Functional Geometric Monitoring for Distributed Streams

Vasilis Samoladas
ECE, Technical University of Crete.
IMIS, Athena R.C.
vsam@softnet.tuc.gr

Minos Garofalakis
ECE, Technical University of Crete.
IMIS, Athena R.C.
minos@softnet.tuc.gr

ABSTRACT

We introduce *Functional Geometric Monitoring (FGM)*, a substantial theoretical and practical improvement on the core ideas of Geometric Monitoring. Instead of a binary constraint, each site is provided with a non-linear function, which, applied to its local summary vector, projects it to a real number. The sites collectively monitor the sum of these one-dimensional projections and as long as the global sum is subzero, the monitoring bounds are guaranteed. We demonstrate that FGM is as generally applicable as Geometric Monitoring, and provides substantial benefits in terms of performance, scalability, and robustness. In addition, in FGM it is possible to prove worst-case results, under standard monotonicity assumptions on the monitoring problem. In terms of performance, the salient quality of FGM is that it can adapt naturally to adverse changes in the monitored problem, such as lack of monotonicity or very tight monitoring bounds, where no method can deliver asymptotically good performance. We provide formal proofs for many of the properties of FGM, and present an extensive empirical performance evaluation under adverse conditions, on real data.

KEYWORDS

distributed functional monitoring, geometric monitoring, distributed streaming

1 INTRODUCTION

The explosion in the amount of data generated online is entering its next phase, as the Internet of Things (IoT) is set to increase the number of networked data sources by orders of magnitude in the near future. Thus, there is a clear need for ever more scalable techniques for *distributed* stream processing, where the tsunami of data generated by networked nodes is filtered and summarized in near-real time at (or, near) the source, drastically reducing the required communication costs.

Motivated by such needs, there has been significant research effort on the distributed functional monitoring problem, over the past decade. Much effort has concentrated on worst-case communication complexity for particular types of important queries, such as frequency moments, heavy hitters, percentiles, distinct elements etc. Early on, it became apparent that many of these problems can have very bad worst-case performance, unless certain assumptions were made, in particular with respect to monotonicity, about the input streams and/or the monitored functions.

A problem not yet addressed by previous work on monitoring algorithms, is that it can be a challenge to *integrate* them into large stream processing frameworks, such as STORM and Spark, or even more targeted systems such as Gorilla and InfluxDB. Indeed, while such frameworks are a staple of modern information systems, the algorithmic work on distributed functional monitoring has not yet found practical adoption in them. Part of the reason is, we believe, lack of uniformity; each distributed monitoring algorithm imposes different requirements on the system architecture and its communication patterns. Historically, this was also the case with databases, until adoption of the relational model (its limitations notwithstanding) created a vibrant industrial and research ecosphere.

A technique—to our knowledge, the only one—intended to be applicable to *arbitraty continuous queries* is Geometric Monitoring (GM) [27]. A strong appeal of GM is that it separates the complexities of the monitored query operator, from the communication protocol that executes the monitoring. Unfortunately, although it can be very successful at reducing the communcation cost in real-world applications, its performance can degenerate under certain circumstances, such as high stream variability, or skew between the relative rates of local streams. On the theoretical side, the GM is not known to provide any cost guarantees, even under monotonicity assumptions.

Related Work. After the introduction of (centralized) streaming algorithms in the mid '90s, several works proposed *distributed streaming* techniques for particular important problems, such as linear functions [16, 17, 23], top-k queries [3, 24], ratio thresholding queries [15], and polynomials of scalar variables [26]. Of particular importance is the problem of tracking sketch synopses [7] of local streams, which can be applied to the approximation of self-join and join aggregates [5, 6].

The first (and, to our knowledge, only) general-purpose technique, Geometric Monitoring (GM), was first proposed in [27, 28]. This paper ignited a rich line of work, part of which related to improving the basic method [14, 18–22], and also utilizing it in important applications (e.g., [4, 11, 12, 25] to name but a few). Interestingly, despite the rich mathematical techniques employed in this body of work, to date there have been no analytical results on the *communication cost* of the method, even under strict assumptions.

Starting with the fundamental results of Cormode et al. [9], the problem of continuous query tracking over distributed streams has also been studied in a theoretical setting in recent years, within the broad framework of communication complexity; the minimum amount of bits that needs to be exchanged between a group of communicating parties, each party observing incrementally a local dataset, so that a global function on the union of the data possessed by all players can be *continuously* tracked with some bounded error. Much of the work has concentrated on the hardness of the problem. It has been shown [1] that the worst-case communication cost of distributed function monitoring can hardly improve upon the baseline method of centralizing all local data, unless restrictive assumptions are made; this is true even for the trivial problem of maintaining a distributed counter. Despite the general negative results, there are also positive results for particular problems of interest, e.g., [1, 9, 29, 30] to name but a few.

In particular, [9] provides optimal results on the communication complexity of monitoring a linear monotone function, while subsequent papers [1, 29] prove strong lower-bounds showing that guaranteeing better than linear worst-case communication costs is probably impossible for complex, non-linear query functions.

Our Contributions. We propose *Functional Geometric Monitoring (FGM)*, a technique which can conceptually be applied to any monitoring problem, in order to perform distributed monitoring with communication costs that are lower (often by orders of magnitude) compared to centralizing all data to a coordinator. The FGM comprises of a distributed algorithm which is independent of the monitoring problem. In order to perform a monitoring task, the FGM must be parameterized by a problem-specific family of functions (termed safe functions later in the paper); to this end, the FGM draws on and can utilize the extensive previous work on distributed monitoring, where such functions have been proposed for a large variety of monitoring problems [11, 13, 21]. Combined with these previous results, FGM is a technique that is ready to be utilized to real distributed monitoring applications.

The strict separation of concerns between distributed systems issues and the monitoring problem, is critically important to anyone wishing to implement distributed monitoring on a general-purpose middleware platform. In addition, despite this strict separation, the FGM offers significant improvements to the communication cost of distributed monitoring, compared to previous monitoring techniques, notably the GM. In particular, FGM has provably better performance than GM, regardless of the monitoring problem. In fact, under FGM it is possible to provide good worst-case guarantees on the communication cost of specific monitoring problems, comparable to the best known theoretical results on distributed functional monitoring. By contrast, no such results are known for GM. In this paper, we provide such worst case analytical results for monitoring frequency norms.

Another issue that has not been treated uniformly—and has often been ignored—by previous techniques, is the detection and response of the monitoring algorithm to circumstances where the monitored constraints (thresholds) are too "tight"; in such occasions, any distributed monitoring algorithm would be unable to do better than to naively centralize all data to a coordinator. Such situations occur frequently in practice, and practical monitoring algorithms should be able to smoothly transition their operation for handling such loads. An important fetaure of FGM is that it can adapt to these high variability situations seamlessly; that is, in a problem-independent manner, and within the logic of the basic protocol.

In addition, FGM's performance is resilient to the presense of skew in the distribution of data among distributed nodes, as well as in the relative rates of local streams; its performance is fundamentally determined by the characteristics of the global stream (i.e., the union of all distributed streams). Again, this is a novel feature; the performance of previous techniques, notably of GM, is adversely affected in the presense of skew. We have performed extensive experiments that demonstrate and quantify the resilience of FGM both in adverse streaming conditions of high variability, and in the presense of skew.

2 FUNCTIONAL GEOMETRIC MONITORING

The focus of this section is to present the basic principles and protocol of Functional Geometric Monitoring (FGM). Our discussion employs standard notation and terminology from functional analysis: Vectors are denoted by boldface letters, and sets of vectors are added by Minkowski addition:

$$A + B = \{\boldsymbol{x} + \boldsymbol{y} \mid \boldsymbol{x} \in A,\ \boldsymbol{y} \in B\}.$$

We write $\boldsymbol{x} + A$ instead of $\{\boldsymbol{x}\} + A$; also, $\lambda A = \{\lambda \boldsymbol{x} | \boldsymbol{x} \in A\}$. Finally, in some proofs, we assume some familiarity with the properties of convex functions and sets; in particular, the biconjugate (convex hull of a function), norms and semi-norms, gauge functions and convex cones. The convex hull of A is $\operatorname{conv} A$.

2.1 Background: Approximate Query Monitoring

We adopt the standard data model for distributed data streams. Assume that there are k distributed sites, and that at each site, a local stream is generated or collected, denoted as a (very high-dimensional) vector in vector space $\mathbb{R}^D$. This vector can be the frequency vector of the stream records, or a linear sketch thereof, and changes as stream updates arrive. Let $\boldsymbol{S}_i(t), i = 1 \ldots k$ denote the local state vectors. Every site communicates with a coordinator, where users pose queries on the global stream. Without loss of generality, assume that the global stream state is the average of the local stream states, i.e., $\boldsymbol{S}(t) = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{S}_i(t)$. (Other linear formulas, e.g., sum, can be treated by multiplying local state vectors with scalars as needed.)

We consider two types of queries on this model. In the *one-shot* query, the coordinator needs to monitor for the event $Q(\boldsymbol{S}(t)) \leq T$, where Q is a query function and T a threshold. On the other hand, for a *continuous* query, the coordinator needs to maintain at all times a close estimate $Q(\boldsymbol{E}(t))$ of the true value of the query $Q(\boldsymbol{S}(t)$, so that

$$Q(\boldsymbol{S}(t)) \in (1 \pm \varepsilon) Q(\boldsymbol{E}(t)). \tag{1}$$

This guarantee is maintained by the local sites periodically flushing the updates received to their local streams. In particular, the coordinator maintains, for each site i, an estimated state vector $\boldsymbol{E}_i$. When a flush occurs, the site transmits its *drift vector* $\boldsymbol{X}_i(t) = \boldsymbol{S}_i(t) - \boldsymbol{E}_i$, and the coordinator updates $\boldsymbol{E}_i$ by adding $\boldsymbol{X}_i$ to it, while the site resets $\boldsymbol{X}_i$ to 0. Then, the coordinator updates the global estimate $\boldsymbol{E} = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{E}_i$.

Note that a site can transmit its drift $\boldsymbol{X}_i$ either as a vector of size D, or as a list of the records that arrived since the previous flush (whichever is smaller), therefore the total communication cost for flushing is never worse than tranmitting all data to the coordinator.

