nodes is not required since time is only used as a rough guideline for nodes to be aware of when to proceed to the next step of the protocol. Each node is responsible for advancing rounds based on its local clock, and advancing phases when it establishes some criteria that indicates that a phase has completed. In this subsection we focus on the steps followed by each node within a single phase.

We assume that nodes maintain a set of neighbors at any given time, by using a decentralized membership protocol such as the one proposed in [2]. Each node maintains a local view (its set of neighbors), and periodically updates it by randomly picking from the local views of its neighbors and from other nodes that contact it in the previous round. Nodes always remember a list of at least as many live distinct nodes as the number of rounds in a phase (usually between 15 and 20).

Every node maintains an array of k numerical values. At the beginning of each phase, all k values in the array are set to the value x_i, originally held by the node. In each round, a node i randomly chooses a partner j and requests the set of values stored by the partner. Once it receives the array of values from j, node i has $2k$ values, which get merged into an array of size k. In the simplest protocol, hereafter called *Swap*, merging consists of randomly picking k of these values and discarding the others.

With the *Swap* protocol, nodes randomly discard data previously available to them, therefore loosing important information when estimating the distribution of values. Data synopsis techniques allow peers to store data previously seen with limited loss of information and consume less space. We next consider three such synopsis construction techniques.

1) Concise Counting: The first technique we considered is an adaptation of the *counting samples* approach proposed in [4] for compressing data in large datawarehouses. In the original approach, values appearing more than once in the sample are represented as a value and a count pair. Given that we are dealing with floating point numbers and have fixed storage space, the following adaptations were made: all entries in our array are tuples of <value, counter>. Whenever new values are added to the sample, the tuples are sorted based on their values, and the closest values in the sample are merged together, so that only a fixed number of tuples are in the sample at any given time. Merging two tuples consists of randomly picking one of the two values and adding their respective counters.

2) Equi-Width Histograms: A straightforward histogram technique consists in breaking the range of possible values into equal sized bins, and maintaining counters for each bin. One difficulty with this *Equi-width* approach occurs when nodes are not aware of what the extreme values of the distribution are. In our implementation, each node i initially considers the set of values to range from 0 to the value they hold (x_i), and later resizes the bins dynamically in case new values beyond the extremities are found. When resizing, each old bin is mapped to a larger new bin, based on the middle value of the old bin, and the ranges of the new resized bins. The counter of each old bin is added to the new bin to which it is mapped. The main advantage of the *Equi-width* approach when compared to the *Concise* approach is that since bins have equal width, only the extreme values of the whole distribution and counters for each bin need to be stored, reducing the amount of data stored and transferred.

3) Equi-Depth Histograms: Dividing the range of values into equi-width partitions may lead to very inaccurate estimations depending on the original distribution of values. Another choice consists in using equi-depth bins, where each bin contains an approximately equal number of points. In our implementation of the *Equi-depth* histogram approach, each node i initially divides

the range [0, x_i] into fixed sized bins, each represented by a pair of <value, counter>. A simple protocol is used to later merge or split bins based on their counters as new data is inserted. After exchanging data with another peer, each node orders all collected pairs of <value, counter> and computes which consecutive bins, when merged, yield the smallest combined bin. The identified bins are merged (their counters are added and the weighted arithmetic mean of their values is used as the value of the new bin) and the process is repeated until only the desired number of bins are left. The main goal of this process is to minimize the disparity across all bins.

III. Evaluation

We built a round-based simulator to evaluate our proposed gossip-based approach and to compare its behavior when coupled with the four data synopsis techniques. Unless otherwise stated, we ran experiments simulating 10 K nodes with partial connectivity. Nodes held arrays containing 50 values, which were simultaneously updated only at the end of each round.

We used the Kolmogorov-Smirnov distance as the quality metric of a distribution estimate relative to the original distribution. The KS-distance measures the maximum vertical distance between the actual cumulative distribution and the cumulative estimated distribution. In practical terms, it measures the maximum disparity between the real and estimated percentages of nodes that hold more or less than any particular value. For instance, a KS-distance of 0.1 implies that the estimated percentage of nodes larger and smaller than some particular value might be off by up to 10%.

Therefore, the KS-distance is a general metric for evaluating the quality of the estimations for calculating percentiles. We always present the *maximum* KS-distance across all nodes in the system, which represents the worst-case estimation, since we aim for a protocol that allows all nodes to compute satisfactory estimates. Even though of interest, due to limited space individual aggregates such as mean, medium, min/max and others are omitted given that they are less general than the KS-distance when evaluating estimated distributions.

