Functional Geometric Monitoring (FGM) for Distrusted Streams

The **FGM** [1] method constitutes the successor of Geometric Monitoring (**GM**) [2]. It provides substantial theoretical and practical improvements on the core ideas of the GM. The FGM method, is considered a generally applicable method, utilizing the so-called safe functions, while providing substantial benefits in terms of communication cost, performance, scalability, and robustness. Additionally, the FGM method is proven to be adapted under adverse conditions of the monitoring problem, such as the lack of monotonicity or tight monitoring bounds and skew in the distributions of data streams among the sites. Last but not least, the FGM method provides worst-case results, under standard assumptions on the monitoring problem. Finally, the FGM method can be easily integrated into any streaming distributed platform, providing arbitrary continuous query monitoring. The only requirement is the method parameterization by a problem-specific family of functions, which is the appropriate safe function.

The FGM method is based on the **Distributed Continuous Model** [3]. There are k distributed sites and a designated node, the Coordinator. Each site receives a stream of elements over time, possibly at varying rates, while the job of the Coordinator is to maintain an approximation of a function f continuously at all times. There is also a direct two-way communication channel between the sites and the Coordinator. The sites do not communicate with each other directly, but this is not a limitation since they can always exchange messages via the Coordinator. Note that broadcasting a message costs k times the communication for a single message. In this context, the site and the Coordinator maintain vectors. We consider the communication cost in two directions. Downstream communication cost consists of messages from sites to the Coordinator, while the upstream communication cost consists of messages from the Coordinator to the sites.

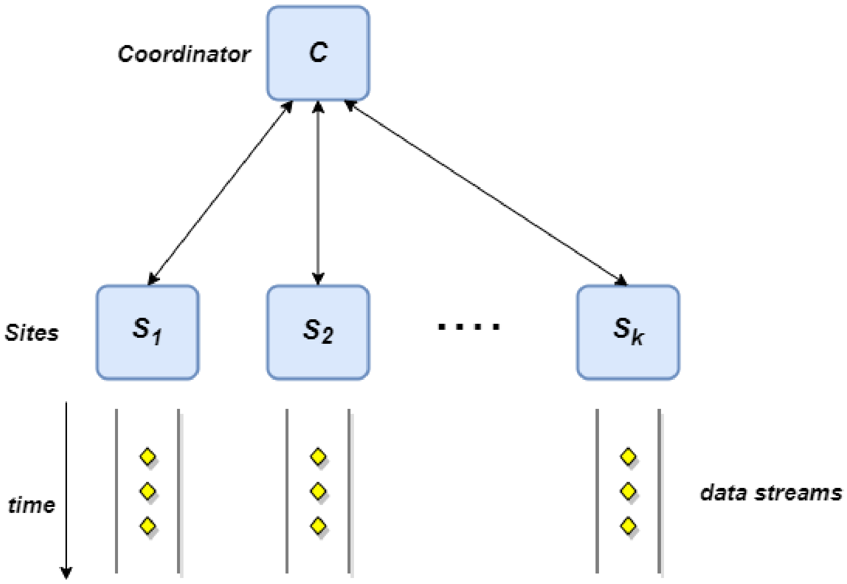

Figure 1: Distributed Continuous Model

The safe functions comprise real functions based on the monitoring problem. The configuration of the system of k sites is a (kD)-dimensional vector consisting of the concatenation of the k local drift vectors of sites. The sites collectively monitor the sum of these one-dimensional local drift vectors, and as long as the global sum is subzero, the monitoring bounds are guaranteed. The guarantee maintenance also involves the local sites periodically flushing the updates received to their local streams. When flushing occurs, the sites transmit their drift vector and the Coordinator updates the global estimator vector by adding the local drift vector, while the site resets its local drift vector.

The application of the method highlights the centrality of convexity in the monitoring. As long as the safe function of the monitoring problem holds the property of convexity, the method is applicable. The quality of the safe functions is crucial and is discussed in detail in [4]. Generally speaking, the safety of each round is guaranteed using the specific safe function.

The basic architecture implemented in Apache Flink is shown in the following picture. It consists of two major components: the Coordinator and the Workers(sites). Workers comprise a two-input keyed operator. The first input represents the Input Source which contains the training instances that need to be received from the sites, while the second input represents the Feedback Source which contains the control messages from the Coordinator. The Coordinator also comprises a two-input keyed operator. The first input represents the messages from the Workers, while the second one represents the Query Source which contains the queries posed by a user. In this case, the feedback loop between the Workers and the Coordinator is implemented using a Kafka topic that acts as a buffer between the Workers and the Coordinator. The worker behaves as a Kafka consumer on this topic, while the Coordinator behaves as a Kafka producer.

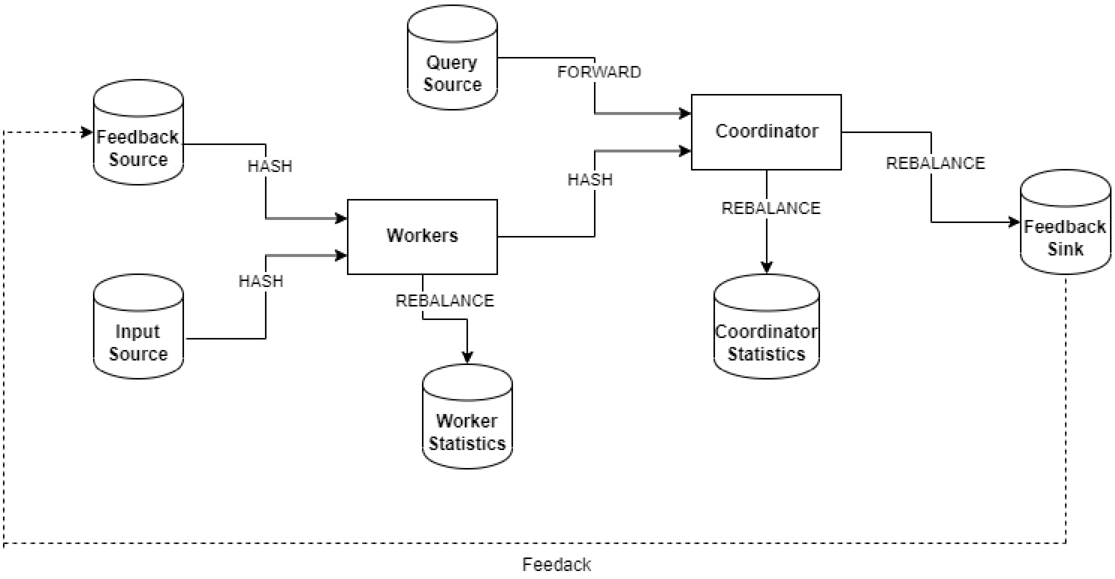

Figure 2: Project Architecture

Bibliography

[1] V. Samoladas and M. Garofalakis, “Functional Geometric Monitoring for Distributed Streams,” 2019, p. 12.

[2] I. Sharfman, A. Schuster, and D. Keren, “A geometric approach to monitoring threshold functions over distributed data streams,” *ACM Trans. Database Syst.*, vol. 32, no. 4, p. 23, Nov. 2007.

[3] G. Cormode, “The continuous distributed monitoring model,” *ACM SIGMOD Rec.*, vol. 42, no. 1, pp. 5–14, May 2013.

[4] A. Lazerson, I. Sharfman, D. Keren, A. Schuster, M. Garofalakis, and V. Samoladas, “Monitoring distributed streams using convex decompositions,” *Proc. VLDB Endow.*, vol. 8, no. 5, pp. 545–556, Jan. 2015.