In geometric monitoring, the correctness criterion is described as a geometric constraint, of the form $\boldsymbol{S} \in A$, where $A \subseteq \mathbb{R}^D$ is the *admissible region*, that is, the set of global stream states where the constraint holds. That is,

$$A = \left\{\boldsymbol{x} \in \mathbb{R}^D \mid Q(\boldsymbol{x}) \leq T\right\} \qquad \text{one-shot queries} \tag{2}$$

$$A = \left\{\boldsymbol{x} \in \mathbb{R}^D \mid Q(\boldsymbol{x}) \in (1 \pm \varepsilon) Q(\boldsymbol{E})\right\} \qquad \text{continuous queries} \tag{3}$$

2.2 Communication costs

We assume that each message consists of a sequence of *words*, of sufficient size. In particular, we assume that each word can store a real number; all our protocols are robust against finite precision, so this is not an unrealistic assumption.

We distinguish two directions in communication. *Downstream communication* consists of messages from local nodes to the coordinator, while *upstream communication* consists of messages from

the coordinator to local nodes. We do not consider a multicast capability, although our work can be adapted to such settings.

2.3 Safe functions

Our starting point relates to the representation of a safe *configuration*, in a monitoring algorithm. The *configuration* of the system of k sites is a is a (kD)-dimensional vector consisting of the *concatenation* of the k local drift vectors, $\boldsymbol{X}_i$. The system is in a safe state as long as $\boldsymbol{E} + \frac{\sum_{i=1}^{k} \boldsymbol{X}_i}{k} = \boldsymbol{S} \in A$.

To guarantee that a configuration is safe, FGM employs a real function $\phi : \mathbb{R}^D \to \mathbb{R}$, depending on A, $\boldsymbol{E}$ and k. Each site tracks its ϕ-value, $\phi(\boldsymbol{X}_i)$, as $\boldsymbol{X}_i$ is updated. System safety is guaranteed by tracking the sign of the *sum* $\psi = \sum_{i=1}^{k} \phi(\boldsymbol{X}_i)$. In particular, we need to guarantee that $\psi \leq 0$ implies $\boldsymbol{S} \in A$.

Definition 2.1 ($(A, \boldsymbol{E}, k)$-safe function). A function $\phi : \mathbb{R}^D \to \mathbb{R}$ is *safe* for admissible region $A \subseteq \mathbb{R}^D$, vector $\boldsymbol{E} \in \mathbb{R}^D$, and $k \geq 1$, if, $\phi(\boldsymbol{0}) < 0$ and, for all $\boldsymbol{X}_i \in \mathbb{R}^D$, $i = 1, \ldots, k$,

$$\sum_{i=1}^{k} \phi(\boldsymbol{X}_i) \leq 0 \implies \boldsymbol{E} + \frac{\sum_{i=1}^{k} \boldsymbol{X}_i}{k} \in A.$$

Much of previous work on distributed monitoring has proposed safe functions for specific problems. Since we indend to explore the FGM in terms of its generality, we are interested in properties of safe functions as a class.

First, note that safety is preserved under common pointwise operations: positive scaling, addition and pointwise supremum. Consequently, $(A, \boldsymbol{E}, k)$ safety is monotone under pointwise dominance, i.e., if ϕ is $(A, \boldsymbol{E}, k)$-safe and $\forall \boldsymbol{x}, \phi(\boldsymbol{x}) \leq \phi'(\boldsymbol{x})$, then ϕ' is also $(A, \boldsymbol{E}, k)$-safe.

In fact, more can be said about addition and pointwise supremum of safe functions; they can be employed to *compose* safe functions for an admissible region A, defined by a set-algebraic expression over some family of admissible regions $\{A_i\}$, when a safe function ϕ_i for each A_i is available.

Theorem 2.2. *Let ϕ_i be $(A_i, \boldsymbol{E}, k)$-safe, for each $i \in I$ respectively. Then,*

- $\sup_{i \in I} \phi_i$ *is* $(\bigcap_{i \in I} A_i, \boldsymbol{E}, k)$*-safe, and*
- $\sum_{i \in I} \phi_i$ *is* $(\bigcup_{i \in I} A_i, \boldsymbol{E}, k)$*-safe, provided that I is finite.*

On the dependence on k, note that, for any divisor k' of k, a k-safe function is also k'-safe , and 1-safe in particular. Therefore a necessary condition for k-safety is that the 0-sublevel $L(\phi) = \{\boldsymbol{x} | \phi(\boldsymbol{x}) \leq 0\}$, shifted by $\boldsymbol{E}$, be a subset of A:

$$\boldsymbol{E} + L(\phi) \subseteq A.$$

2.3.1 Safe functions and convexity. Intuitively, we can "improve" a safe function ϕ by finding a function $\phi' \leq \phi$ that is still safe; ϕ' is improved in the sense that the set of configurations where $\psi \leq 0$ is larger for ϕ' than for ϕ.

In this respect, of particular interest are functions that are safe for all k (and fixed A, $\boldsymbol{E}$). We denote such functions as $(A, \boldsymbol{E})$-safe. The salient property of $(A, \boldsymbol{E})$-safe functions is that they can always be "improved" into dominated, *convex* safe functions.

Theorem 2.3. *If a function ϕ is $(A, \boldsymbol{E})$-safe, there exists a **convex** function $\zeta \leq \phi$ which is also $(A, \boldsymbol{E})$-safe.*

Proof. Ommitted due to space constraints. □

Convex safe functions are appealing because they admit a very simple criterion for $(A, \boldsymbol{E})$-safety.

Lemma 2.4. *A convex function ζ with $\zeta(\boldsymbol{0}) < 0$ is $(A, \boldsymbol{E})$-safe, if and only if, $\boldsymbol{E} + L(\zeta) \subseteq A$.*

Proof. We have already seen that $\boldsymbol{E} + L(\zeta) \subseteq A$ is necessary for safety. Sufficiency follows directly from the convexity of ζ. □

2.3.2 Quality of safe functions. Consider the set $C \subseteq \mathbb{R}^{kD}$ of safe configurations of a monitoring problem

$$C = \left\{(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k) \mid \boldsymbol{E} + \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{X}_i \in A\right\} \tag{4}$$

The FGM protocol under-approximates this set by the set of configurations, where $\psi \leq 0$. Call this the *quiescent region* $\boldsymbol{Q}_\phi$, which is determined by the choice of safe function ϕ. We would like to characterize ϕ so that $\boldsymbol{Q}_\phi$ is as large (inclusion-wise) as possible, in order to improve the approximation of C. Such a characterization is possible if we restrict our attention to $(A, \boldsymbol{E})$-safe (and by virtue of the above theorem, convex) functions.

Intuitively, the issue that we examine can be presented with an example: assume that $A = \{\boldsymbol{x} | \|\boldsymbol{x}\| \leq 1\}$ is the unit ball, and take $\boldsymbol{E} = \boldsymbol{0}$. It is easy to see that both convex functions $\|\boldsymbol{x}\| - 1$ and $\|\boldsymbol{x}\|^2 - 1$ are suitable safe functions. However, the former choice is superior to the latter, when the size of the quiescent region is taken into account. To see this, note that $(1/2)(\|\boldsymbol{x}\|^2 - 1)$ strictly dominates $\|\boldsymbol{x}\| - 1$; therefore, a configuration say, $(\boldsymbol{0}, \boldsymbol{p})$ with $\|\boldsymbol{p}\| = \sqrt{3}$ is quiescent for $\|\boldsymbol{x}\| - 1$ but not for $\|\boldsymbol{x}\|^2 - 1$.

It turns out that safe functions that are best, are those that are *level-minimal*, that is, they do not strictly dominate any function with equal level set.

Theorem 2.5. *A $(A, \boldsymbol{E})$-safe function ϕ has maximal quiescent region, among all $(A, \boldsymbol{E})$-safe functions, for every k, iff,*

- *ϕ is convex*
- *$L(\phi)$ is a maximal convex subset of A, and*
- *ϕ is level-minimal*

Proof. Ommitted due to space constraints. □

The above results highlight the centrality of convexity in the monitoring problem; starting from very broad principles, we have shown that convexity enters in a natural way from the definition of safety, and furthermore, we have formally identified the requirement of level minimality, in order to maximize the quiescent region in FGM.

We will return to the issue of safe functions, with respect to the communication costs they entail, after we present the FGM distributed protocol.

2.4 The basic FGM protocol

The FGM protocol works in rounds. Monitoring the threshold condition

$$\sum_{i=1}^{k} \phi(\boldsymbol{X}_i) \leq 0, \tag{5}$$

over the duration of the round is performed along the lines of the algorithm in [9].

At the beginning of a round, the coordinator knows the current state of the system $\boldsymbol{E} = \boldsymbol{S}$. It selects an $(A, \boldsymbol{E}, k)$-safe function ϕ. At each point in time, let $\psi = \sum_{i=1}^{k} \phi(\boldsymbol{X}_i)$. The round's steps are:

(1) At the beginning of a round, the coordinator ships ϕ to every site (it is sufficient to ship vector $\boldsymbol{E}$, since A can then be determined from it). Local sites initialize their drift vectors to $\boldsymbol{0}$. With these settings, initially it is $\psi = k\phi(\boldsymbol{0})$.

(2) Then, the coordinator initiates a number of subrounds, to be described below. At the end of all subrounds, $\psi > \epsilon_\psi k\phi(\mathbf{0})$, for some small ϵ_ψ (Note that ϵ_ψ is not related to the desired accuracy for the monitored query, ε, but only to the desired quantization for monitoring ψ. We have used $\epsilon_\psi = 0.01$ in our experiments).
(3) Finally, the coordinator ends the round by collecting all drift vectors and updating $\boldsymbol{E}$.