A. Effect of Duplicates

In our first experiment we explore the effect of duplicate values in the data samples accumulated by nodes. To estimate the time and data required in an optimal data collection scenario, we considered an unpractical protocol where nodes use gossip to exchange vectors that accumulate all data received from peers (*Gossip with Duplicates*). Nodes exchange larger arrays as rounds progress, and arrays contain duplicate values. Next, we considered a similar setting, but in which duplicates were removed (*Gossip with Duplicates Removed*). Our goal with these experiments was to analyze the penalty incurred by keeping duplicates in the collected samples.

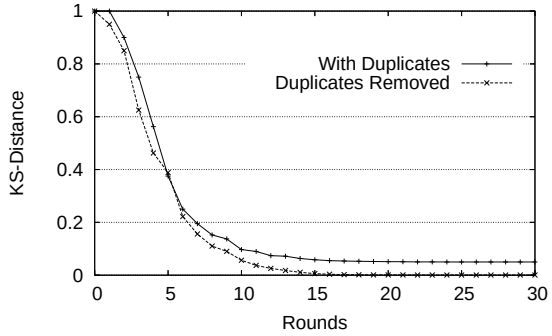


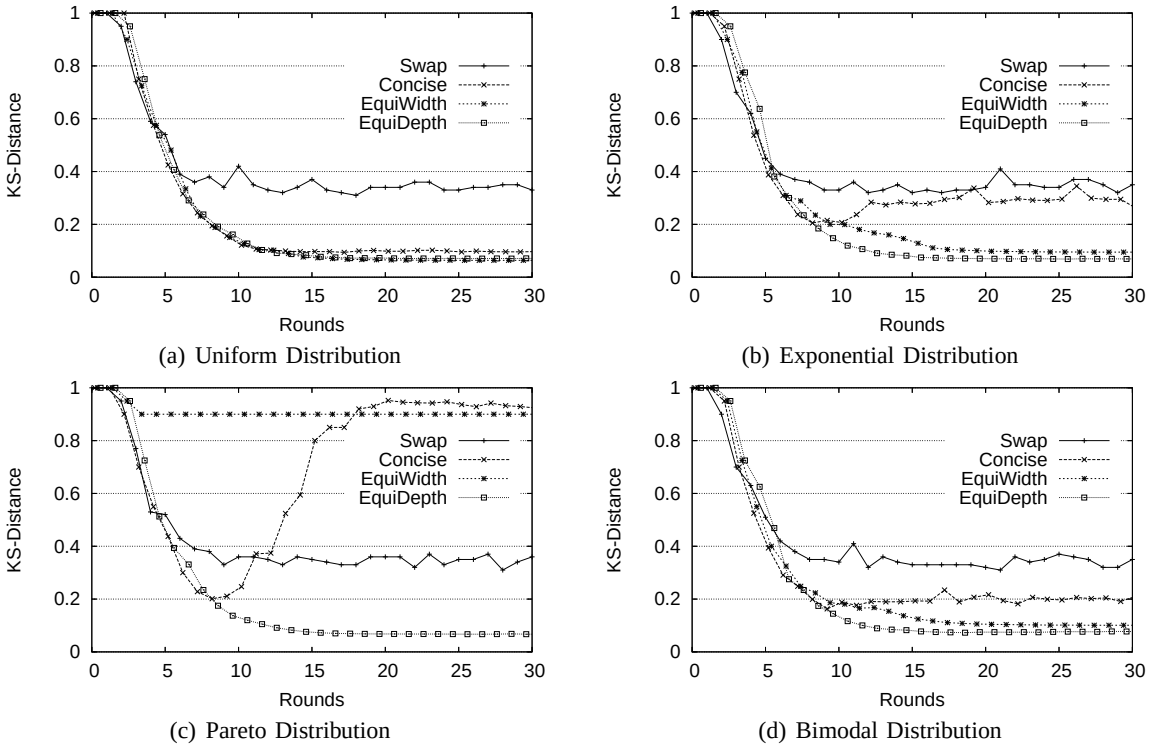
Fig. 1. Effect of removing duplicates from collected sample

In Figure 1, a comparison of the two experiments is presented over increasing numbers of rounds. In the presented example the set of values held by nodes followed an exponential distribution; the results for other distributions were similar or better and are omitted from the paper. The curve for the setting where duplicates are removed shows that all nodes converge to the ideal distribution around the 15th round (the lines show the metrics for the worst-case node at any round). When duplicates are not removed, nodes converge to a maximum KS-distance of approximately 0.06, again around the 15th round. This difference in quality of the estimates is the tradeoff for the simplicity of not having to remove duplicates from the samples. All data summarization techniques we evaluate can perform at best as well as the curve for *Gossip with Duplicates*.