2.4.1 Execution of subrounds. The goal of each subround is to monitor the condition $\psi \leq 0$ coarsely, with a precision of roughly θ, performing as little communication as possible. Subrounds are executed as follows:

(1) At the beginning of a subround, the coordinator knows the value of ψ. It computes the subround's *quantum* $\theta = -\psi/(2k)$, and ships θ to each local site. Also, the coordinator initializes a counter $c = 0$. Each local site records its initial value $z_i = \phi(\boldsymbol{X_i})$, where $2k\theta = -\sum_{i=0}^{k} z_i$. Also, each local site initializes a counter $c_i = 0$.
(2) Each local site i maintains its local drift vector $\boldsymbol{X}_i$, as it processes stream updates. When $\boldsymbol{X_i}$ is updated, site i updates its counter
$$c_i := \max\{c_i, \lfloor \frac{\phi(\boldsymbol{X}_i) - z_i}{\theta} \rfloor\}.$$
If this update increases the counter, the local site sends a message to the coordinator, with the increase to c_i.
(3) When the coordinator receives a message with a counter increment from some site, it adds the increment to its global counter c. If the global counter c exceeds k, the coordinator finishes the subround by collecting all $\phi(\boldsymbol{X}_i)$ from all local sites, recomputing ψ. If $\psi \geq \epsilon_\psi k\phi(\mathbf{0})$, the subrounds end, else another subround begins.

The following simple statement guarantees the correctness of the protocol.

Proposition 2.6. *During the execution of a subround, if $c \leq k$ then $\sum_{i=1}^{k} \phi(\boldsymbol{X}_i) < 0$.*

Proof. At each point in time and for any site, it must be
$$\frac{\phi(\boldsymbol{X}_i) - z_i}{\theta} - 1 < \lfloor \frac{\phi(\boldsymbol{X}_i) - z_i}{\theta} \rfloor \leq c_i.$$
Summing both sides, we get $\frac{1}{\theta}\left(\psi + 2k\theta\right) - k < c$, which simplifies to $\sum_{i=1}^{k} \phi(\boldsymbol{X}_i) < (c-k)\theta \leq 0$. □

2.5 Performance analysis

Apart from the communication incurred during the subrounds of the FGM protocol, the communication cost of a round consists of two parts; an upstream cost $\Theta(kD)$ for shipping $\boldsymbol{E}$ to all sites at the beginning of a round, and a downstream cost $O(\min\{kD, \tau\})$, for shipping drift vectors to the coordinator at the end of the round. Here, τ stands for the total number of stream updates processed by all sites during a round—as mentioned before, sites that received few stream updates during a round, can ship them verbatim to the coordinator.

2.5.1 The cost of subrounds. Each subround itself costs only $3k + 1$ one-word messages: k messages to broadcast quantum θ, k messages to collect ζ-values at the end of a subround, and up to $k + 1$ downstream messages carrying counter updates.

The problem of monitoring $\psi \leq 0$ is of course an instance of the non-monotone distributed counting problem. As shown in [9], if ψ is an increasing function of time, then the number of subrounds is at most $\log_2 \frac{1}{\epsilon_\psi}$.

In general, there is no guarantee that ψ will be increasing; therefore we also provide an analysis within the framework of *variability*, as set out in [10]. In this framework, the cost of tracking a non-monotone counter $f(t)$ within accuracy ε is shown to take $O(\frac{k}{\varepsilon} V_f)$ messages, where $V_f(t) = \sum_{\tau=0}^{t} \min\{1, \frac{|\delta f(t)|}{|f(t)|}\}$. They provide space-plus-time lower bounds for the tracking problem that match the communication cost.

In our setting, we are not interested in *tracking* the value of ψ. Still, the definition of variability during a round can be given as follows: let the sequence ψ_n represent the value of ψ at the end of the n-th subround. Also, define the set of values that $\phi(\boldsymbol{X}_i)$ takes, during subround n, as
$$\Phi_{i,n} = \{\phi(\boldsymbol{X}_i(t)) \mid \forall t \text{ during subround } n\}$$
Then the change $\Delta\psi_n$ is
$$\Delta\psi_n = \sum_{i=1}^{k} \sup \Phi_{i,n} - \inf \Phi_{i,n}$$
Finally, the ψ-variability over a round with q subrounds is
$$V = \sum_{n=1}^{q} \frac{|\Delta\psi_n|}{|\psi_n|}.$$

Theorem 2.7. *The communication cost of all subrounds of a round is $O(kV)$ words.*

Proof. Consider the n-th subround, with quantum $\theta = -\psi_{n-1}/2k$. If the counter of site i is c_i, then, at some point during this subround we had
$$\lfloor \frac{\phi(\boldsymbol{X}_i) - z_i}{\theta} \rfloor = c_i.$$
We conclude that, since $c > k$,
$$\Delta\psi_n \geq (k+1)\theta = \frac{k+1}{2k}|\psi_{n-1}| \geq \frac{|\psi_{n-1}|}{2}.$$
On the other hand, $|\psi_n - \psi_{n-1}| \leq \Delta\psi_n$, thus the variability has increased by at least $1/3$ during this subround. Therefore, the total cost of all subrounds is at most $(9k + 3)V$ words. □

At this point, we should report that in our extensive experiments with complex non-monotone functions over sketches and streams created from real data with deletions (using windows), the number of subrounds per round q was always at most 10, and almost always $7 \approx \log_2 \frac{1}{0.01}$. In fact, the total cost $O(kq)$ of all subrounds in a round, was dominated by several orders of magnitude, by the upstream cost $\Theta(kD)$. This good fortune is possibly due to tight monitoring bounds in our experiments, but still, it is an interesting observation, considering how bad the worst-case costs are.

An interesting open question is to relate ψ-variability to the variability of the query function Q, and in particular derive lower bounds based on this concept.

We should also discuss the role of ϵ_ψ. Note that this bound is unrelated to the approximation bound ε of the monitored query Q. It is simply a threshold of accuracy, with which the value of ψ is approximated. In practice, a fixed value of 0.01 seemed to suffice. The choice of ϵ_ψ can be understood as the precision to which $\phi(\boldsymbol{X}_i)$ are evaluated; their values are quantized to $\epsilon_\psi\phi(\mathbf{0})$ absolute error. Selecting a different value depends on the geometry of a particular problem; we omit the details.

2.5.2 Comparison to classic geometric monitoring. Both the FGM protocol and the standard protocol of geometric monitoring (GM) are generally applicable. The two protocols can be rendered comparable; when FGM is used together with convex safe functions, the condition $\phi(\boldsymbol{x}) \leq 0$ is akin to testing membersip in a convex Safe Zone [22]. However, the GM protocol adopts a much stricter safeness condition, equivalent to

$$\max_{i=1,\ldots,k} \phi(\boldsymbol{X}_i) < 0. \tag{6}$$

When the above condition is violated, the GM protocol performs substantial communication (either a partial, or a full synchronization, by flushing the local state vectors). By contrast, the FGM is much more patient; in fact, it is easy to see that, if the two protocols start from the same estimate $\boldsymbol{E}$ at the beginning of a round, as long as safeness condition (6) holds, the first subround of FGM has not yet finished.

Proof. As long as $\phi(\boldsymbol{X}_i) < 0$, it is $1 - \phi(\boldsymbol{X}_i)/\phi(\boldsymbol{0}) < 1$. The quantum of the first FGM subround is $\theta = -\phi(\boldsymbol{0})/2$, therefore, for each site i, it is

$$\lfloor \frac{\phi(\boldsymbol{X}_i) - \phi(\boldsymbol{0})}{-\phi(\boldsymbol{0})/2} \rfloor = \lfloor 2(1 - \frac{\phi(\boldsymbol{X}_i)}{\phi(\boldsymbol{0})}) \rfloor \leq 1,$$

and thus the coordinator has received at most k bits. □

The advantage of FGM over GM becomes more apparent by considering the size of the quiescent regions for these protocols. Each protocol will synchronize (flush local sites) as soon as the system escapes the quiescent region. It is therefore advantageous to admit a quiescent region that better approximates the set of safe configurations C.

Fig. 1 depicts the situation for $D = 1$ and $k = 2$. Without loss of generality, we are assuming $A = [-1, 1]$. The quiescent region for GM is simply $A \times A$, whereas for FGM the region depends on the choice of ϕ. Choosing $\phi(x) = |x|^p - 1$ will be correct for every $p \geq 1$, but naturally the best function is the level-maximal function $|x| - 1$ (i.e., $p = 1$). In fact, it can be seen in Fig. 1 that as

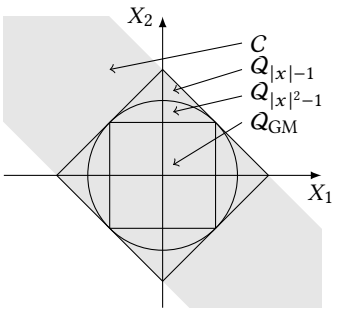

Figure 1: Configuration space for $A = [-1, 1] \subseteq \mathbb{R}$ and $k = 2$, depicting the set of safe configurations C, the FGM quiescent regions $Q_{|x|^p-1}$ for $p = 1, 2$ and the GM quiescent region Q_{GM}.

p grows, the benefit of FGM over GM decreases; however, FGM will never be inferior to GM.

3 COMPLEXITY RESULTS FOR F_p MOMENTS

We now turn to an analysis of the worst-case communication cost of FGM for F_p moments. In this analysis, it is necessary to introduce monotonicity assumptions about the studied problems; otherwise, even the simplest problems can have very bad worst case complexity. In particular, we assume that all local state vectors and drifts are frequency vectors with nonnegative coefficients; this assumption is compatible with an insert-only stream.

Similar complexity results were obtained by [9], by an algorithm which is similar to FGM; under monotonicity, each round of their algorithm is essentially a round similar to ours, but with carefully selected thresholds. Here, we will strive for a simpler approach.

Since the functions to be monitored are convex, and in fact the $F_p(\boldsymbol{x})$ moment of a frequency vector $\boldsymbol{x}$ is just the norm $\|\boldsymbol{x}\|_p^p$, we use safe functions of the form $\|\boldsymbol{x}+\boldsymbol{E}\|_p - T$. Note that selecting not to raise the norm to p yields better quiescent regions, although it does not make much difference to the asymptotic results under monotonicity.

We begin by examining the effect of a single FGM round. In such a round, we start from some global state $\boldsymbol{E}$ and we allow the protocol to proceed to termination, under an admissible region of the form $\{\boldsymbol{x} | \|\boldsymbol{x}\|_p \leq T\}$, where T depends on whether we are interested in an one-shot query or a continuous one.

Lemma 3.1. *Assume the problem of monitoring admissible region $A = \{\boldsymbol{x} | \|\boldsymbol{x}\|_p \leq T\}$ for $p \geq 1$, starting at some $\boldsymbol{E} \in A$.*

Under monotonicity assumptions, at the end of a single FGM round with safe function $\phi = \|\boldsymbol{x} + \boldsymbol{E}\|_p - T$, the final stream state $\boldsymbol{S}$ will have

$$\|\boldsymbol{S}\|_p^p \geq (1 - \frac{1}{k^{p-1}})\|\boldsymbol{E}\|_p^p + \frac{1}{k^{p-1}}\tilde{T}^p,$$

where $\tilde{T} = T(1 - \epsilon_\psi) + \epsilon_\psi \|\boldsymbol{E}\|_p \approx T$.

Proof. At the end of the round, the value of ψ has become greater than $\epsilon_\psi k \phi(0)$. Therefore, under our safe function, at the end of the round, the coordinator collected drift vectors $\boldsymbol{X}_i$, with

$$\tilde{T} \leq \frac{1}{k} \sum_{i=1}^{k} \|\boldsymbol{X}_i + \boldsymbol{E}\|_p.$$

From Hölder's inequality (thinking of the sum as an inner product with vector $\boldsymbol{1} = (1, 1, \ldots, 1)$), we get

$$\sum_{i=1}^{k} \|\boldsymbol{X}_i + \boldsymbol{E}\|_p \leq \|\boldsymbol{1}\|_q \left(\sum_{i=1}^{k} \|\boldsymbol{X}_i + \boldsymbol{E}\|_p^p \right)^{1/p},$$

where $q = p/(p-1)$ is the Hölder conjugate of p. Combining the above inequalities by raising to p, and noting that $\|\boldsymbol{1}\|_q^p = k^{p-1}$, we get

$$\sum_{i=1}^{k} \|\boldsymbol{X}_i + \boldsymbol{E}\|_p^p \geq k\tilde{T}^p. \tag{7}$$

To proceed, first observe that $k\boldsymbol{S} = k\boldsymbol{E} + \sum_{i=1}^{k} \boldsymbol{X}_i$. Consider the following real inequality (over nonnegative numbers):

$$\big(ke + \sum_{i=1}^{k} x_i\big)^p = \big(\sum_{i=1}^{k} (x_i + e)\big)^p \geq \sum_{i=1}^{k} (x_i + e)^p + (k^p - k)e^p.$$

When applied to each coordinate of $\boldsymbol{E}$ and $\boldsymbol{X}_i$, it follows from 7 that

$$k^p \|\boldsymbol{S}\|_p^p \geq k\tilde{T}^p + (k^p - k)\|\boldsymbol{E}\|_p^p.$$

Dividing both sides with k^p finishes the proof. □

Observe from the above lemma that the effect of ϵ_ψ is localized in reducing slightly the threshold T in each round; however, as $\|\boldsymbol{E}\|_p \to T$, the effect dissappears; this case makes apparent that ϵ_ψ does not affect the accuracy of the monitoring; it only increases (slightly) the number of rounds. We do not discuss ϵ_ψ any further.

We can consider two scenaria for monitoring the F_p norm.

3.0.1 One-shot query. In this scenario, which is exactly the framework of ditributed functional monitoring, we set an initial T and we estimate the number of rounds needed until $\|\boldsymbol{S}\|_p$ exceeds $(1-\varepsilon)T$, starting at $\boldsymbol{E} = \boldsymbol{0}$. By solving a simple linear recurrence, we get the following

Theorem 3.2. *For safe function $\|\boldsymbol{x}\|_p - T$, the FGM protocol can monitor the F_p moment of a monotone stream in $O(k^{p-1}\log\frac{1}{\varepsilon})$ rounds.*

For the actual communication cost, when using sketches or other summaries for F_p norms, we refer to the discussion of [9].

3.0.2 Continuous query. In this case, the threshold T given to the FGM protocol at each round is set to $(1+\varepsilon)\|\boldsymbol{E}\|_p$, that is, it changes with each round. To describe the communication cost in the continuous setting, we express the number n of rounds as a function of a starting query value Q_0 and an ending query value Q_n. Ommitting the easy details, we have

Theorem 3.3. *For safe function $\|\boldsymbol{x}\|_p - T$, the FGM protocol can monitor continuously the F_p moment of a monotone stream, as it transitions from value O_0 to Q_n in $O(\frac{k^{p-1}}{\varepsilon}\log\frac{Q_n}{Q_0})$ rounds.*

3.0.3 Discussion. The above complexity results match those of the original paper by Cormode et al. [9]. Our proofs show that they can be obtained without resorting to a special-purpose protocol like the one proposed in [9]. By contrast, the FGM protocol is built along the lines of greedy relaxation: setting a safe zone and letting the protocol iterate to completion. Naturally, there is nothing to forbid a clever coordinator algorithm to set more precise targets, in order to achieve better results.

The real limitation of FGM comes from the fact that in FGM, local nodes are memoryless; once a local drift is transmitted, the node has no memory of its past state. It should not come as a surpsise that better communication complexity can be achieved with stateful nodes. In particular, the results of [29] on very good upper bounds for frequency moments require nodes to retain much information for a long time. On the other hand, this may be quite undesirable from an implementation point of view. Thus, such results are important in the context of communication complexity with unrestricted parties, but arguably not immediately practical.

Another point compares the implementation cost of algorithms; arguably, the algorithms presented in [9] and elsewhere, are harder to adapt to more general setting. To illustrate the point, consider the problem of monitoring, say, the F_2 moment, in a stream allowing deletions as well as insertions. With FGM, it suffices to augment the safe function: starting at state $\boldsymbol{E}$, a good safe function for admissible region $A = \{\boldsymbol{x} | \|\boldsymbol{x}\|_2 \geq (1-\varepsilon)\|\boldsymbol{E}\|\}$ is defined as a half space, tangent to ball A at the projection of $\boldsymbol{E}$ onto A. Then, the safe functions for upper and lower bound of the F_2 moment can be combined via the pointwise-max operation:

$$\phi(\boldsymbol{x}) = \max\{-\varepsilon\|\boldsymbol{E}\| - \boldsymbol{x}\frac{\boldsymbol{E}}{\|\boldsymbol{E}\|}, \|\boldsymbol{x}+\boldsymbol{E}\| - (1+\varepsilon)\|\boldsymbol{E}\|\}.$$

If the above function is employed in an insertion-only setting, it will retain the cost guarantees proved above.

4 ROBUSTNESS UNDER ADVERSE CONDITIONS

We now turn our attention to features of the FGM that allow it to handle gracefully adverse streaming conditions. These conditions can arise from a number of factors, such as:

- Setting the monitoring accuracy ε to a very low value, resulting in tight thresholds for monitoring.
- In the absense of monotonicity, handling local streams which tend to cancel each other (this is a multidimensional version of the problem of non-monotone counter).
- Handling cases where the local stream rates are very uneven (e.g., following a power-law distribution).

To handle the above situations, the FGM protocol offers a number of enhancements; the effect of these enhancements is quite apparent in our experiments, but to our knowledge they do not provide asymptotic improvements in communication cost.

The guiding intuition in the following is the observation that, under a greedy view, it is preferable to have FGM rounds last longer (consuming more stream updates), since then, not only the streams are better summarized in local state vectors, but also, the upstream overhead of shipping $\boldsymbol{E}$ to local sites at the the beginning of a round is paid less often.

4.1 Rebalancing

Our starting point is the observation that $\sum_{i=1}^k \phi(\boldsymbol{X}_i) > 0$ does not generally imply that $\phi(\frac{1}{k}\sum_{i=1}^k \boldsymbol{X}_i)$ is also positive, or even much different than $\phi(\boldsymbol{0})$, i.e., often, at the end of a round, the global stream state $\boldsymbol{S}$ has not moved significantly far from $\boldsymbol{E}$. Therefore, the current safe function ϕ may still be quite useful, and we would like to avoid the overhead of shipping a new safe function to the sites.

Rebalancing is an important technique in classic Geometric Monitoring. The idea in GM is to flush a subset of the local sites, and then ship them the average of their previous drifts. A straightforward adaptation of the rebalancing method of GM, could benefit FGM. Unfortunately, the method is highly uncertain as to the benefit it provides, versus the added upstream communication overhead (which is a multiple of $O(D)$).

A simple approach to rebalancing, that incurs negligibly small additional upstream communication cost, is to ship to the sites a scaling factor, with which to scale their local drifts. We restrict the discussion to convex safe functions.

In our rebalancing scheme, the coordinator holds an extra state vector, the *balance vector* $\boldsymbol{B}$, which is used to aggregate drift vectors from local sites, without ending the round. At the beginning of a round, the balance vector is set to $\boldsymbol{0}$. During the round, sites update their drift vectors as local stream updates arrive. However, with rebalancing allowed, it is possible for a site to flush its current drift vector to the coordinator, during the round. When a flush occurs, the coordinator updates the balance, by adding $\boldsymbol{X}_i$ to it. After drift vector $\boldsymbol{X}_i$ is flushed, it is reset to $\boldsymbol{0}$.

Therefore, the global drift is always equal to

$$\boldsymbol{B}/k + \frac{1}{k}\sum_{i=1}^k \boldsymbol{X}_i. \tag{8}$$

This can be rewritten as

$$\mu \frac{\boldsymbol{B}}{\mu k} + \frac{\lambda}{k} \sum_{i=1}^{k} \frac{\boldsymbol{X}_i}{\lambda} \tag{9}$$

for some $\lambda > 0$, $\mu \geq 0$, with $\lambda + \mu = 1$ (note that we allow $\mu = 0$ when $\boldsymbol{B} = \boldsymbol{0}$, namely at the beginning of a round).