B. Sample Distributions

We compared the performance of the four data summarization techniques when combined with the gossip protocol in terms of quality of the estimates under a diverse set of distributions. We considered uniform, normal, exponential, Pareto, chi-square, lognormal, Weibull and multimodal distributions, all with varying parameter values. Due to our limited space, we only present graphs for four representative distributions: uniform, exponential ($\lambda = 1.5$), Pareto ($k = 5$, $x_m = 1$), and one bimodal distribution composed by adding two normal distributions (with parameters $\mu_1 = 5, \sigma_1 = 1$, and $\mu_2 = 8, \sigma_2 = 0.5$).

Among the distributions considered, uniform was the easiest to estimate, as observed in Figure 2(a). While the *Swap* technique falls behind with a large KS-distance, the three other summarization techniques perform well, with KS-distances always smaller than 0.1. This obviates



(a) Uniform Distribution (b) Exponential Distribution

(c) Pareto Distribution (d) Bimodal Distribution

Fig. 2. Maximum KS-distance across all nodes under different distributions.

the importance of accumulating enough values to make adequate predictions. It is worth mentioning that in the *Swap* approach some nodes are able to estimate the distribution as well as nodes in the other three approaches (not shown in the graph). However, as previously stated, we use the worst node's estimate as a metric since we expect all nodes to achieve satisfactory knowledge.

Next, we considered an exponential distribution, which as observed, affected the performance of the *Concise* technique (Figure 2(b)). The main reason for this technique's poor performance is that it maintains a fixed number of <value,counter> pairs, and merges the pairs with closest values whenever needed. In distributions where data is not evenly distributed, the approach uses most space storing dispersed values which represent a minority of the values in the system.

The *Equi-Width* histogram approach suffers from the same problem as the *Concise* approach since it divides the space of values into equal-sized bins. While the *Equi-Width* approach worked well with most of the distributions we considered, it failed to do so with heavy-tailed distributions such as the Pareto distribution considered for the experiment in Figure 2(c). In this distribution, most nodes hold small values that do not get differentiated into separate bins, which leads to poor computation of percentiles and large KS-distance metrics.

The bimodal distribution did not present further challenges when compared to the previous distributions despite the presence of two modes. A more detailed analysis confirmed that it is possible to compute the mean, median, and other percentiles accurately with the three later synopsis techniques.

As evidenced from the graphs for these four distributions, the *Equi-Depth* approach consistently performs well, maintaining the worst-case KS-distance metric around 0.07. This behavior was maintained when we experimented with several other distributions and parameters not presented in this paper. The *Equi-Width* technique also performed satisfactorily for most of the distributions, but as observed, significantly degrades under severely skewed distributions. The advantage of the *Equi-Width* technique lies on the fact that it requires only approximately half the space required by the *Equi-Depth* approach since only the extremity values and the bin size need to be stored.

Another approach to evaluate the different data summarization techniques consists in using the estimated distributions computed by the nodes to estimate the parameters of the original distributions (known a priori in a controlled setting). We show in Figure 3 how well the techniques perform in terms of estimating the parameters of exponential and Pareto distributions. On the x-axis we varied the value of the parameter used to generate the distribution of values across the nodes, and on the y-axis we present the actual worst-case estimates computed by the nodes at the end of 15 rounds.