If ϕ is a convex $(A, \boldsymbol{E})$-safe function, known to the sites, we can adapt the safety condition by applying ϕ to Eq. 9. Define

$$\psi = \sum_{i=0}^{k} \lambda \phi(\frac{\boldsymbol{X}_i}{\lambda}) \qquad \text{and} \qquad \psi_B = \begin{cases} (1-\lambda)k\phi(\frac{B}{(1-\lambda)k}) & \lambda < 1, \\ 0 & \lambda = 1. \end{cases}$$

Theorem 4.1. *If ϕ is convex $(A, \boldsymbol{E})$-safe, then, for any $\lambda \in (0, 1]$,*

$$\psi + \psi_B \leq 0 \implies \boldsymbol{E} + \frac{\boldsymbol{B} + \sum_{i=1}^{k} \boldsymbol{X}_i}{k} \in A.$$

Proof. Let $\mu = 1 - \lambda$. By convexity,

$$k\phi(\frac{\boldsymbol{B} + \sum_{i=1}^{k} \boldsymbol{X}_i}{k}) \leq \mu k \phi(\frac{\boldsymbol{B}}{\mu k}) + \lambda k \phi(\frac{\sum_{i=1}^{k} \boldsymbol{X}_i}{\lambda k}) \leq \psi_B + \psi.$$

Since $k\phi$ is $(A, \boldsymbol{E})$-safe, and dominated by $\psi + \psi_B$, the claim follows. □

To monitor condition $\psi + \psi_B \leq 0$, the only modification needed to the FGM algorithm during subrounds, is in the selection of a suitable quantum θ at the beginning of each subround, so that

$$2k\theta = -(\psi + \psi_B). \tag{10}$$

4.1.1 Rebalancing FGM protocol. The extended protocol begins exactly as described in §2.4, with $\lambda = 1$. At the end of all subrounds, it is $\psi > \epsilon_\psi k\phi(\boldsymbol{0})$. Where the basic protocol would start a new round, the rebalancing protocol restores the invariant $\psi + \psi_B \leq 0$ as follows:

(1) The coordinator asks some or all of the sites to flush their local drift vectors, and updates $\boldsymbol{B}$. There are many possible heuristics that can be employed to do this as conservatively as possible, dealying flushes and thus giving the opportunity to local streams to summarize their results better.
(2) When all drift vectors have been received, the coordinator recomputes ψ_B and ψ, choosing a new value for λ, or failing. The choice of λ is discussed below.
(3) If condition $\psi + \psi_B \leq \epsilon_\psi k\phi(\boldsymbol{0})$ is restored, a new subround is started with quantum $\theta = -(\psi + \psi_B)/(2k)$,
(4) else, the round finishes and a new round starts by computing the new $\boldsymbol{E}$ and shipping it to all sites.

4.1.2 Selection of λ. The choice of a good λ is a generally dependent on the statistics of the monitored streams. Consider the "ideal" case, where $\boldsymbol{B}$ was shipped back to the sites; then, the sites could instead monitor function $\phi(\boldsymbol{x} + \boldsymbol{B}/k)$ (we would have $\psi_B = 0$). This is "ideal" in the sense that for any $\lambda > 0$,

$$\sum_{i=1}^{k} \phi(\boldsymbol{X}_i + \boldsymbol{B}/k) \leq \sum_{i=1}^{k} \lambda\phi(\boldsymbol{X}_i/\lambda) + \mu k \phi(\boldsymbol{B}/(\mu k)) = \psi + \psi_B. \tag{11}$$

Scaling the input streams. Let $Z = L(\phi)$. Geometrically, the level set of $\phi(\boldsymbol{x} + \boldsymbol{B}/k)$ is $Z - \boldsymbol{B}/k$, that is, it is a shift of Z along the $\boldsymbol{B}$-direction. The new safe zone, by our choice of λ has to be a subset of this set. We could then "scale down" Z to λZ, so that $\lambda Z \subseteq Z - \boldsymbol{B}/k$, and their boundaries touch at a point along the axis of the shift, that is, at $\boldsymbol{B}/(\mu^* k)$, where

$$\mu^* = \inf\{\mu > 0 \mid \phi(\boldsymbol{B}/(\mu k)) = 0\}.$$

This value of μ^* can easily be found iteratively by bisection. Then, $\lambda = 1 - \mu^*$. This heuristic is well-behaved in practice and is the one we have used in our experiments.

4.1.3 Discussion. To assess the effect of rebalancing on round duration, assume that the statistics of the *global stream* are such that the global state vector $\boldsymbol{S}$ maintains a rougly constant "velocity" over the stream data. Under this "statistical inertia" assumption, which is often a realistic approximation of stream statistics, our rebalancing protocol achieves a round duration at least 1/2 of the ideal maximum: if τ stream updates were processed, then processing another τ updates would lead the total drift outside the safety bounds (i.e., outside $L(\phi)$). In such conditions, rebalancing ameliorates the presence of *skew* in the trends and rates of *local* streams.

4.2 Adaptively shipping safe zones to local sites

In order to amortize the upstream cost of a round with communication benefits, it is necessary for a round to last for at least twice this many updates totally; that is, the round must last for at least $2kD$ updates, if the total communication cost of the round is to be better than the naive method. This minimum duration of a round (in terms of local stream updates) may not be not be achievable when overall variability is high.

Another practical issue, even with low variability, is the case where the stream rates of individual sites are highly unequal, e.g., they follow a 98-2 power law. Then, the cost of shipping safe zones to 98% of the sites is probably wasteful; those sites could just forward their local streams to the coordinator, and a protocol should try to save communication on those 2% of the sites which provide 98% of the stream updates.

Many previous protocols for distributed monitoring, including much of the previous work on Geometric Monitoring, do not adapt well to such problematic situations. In this section, we introduce an enhancement the FGM protocol, where such situations are handled within the protocol's basic logic. There are two subproblems addressed by our solution; (a) a systematic way to eliminate the upstream cost of shipping $\boldsymbol{E}$ to selected sites at the beginning of a round, and (b) a cost-based way to select those sites.

4.2.1 Reducing the upstream costs. One simplistic way to avoid shipping $\boldsymbol{E}$ to a site at the beginning of a round, is to put this site into "promiscuous mode", that is, to let it ship all local stream updates to the coordinator, which can then "simulate" the local node, and otherwise execute the protocol as is.

Naturally, this simplistic method will create many small messages, which we would like to avoid. This can be done if we ship to the site a cheaper safe function, such as some function of the form $b(\boldsymbol{x}) = \|\boldsymbol{x}\|_p^q + a$, which takes only 3 words (carrying p, q, a) to tramsmit. To maintain correctness, it is sufficient to guarantee simply that $\phi \leq b$. Given such a function, a site can participate normally in the FGM protocol. Naturally, the fact that this site is not equipped with the full function ϕ may cause it to end subrounds prematurely (sending many bits rapidly) and thus interfering with other sites. Although this is certainly possible, our experiments revealed that, under adverse monitoring conditions, the coordinator will often decide to ship the cheap safe function to *every* site, in which case the interference problem vanishes.

Selecting a function $b \geq \phi$ depends on the analytic properties of ϕ, and can in general be done easily. In general, we should avoid higher-degree functions, as they grow too quickly; this is possible if the degree of ϕ itself is small (which, is important for achieving better quiescent region for ϕ, as discussed previously). In order to keep our exposition simple, we do not discuss this issue in full analytic generality. Instead, we note that a 1-degree requirement can be met, if the safe zone function ϕ is nonexpansive:

$$\forall \boldsymbol{x}, \boldsymbol{y} \in V,\ |\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|. \tag{12}$$

This property is well-known in functional analysis, and is also known as *Lipschitz continuity*. It is easy to see that, in this case,

$$|\phi(\boldsymbol{x}) - \phi(\boldsymbol{0})| \leq \|\boldsymbol{x}\|,$$

which implies that

$$\phi(\boldsymbol{x}) \leq \|\boldsymbol{x}\| + \phi(0).$$

An important class of safe functions that are non-expansive are the Signed Distance Functions of convex sets. Also, the gauge functions with bounded level-set (including all norms and semi-norms) can be scaled to be nonexpansive.

Selecting the sites which will use the "cheap" function b is crucial. We propose a solution based on a cost model and some collected statistics, much in the spirit of database query optimization. In the rest of this section, we will ignore the FGM rebalancing protocol, and instead focus on the basic FGM protocol. This is done for the sake of keeping our optimization algorithm simple. However, once the "plan" for a round is selected, the full FGM protocol with rebalancing can be executed for the round.

4.2.2 Modeling the communication cost of a round. Assume that the coordinator is at the beginning of a round, with current estimate $\boldsymbol{E}$ and has selected a non-expansive safe function ϕ. Let $d_i, i = 1, \ldots, k$ be indicator variables; d_i is equal to 1 when the full safe function ϕ is to be shipped to local site i, and 0 if the cheap safe function is to be used. In other words, d_i encodes the optimized "query plan" for the upcoming round. Let $\boldsymbol{d}$ denote the vector of d_i values. Our goal is to select the plan $\boldsymbol{d}$ that maximizes the gain of the round.

Assume that, based on the decision $\boldsymbol{d}$, the length of the next round is going to be τ. Furthermore, assume that a fraction $\gamma_i \tau$ of these updates arrives at local stream i. The benefit of the round in terms of summarizing τ updates in the local state vectors, is

$$g_0 = \tau - \sum_{i=1}^{k} \min\{\gamma_i \tau, D\}, \tag{13}$$

where $\min\{\gamma_i \tau, D\}$ reflects the downstream cost of site i, which will ship $\gamma_i \tau$ raw updates, instead of the D-dimensional drift vector, if $\gamma_i \tau < D$. In addition, the upstream cost of the round is $D \sum_{i=1}^{k} d_i$ (where we assume that the difference in the cost of shipping ϕ vs. b is D). Therefore, we must select $\boldsymbol{d}$ so as maximize the round's *gain*,

$$g = \tau - \sum_{i=1}^{k} \min\{\gamma_i \tau, D\} - D \sum_{i=1}^{k} d_i. \tag{14}$$

The challenge is to predict $\tau(\boldsymbol{d})$, given a choice for $\boldsymbol{d}$. To this end, consider ψ as function of "time" (updates):

$$\psi(t) = \sum_{d_i=1} \phi(\boldsymbol{X_i}(t)) + \sum_{d_i=0} \|\boldsymbol{X}_i(t)\| - \phi(\boldsymbol{0}) \tag{15}$$

The current round can be seen as the transition of the system from a state where $\psi = k\phi(\boldsymbol{0})$ to a state where $\psi = 0$. Of course, this transition will in general follow a complicated, non-linear trajectory in the quiescent region. However, we adopt a simplistic linear estimate. In particular, we model the behaviour of each local stream i by two rates, α_i and β_i, assuming simplistically that

$$\phi(\boldsymbol{X}_i(t)) \approx \phi(\boldsymbol{0}) + |\phi(\boldsymbol{0})|\alpha_i t \tag{16}$$

$$\|\boldsymbol{X}_i(t)\| + \phi(\boldsymbol{0}) \approx \phi(\boldsymbol{0}) + |\phi(\boldsymbol{0})|\beta_i t \tag{17}$$

We shall assume that $0 < \alpha_i < \beta_i$.

Based on this simple-minded model, the prediction of a round's length τ, as a function of $\boldsymbol{d}$, based on Eq. 15, is

$$\tau = \frac{k}{\beta_{\text{tot}} - \boldsymbol{d} \cdot \boldsymbol{\theta}}, \tag{18}$$

where $\beta_{\text{tot}} = \sum_{i=1}^{k} \beta_i$ and $\boldsymbol{\theta}$ is the vector of values $\theta_i = \beta_i - \alpha_i$.

4.2.3 Maximizing the gain of a round. It is required to find the value of $\boldsymbol{d}$ that maximizes the gain g (Eq. 14). An exhaustive search of the solution space would require time $O(2^k)$, which would not scale well to large k. Thankfully, it turns out that a simple greedy algorithm is sufficient to maximize g. The key observation is that $g_0(\tau)$ (from Eq. 13) is non-decreasing in τ. Fix some number $0 \leq n \leq k$. We wish to find a feasible solution $\boldsymbol{d}^*$, with $\sum_i d_i^* = n$, which maximizes g among all solutions $\boldsymbol{d'}$ with $\sum_i d_i' = n$. But since $g(\boldsymbol{d}^*) = g_0(\tau(\boldsymbol{d}^*)) - nD$, and g_0 is non-decreasing in τ, it suffices to maximize $\tau(\boldsymbol{d}^*)$. To do this, simply set $d_i^* = 1$, iff θ_i is among the n largest coordinates of $\boldsymbol{\theta}$ (ties are broken arbitrarily). Now, g can be optimized by comparing among $k+1$ solutions, one for each value of n. Furthermore, since an optimal solution for $n+1$ subsumes an optimal solution for n, the whole computation can be performed in $O(k \log k)$ steps (essentially for sorting vector $\boldsymbol{\theta}$).

4.2.4 Obtaining estimates for local streams. It remains to discuss the estimation of α_i, β_i and γ_i in each round. In this paper, we explored the simplest possible alternative: simply use the data collected at the end of the previous round, to obtain fresh estimates of all three parameters. Since at the end of a round the coordinator has received each drift $\boldsymbol{X}_i$, together with a count of updates to each local stream during the round, all three parameters can be computed directly, more or less from Eqs. 16 and 17.

Some care must be taken, to ensure $0 < \alpha_i < \beta_i$; in particular, Eq. 16 may yield a non-positive value. If this occurs, then simply set α_i to a small positive value (so that θ_i is minimum among the components of vector $\boldsymbol{\theta}$. Also, when $\beta_i = 0$ or $\gamma_i = 0$ (there were no updates to the site in the previous round), simply set $d_i = 0$ and ignore this site in the optimization process.

4.2.5 Discussion. Our estimates for modeling local streams are simple to acquire in practice, but may yield estimates which may not represent well the evolution of the system. After all, predictions are hard, especially about the future! Thankfully, our approach, of estimating τ in order to decide on the next round's plan, is relatively insensitive to the exact value of τ, as it is essentially a based of thresholds, determined by local stream rate predictions, which can be predicted much more accurately.

In practice the algorithm managed to perform quite well, making the FGM protocol quite robust in adverse situations of very high variability, compared to executions that shipped a safe function to every site. Most of the time, the selection of $\boldsymbol{d}$ values either resulted in almost all 1s (when variability was low) or in almost all 0s (during high variability). Also, during periods of

medium variability, the algorithm would alternate between these two decisions for a few rounds.

Improving on this algorithm is certainly an interesting problem. On the prediction side, higher-order polynomial models in place of Eqs. (16–17) can in principle be constructed. Whether these more elaborate modeling would benefit the final communication cost in real data settings, remains to be seen.

5 EXPERIMENTAL EVALUATION

We performed an extensive experimental study of the FGM protocol, over a variety of datasets and streaming parameters, with emphasis on validating our claims of resilience to adverse situiations. For lack of space, we only present results from the WorldCup dataset [2], which contains log traces of all requests sent to the 1998 World Cup web site, consisting of 33 mirrors spread around the globe and receiving 1.3 billion http requests. Our experiments used only data from day 46, during which 50.3 million requests where received by 27 mirror sites. From this data, we constructed stream records over the schema R(CID, TYPE), where CID is the (anonymized) client address of the http request, and TYPE is the type of file requested (HTML, image etc).

On this stream, we approximately monitored two continuous queries. Both queries operate on Fast-AGMS sketches [8] on the input streams. A Fast-AGMS sketch $\boldsymbol{S}$ is stored as a $d \times w$ matrix $\boldsymbol{S}$ of integer counters, and can be used to estimate join and self-join sizes within accuracy $\Theta(1/\sqrt{w})$ with probability at least $1-2^{-\Theta(d)}$. Each stream update changes the sketch by modifying one cell in each row vector $\boldsymbol{S}[i]$ (totally, d cells) by ± 1, according to certain hash functions.

The first query monitors the self-join size of $R \bowtie_{\text{CID}} R$. To estimate this query, an AGMS sketch is used as the state vector to summarize all records. The query function is the self-join size estimate,

$$Q_1(S) = \underset{i=1,\ldots,d}{\text{median}}\{\sum_{j=1}^{w} S[i,j]^2\} = \underset{i=1,\ldots,d}{\text{median}}\{S[i]^2\}.$$

The second query monitors the join size of

$$\sigma_{\text{TYPE=HTML}}(R) \bowtie_{\text{CID}} \sigma_{\text{TYPE}\neq\text{HTML}}(R).$$

For this query, the state vector consisted of the concatenation of two sketches, $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$. The monitored query function is

$$Q_2(S_1S_2) = \underset{i=1,\ldots,d}{\text{median}}\{\sum_{j=1}^{w} S_1[i,j]S_2[i,j]\} = \underset{i=1,\ldots,d}{\text{median}}\{S_1[i]S_2[i]\}.$$

Note that query function Q_2 is much more challenging than query Q_1 in terms of variability.

5.1 Experimental setup

We explored the space of four parameters: AGMS sketch size, size of sliding window over the streams, monitoring accuracy ε and the number of sites k.

We evaluated queries Q_1 and Q_2 both in the cash-register model (each record was inserted one at a time), and also in the turnstile model, where we used a time-based sliding window (ranging from 1hr to 4hrs) to generate record deletions. Naturally, the variability of our queries decreases as the time window increases. Also, time-based windows yield higher variability than fixed-size ones. We allowed the monitoring accuracy to vary as $\varepsilon \in [0.02, 0.1]$.

Finally, in order to study the effect of k (the number of sites) on performance, we created synthetic streams by hashing the original 27 local site ids to fewer site ids, for $k \in [2:20]$. Naturally, we also used the original (real) data, for $k = 27$.

Note that, in all experiments presented, the "global" stream was identical, and we simply changed the distribution of the data in time (by sliding windows), and among local streams.

5.1.1 Safe functions employed. We implemented nonexpansive, convex safe functions for queries Q_1 and Q_2 following the technique of [13]. To monitor query Q_1 for estimate sketch $\boldsymbol{E}$ we need to ensure that $(1-\varepsilon)|Q_1(\boldsymbol{E})| \leq Q_1(\boldsymbol{S}) \leq (1+\varepsilon)|Q_1(\boldsymbol{E})|$.

We can rewrite it compactly (applying properties of the median) as

$$\pm\big(Q_1(\boldsymbol{S}) - T^{\pm}\big) = \underset{i=1,\ldots,d}{\text{median}}\{\pm(\boldsymbol{S}[i]^2 - T^{\pm})\} \leq 0.$$

where, for $\pm \in \{+,-\}$, $T^{\pm} = (1 \pm \varepsilon)|Q_1(\boldsymbol{E})|$, respectively.

The safe function $\phi(\boldsymbol{X})$ we used is *composed* as

$$\phi(\boldsymbol{X}) = \max\big(\phi^-(\boldsymbol{X}), \phi^+(\boldsymbol{X})\big),$$

where $\phi^{\pm}$ is safe for condition $\pm\big(Q_1(\boldsymbol{S}) - T^{\pm}\big) \leq 0$ respectively. Following the methodology of [13] for the median, we used

$$\phi^{\pm}(\boldsymbol{X}) = \max_{I \in \binom{D^{\pm}}{|D^{\pm}|-(d-1)/2)}} \frac{\sum_{i\in I} |\phi_i^{\pm}(\boldsymbol{0})| \cdot \phi_i^{\pm}(\boldsymbol{X}[i])}{\sqrt{\sum_{i\in I} |\phi_i^{\pm}(\boldsymbol{0})|^2}},$$

where $D^{\pm} = \{i \mid 1 \leq i \leq d \text{ and } \pm(\boldsymbol{E}[i]^2 - T^{\pm}) < 0\}$. Note that the notation $I \in \binom{D}{n}$ means "I ranges over all n-subsets of D".

Functions $\phi_i^{\pm}(\boldsymbol{x}), i = 1,\ldots,d$, must be safe for conditions $\pm(\boldsymbol{S}[i]^2 - T^{\pm}) \leq 0$ respectively. We used

$$\phi_i^+(\boldsymbol{x}) = \|\boldsymbol{x}+\boldsymbol{E}[i]\| - \sqrt{T^+} \quad \text{and} \quad \phi_i^-(\boldsymbol{x}) = \sqrt{T^-} - \frac{\boldsymbol{E}[i]}{\|\boldsymbol{E}[i]\|}(\boldsymbol{E}[i]+\boldsymbol{x}).$$

The same methodology was applied to derive the safe functions for the Q_2 query; the derivation is very similar to the above, however, the actual formulas for $\phi_i^{\pm}$ for conditions $\pm(\boldsymbol{S_1}[i]\boldsymbol{S_2}[i] - T^{\pm}) \leq 0$ are a bit involved and are omitted due to space constraints; we refer the reader to [13] (Section 6.3) for details, as well as for the justification of the above steps.

5.1.2 Tested protocols. In order to compare the performance of the FGM protocol to previous work, we implemented a well-studied version of the GM protocol, based on Safe Zones [22], with a rebalancing policy along the lines of [28]. The Safe Zones used where defined using the safe functions of the FGM described above, so as to fairly contrast the inherent communication costs of the GM and FGM protocols.