A perfect estimation of values would be represented

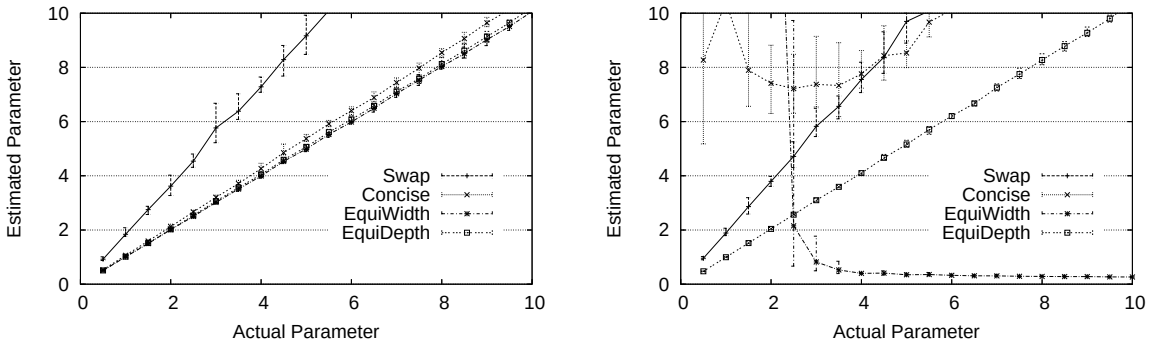


Fig. 3. Estimation of distribution parameters based on the data accumulated through different synopsis techniques

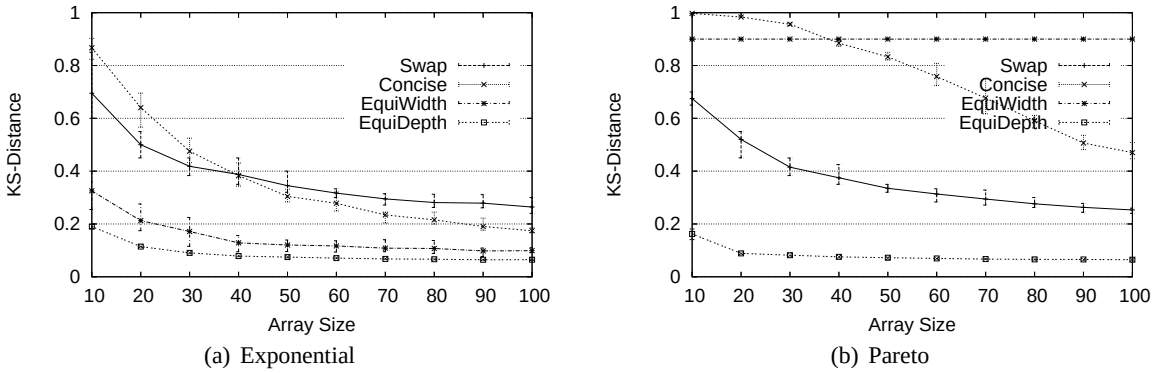


Fig. 4. Effect of array size in the quality of estimations

by the identity function. These curves validate the observations that the *Equi-Depth* approach yields the most accurate results, and shows how the *Equi-Width* and the *Concise* approach can lead to completely erroneous estimates of the original parameters of the distribution for heavy-tailed applications. Even though the *Swap* technique does not yield accurate distribution estimates, it is interesting to note that it performs consistently on all distributions.

We also performed experiments with larger numbers of nodes to study how the number of rounds varies with larger systems. We experimented with up to 100 K nodes using the *Equi-Depth* technique. One interesting thing that was observed was that the number of rounds for convergence remained 15, even when experimenting with 100 K nodes. While further analysis would be required to estimate the number of rounds for larger networks, the observed results are a positive indication on the time complexity of the protocol.

One particular property of our gossip-based approach is that it is able to capture the general distribution of values with satisfactory accuracy, but it does not necessarily register the effect of individual values. This means that extreme values are not accurately recorded by nodes, and in situations where a very small number of nodes significantly affects particular aggregate values, nodes may not be able to accurately estimate these. This problem may be alleviated by pairing our approach with previously proposed approaches to compute individual aggregates. Further work is required to study whether an efficient solution combining benefits of both approaches is feasible.

C. Array Size

One important parameter to consider in our approach is the size of the arrays used to store and exchange data. In all previous experiments we employed arrays with 50 elements. The types of elements in each array vary with each approach: floating point values for the *Swap* technique, integers for the *EquiWidth* approach, and pairs of <float, integer> for the *Concise* and *Equi-Depth* approaches. Storage-wise, the *Swap* and *Equi-Width* approaches were more efficient in the previous experiments. To confirm that adding further storage space to these techniques would not lead to different outcomes, we evaluated the effect the array-size has on the estimations.