To study the effect of our cost-based optimizer, we ran versions of FGM with and without it. Overall, the acronyms of the 3 protocols tested are as follows:

Acronym	Protocol
GM	classic GM protocol with rebalancing.
FGM	FGM protocol without cost-based optimizer.
FGM/O	FGM protocol with cost-based optimizer.

5.2 Performance in typical workloads

Our first set of experiments concerns the behaviour of the protocols in a non-adverse scenario (using a 4ht window over the data), monitoring accuracy $\varepsilon = 0.1$, and sketch sizes, $D = 7000$. The results depicted in Figs. 2 and 3 (corresponding to semi-join and join queries) depict this cost as a function of k, both in the turnstile and in the cash-register model.

With respect to communication cost, observe that, as k grows, the FGM protocols exhibit 2–3 times lower communication cost

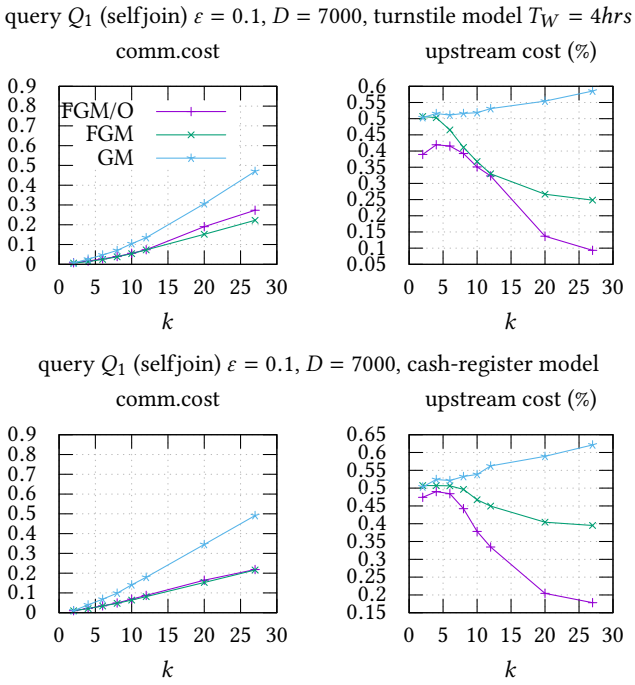

Figure 2: Performance of the GM and FGM protocols, monitoring a self-join query, over k. The top row shows the cost in the tunrstile model (with a window over the streams) and the bottom row show the cost in the cash-register model.

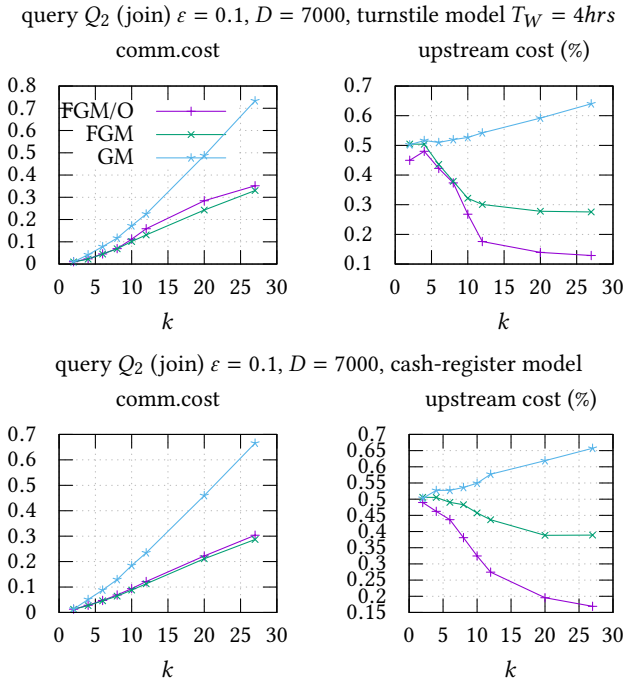

Figure 3: Performance of the GM and FGM protocols, monitoring a join query, over k. The top row shows the cost in the tunrstile model (with a window over the streams) and the bottom row show the cost in the cash-register model.

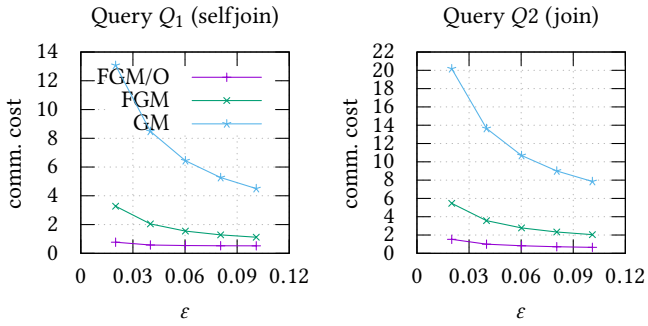

Figure 4: Communication cost for queries Q_1, Q_2, under a difficult workload. Observe that, except for FGM/*O, all protocols exhibit much higher communication cost that the size of the streamed data. Here, $k = 27$, $D = 35000$, $T_W = 1hr$.

than the GM protocols. On the other hand, for small values of k, the difference is not as pronounced.

The graphs on the right side, depicting upsteam communication costs as a percentage of total communication cost, reveals the cause of this behaviour. It is shown that the upstream cost of the standard geometric method grows as a percent of total, as the number of sites increases. This is due to two causes: first, as more sites partake in the monitoring, the strictness of the GM's monitoring condition causes frequent violations of the safety invariant of GM, while most of these violations are false positives. The rebalancing strategy of GM algorithms is unable to overcome this increase (note that, without rebalancing, the GM algorithm's total cost increases even faster, as each false violation would cause a full synchronization).

By contrast, the upstream cost of FGM *decreases* (as a percent of total communication). This is the case both with and without the cost-based optimizer. When k is small, upstream and downstream costs are roughly similar, which is true for the GM as well. As k increases however, the total cost (which increases with k naturally) is dominated by the downstream cost, of shipping data to the coordinator. This is both due to the improved safety condition of FGM, but also to the ovehead-free rebalancing performed.

Note finally the effect of cost-based optimization, which tries to aggressively minimize the upstream cost, even at the expense of downstream cost. Although total cost does not change much, the upstream percentage reduces much further. This is because the cost-based optimizer will decide not to ship safe functions to the sites in many rounds. This choice worsens the quality of summarization at the local nodes, increasing downstream costs, but manages to keep upstream costs low, while achieving good total cost.

5.3 Performance in adverse conditions

We now evaluate the performance of FGM and GM protocols under an adverse scenario, on the real WorldCup dataset, where $k = 27$. We have a large $D = 35,000$, and the stream's window is 1hr, leading to high variability. Fig. 4.

Under these conditions, round lengths are too short to amortize the cost of shipping safe zones to the sites. Therefore, all methods except for FGM/O incur excessive communication costs, in fact several times over the size of the streamed data. This

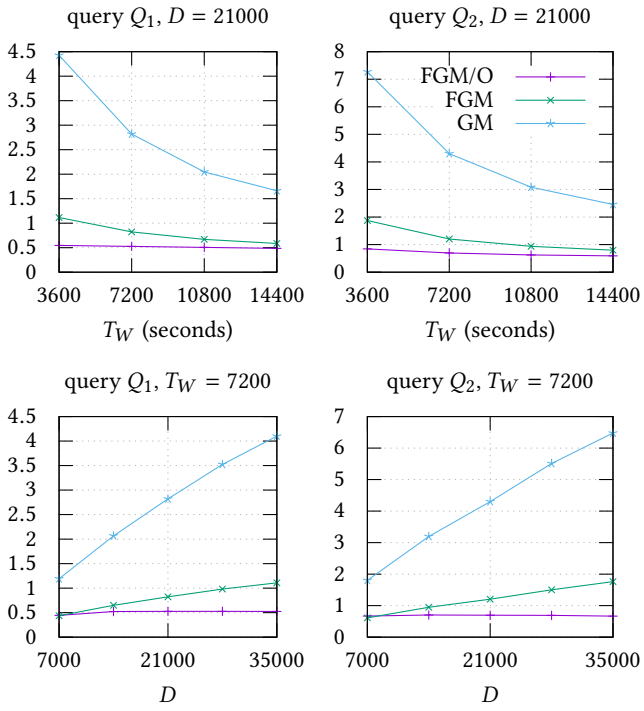

Figure 5: Communication cost for queries Q_1 (left column) and Q_2 (right column), over varying sliding windows (T_w, top row) and sketch size (D, bottom row). In all cases, it was $k = 27$ and $\varepsilon = 0.06$.

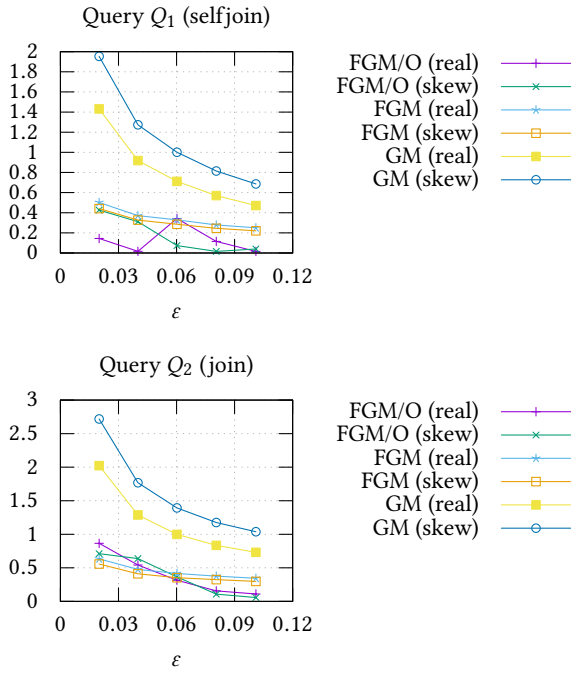

Figure 6: Communication cost for queries Q_1 (top) and Q_2 (bottom), over varying accuracy ε. For each protocol, two curves are shown, one for the real dataset and one for the skewed dataset. For both plots: $k = 27$, $D = 7000$, turnstile model (T_W=4hrs)

is not unexpected; consider that, shipping safe zones to all 27 sites, transmits roughly 3.8 Mbytes of data. Combined with short rounds due to high variability and low values of ε results in excessive overhead.

By contrast, the cost-based optimizer, although it did not deliver significant gains compared to the size of the streamed data, managed to keep the total cost quite low. This was achieved by selecting to avoid the overhead of shipping safe functions in most of the rounds.

5.3.1 Dependence on size of state vectors and on variability. The effect of variability, which decreases as the time window sliding over the stream becomes wider, is quite strong on performance. The top row of plots in Fig. 5 demonstrates this for turnstile queries, where the time window T_W changed from 1 hour to 4 hours. In particular, for the Q_2 function, using the cost model improved performance by two times over FGM (and by 4 over the GM methods), when T_W=1hr.

Similarly strong is the effect of D on performance, as depicted in the bottom row of plots of Fig. 5. In fact, the cost grows linearly with D, except for the case of FGM/O, where the cost-based optimizer switched to the cheap safe functions, achieving a small amount of compression.

5.4 The effect of skew

In order to evaluate the behaviour of our protocols under the presense of skew, we contrast the change in communication when the (real) dataset becomes more skewed. To introduce skew, we constructed a new dataset as follows: we selected 8 sites (out of a total of 27), namely those with local streams of greatest size. Then, we replaced the local stream of one of these sites—which will be referred to as the *hot site*—by the union of all 8 local streams, while the 7 remaining sites received empty local streams. In this *skewed* dataset, one local stream now provides almost half the data to the system, while 7 out of 27 local streams provide no data. However, at each point in time, the *global stream* of the skewed dataset is identical to the global stream of the real dataset.

Fig. 6 depicts the effect of skew on the communication cost. Each protocol was run with both the real and the skewed dataset. Unsuprisingly, the GM protocol's communication cost increases as skew is introduced; this is a well-known weakness of the classic geometric method. The source of the increased cost is a substantial increase in the upstream cost, because of frequent local violations at the hot site.

The FGM prococol without the cost-based optimizer on the other hand, shows resilience in the presence of skew; in fact, the communication cost improves slightly under skew. The key reason is that the ψ-value of the system under the real dataset, is always equal to the ψ-value of the system under the skewed dataset. Therefore, the coordinators in the two systems will perform the exact same number of rounds. The slight improvement is due to a reduction in the downstream cost among the 8 sites; the downstream cost of the hot site has not increased substantially (since the number of rounds remains the same), but the downstream costs of the 7 sites whose local stream vanished has decreased to almost 0 (since these sites will not ship local vectors to the coordinator).

The introduction of the cost-based optimizer is again largely beneficial to the performance. In previous experiments under adverse conditions (e.g., Fig. 4), the benefit of the optimizer was in keeping the upstream cost from becoming too large. In this scenario where skew is introduced, the benefit of the cost-based optimizer materializes more consistently when $\varepsilon \geq 0.05$, where significant benefit to the upstream cost of a round accrues, since the coordinator will undoubtedly choose the cheap safe functions for the 7 sites with empty local streams. Note that, in this scenario, the ψ-values of the constrasted systems are no longer equal (since different optimizer choices affect the actual ψ).

In this experiment, one can also observe the somewhat erratic effect of the cost-based optimizer, due to the crudeness of modeling local stream behaviour (interestingly, in the presense of skew, the behaviour is less erratic). The erratic behaviour is observed in the transition between the two extremes of small and large values of ε; for values of ε around 0.05, it seems that the cost-based optimizer will often be fooled into making sub-optimal choices. However, this is preferable to not using it at all.

Overall, our experimental results demonstrate that the FGM protocol manages to ameliorate the shortcomings of classic GM protocols, both under adverse conditions as well as in the presense of skew in the distributed stream.

6 CONCLUSIONS AND IMPLICATIONS FOR PRACTICE

We have proposed Functional Geometric Monitoring, a novel method for distributed stream monitoring, which offers significant improvements over previous techniques in terms of performance, scalability and robustness. FGM is generally applicable, it can provide worst-case guarantees for problems that were hitherto provided only by problem-specific algorithms, and it is robust in high variability and skew situations, curing an important shortcoming of previous general techniques.

Real-world stream-processing engines are typically customized by providing data-handling code (e.g., mapper/reducer functions in Hadoop, spouts and bolts in STORM, etc), which is independent of distributed execution concerns. The engine orchestrates the distributed execution of this code on a distributed platform, applying complex execution policies (resource allocation, load balancing, networking patterns, failure tolerance, etc).

The salient practical feature of FGM is that it fits this pattern extremely well, as it strictly encapsulates the specifics of monitored queries into data-handling code, namely, routines and data structures—such as sketches—to summarize local streams, and safe function implementations on these summaries. This code is platform-agnostic and an FGM implementation can deploy it on a distributed platform and execute it in a black-box fashion, under any desired execution policy.

Other aspects of FGM are alse important in practice. Although high-quality safe functions for complex query operators can be hard to derive, safe function composition can ease the burden many cases. Furthermore, the FGM protocol is resilient to loss of precision due to computational round-off errors. In addition, since local nodes are memoryless from one round to the next, the FGM protocol is compatible with relatively simple and cheap failure recovery policies.

Acknowledgement. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825070.

REFERENCES

[1] C. Arackaparambil, J. Brody, and A. Chakrabarti. Functional monitoring without monotonicity. In *ICALP (1)*, 2009.
[2] M. Arlitt and T. Jin. A workload characterization study of the 1998 world cup web site. *Netwrk. Mag. of Global Internetwkg.*, 14(3):30–37, May 2000.
[3] B. Babcock and C. Olston. Distributed top-k monitoring. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2003. ACM.
[4] S. Burdakis and A. Deligiannakis. Detecting outliers in sensor networks using the geometric approach. In *ICDE*, 2012.
[5] G. Cormode and M. Garofalakis. "Sketching Streams Through the Net: Distributed Approximate Query Tracking". In *Proc. of the 31st Intl. Conference on Very Large Data Bases*, Trondheim, Norway, Sept. 2005.
[6] G. Cormode and M. Garofalakis. "Approximate Continuous Querying over Distributed Streams". *ACM Transactions on Database Systems*, 33(2), June 2008.
[7] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine. "Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches". *Foundations and Trends in Databases*, 4(1-3), 2012.
[8] G. Cormode and M. N. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In *VLDB*, 2005.
[9] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed functional monitoring. In *SODA*, 2008.
[10] D. Felber and R. Ostrovsky. Variability in data streams. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '16, pages 251–260, New York, NY, USA, 2016. ACM.
[11] M. Gabel, D. Keren, and A. Schuster. Anarchists, unite: Practical entropy approximation for distributed streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 837–846, New York, NY, USA, 2017. ACM.
[12] M. Garofalakis, D. Keren, and V. Samoladas. "Sketch-based Geometric Monitoring of Distributed Stream Queries". In *Proc. of the 39th Intl. Conference on Very Large Data Bases*, Trento, Italy, Aug. 2013.
[13] M. N. Garofalakis and V. Samoladas. Distributed query monitoring through convex analysis: Towards composable safe zones. In *20th International Conference on Database Theory, ICDT 2017, March 21-24, 2017, Venice, Italy*, pages 14:1–14:18, 2017.
[14] N. Giatrakos, A. Deligiannakis, M. N. Garofalakis, I. Sharfman, and A. Schuster. Prediction-based geometric monitoring over distributed data streams. In *SIGMOD*, 2012.
[15] R. Gupta, K. Ramamritham, and M. K. Mohania. "Ratio threshold queries over distributed data sources". In *Proc. of the 39th Intl. Conference on Very Large Data Bases*, Trento, Italy, Aug. 2013.
[16] S. R. Kashyap, J. Ramamirtham, R. Rastogi, and P. Shukla. Efficient constraint monitoring using adaptive thresholds. In *ICDE*, pages 526–535, 2008.
[17] R. Keralapura, G. Cormode, and J. Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *SIGMOD*, 2006.
[18] D. Keren, G. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman, and A. Deligiannakis. "Geometric Monitoring of Heterogeneous Streams". *IEEE Transactions on Knowledge and Data Engineering*, 26(8), Aug. 2014.
[19] D. Keren, I. Sharfman, A. Schuster, and A. Livne. Shape sensitive geometric monitoring. *IEEE Trans. Knowl. Data Eng.*, 24(8), 2012.
[20] A. Lazerson, M. Gabel, D. Keren, and A. Schuster. One for all and all for one: Simultaneous approximation of multiple functions over distributed streams. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, DEBS '17, pages 203–214, New York, NY, USA, 2017. ACM.
[21] A. Lazerson, D. Keren, and A. Schuster. Lightweight monitoring of distributed streams. In *Proc. of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1685–1694, New York, NY, USA, 2016. ACM.
[22] A. Lazerson, I. Sharfman, D. Keren, A. Schuster, M. Garofalakis, and V. Samoladas. "Monitoring Distributed Streams using Convex Decompositions". In *Proc. of the 41st Intl. Conference on Very Large Data Bases*, Aug. 2015.
[23] S. Meng, T. Wang, and L. Liu. Monitoring continuous state violation in datacenters: Exploring the time dimension. In *ICDE*, pages 968–979, 2010.
[24] S. Michel, P. Triantafillou, and G. Weikum. Klee: a framework for distributed top-k query algorithms. In *VLDB '05*. VLDB Endowment, 2005.
[25] O. Papapetrou and M. Garofalakis. "Continuous Fragmented Skylines over Distributed Streams". In *Proc. of the 30th Intl. Conference on Data Engineering*, Chicago, Illinois, Apr. 2014.
[26] S. Shah and K. Ramamritham. Handling non-linear polynomial queries over dynamic data. In *ICDE*, 2008.
[27] I. Sharfman, A. Schuster, and D. Keren. "A geometric approach to monitoring threshold functions over distributed data streams". In *SIGMOD*, 2006.
[28] I. Sharfman, A. Schuster, and D. Keren. "A geometric approach to monitoring threshold functions over distributed data streams". *ACM Trans. Database Syst.*, 32(4), 2007.
[29] D. P. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 941–960, New York, NY, USA, 2012. ACM.
[30] K. Yi and Q. Zhang. Optimal tracking of distributed heavy hitters and quantiles. In *PODS*, 2009.