In Figure 4, we present how the maximum KS-distance among nodes at the end of the 15th round varies with increasing array-sizes. We again present results for the exponential and Pareto distributions. Increasing the array-size did lead to improved results in most cases, as expected, but even with arrays of 100 elements, none of the three techniques is able to outperform the *EquiDepth*

technique with 50 elements. Another important point to notice is that the *EquiDepth* approach does not benefit significantly from using arrays containing more than 40 or 50 elements. Depending on the bandwidth requirements of applications, even 20 or 30 elements may produce satisfactory estimates.

IV. Related Work

In [3] a tree-based solution and some variants to the aggregation problem in P2P systems are proposed, with focus on queries issued by a single peer. In their basic scheme, the querying peer broadcasts the query to the network, and a spanning tree is constructed during the dissemination of the message. In the second phase of the protocol, the answer to the query is computed in a bottom-up fashion. Unlike our scheme, in the proposed model the peer issuing the query is the only one that obtains the information at the end of the aggregation process.

Astrolabe [9] and SDIMS [10] are management systems that hierarchically aggregate information about large-scale networked systems. Nodes are organized into a hierarchy and continuously compute aggregate values of the nodes immediately below them. Despite its decentralized nature, tree-based approaches are vulnerable to node failures and costs related to building and maintaining the tree-based hierarchy.

The idea of exchanging vectors containing multiple values for computing aggregate values was previously explored in the Newscast protocol [5]. In each round peers randomly select another peer to exchange all cache entries they currently hold. The choice of which cache entries are kept after the new entries are received is based on the age of the entries. Only the youngest entries are maintained, and the set of peers associated with each entry constitute the set of neighbors known by the owner of the cache. The use of the proposed solution was shown for computing extreme values and the mean of values.

A thorough survey of synopsis construction techniques for data streams is presented in [1]. The need for efficient synopsis techniques in the context of database systems has led to the proposal of a variety of techniques. Even though the general problem specification is different, many previously proposed techniques, with modifications, can be employed in the context of estimating distributions of values in P2P systems.

Work on gossip-based aggregation has also been done in the context of sensor networks, where energy and constant loss of communication are important factors to be considered. TAG [7] proposes a tree topology to compute aggregates without spending much energy, and avoiding duplicate information. [8] proposes the diffusion of synopsis, but focuses on finding solutions that avoid double-counting. By proposing techniques that are duplicate-insensitive, different topologies can be used for collecting information, and redundant paths can be explored to avoid loss of data when nodes fail.

V. Conclusion

In this paper we proposed and evaluated a scalable gossip-based technique that allows nodes to estimate distributions of values held by other peers. Unlike approaches which attempt to compute individual aggregates of values, our approach aims at complementing this information with knowledge about how any value ranks relative to others. We compared different synopsis techniques for compressing data that is gossiped among nodes, thereby saving space required for storing and exchanging data. We observed that the data synopsis technique adopted can severely impact the quality of the estimates collected, and that the *Equi-depth* histogram technique provides a good balance between space requirements and quality of estimation.

References

[1] C. C. Aggarwal and P. S. Yu. A Survey of Synopsis Construction in Data Streams. In *Data Streams: Models and Algorithms*, chapter 9. Springer-Verlag New York, LLC, 2006.

[2] A. Allavena, A. Demers, and J. E. Hopcroft. Correctness of a gossip based membership protocol. In *Proc. of the 24th ACM Symposium on Principles of Distributed Computing*, Las Vegas, NV, 2005.

[3] M. Bawa, H. Garcia-Molina, A. Geonis, and R. Motwani. Estimating Aggregates on a Peer-to-Peer Network. In *Technical Report*, Stanford University, 2003.

[4] P. B. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. In *Proc. of ACM SIGMOD International Conference on Management of Data*, Seattle, WA, 1998.

[5] M. Jelasity, W. Kovalczyk, and M. van Steen. An Approach to Massively Distributed Aggregate Computing on Peer-to-Peer Networks. In *Proc. of Euromicro Conference on Parallel, Distributed and Network-Based Processing*, A Coruña, Spain, 2004.

[6] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. of the IEEE Symposium on Foundations of Computer Science*, Washington, DC, 2003.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a Tiny AGgregation Service for Ad-hoc Sensor Networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.

[8] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. Synopsis Diffusion for Robust Aggregation in Sensor Networks. In *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, 2004.

[9] R. van Renesse, K. P. 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.

[10] P. Yalagandula and M. Dahlin. A Scalable Distributed Information Management System. In *Proc. of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Portland, OR, 2004.