Chapter 7
Distributed Adaptive Windowed Stream Join Processing

Tri Minh Tran
University of Vermont, USA

Byung Suk Lee
University of Vermont, USA

ABSTRACT
This paper presents an adaptive framework for processing a window-based multi-way join query over distributed data streams. The framework integrates distributed plan modification and distributed plan migration within the same scope by using a building block called the node operator set (NOS). An NOS is housed in each node that participates in the join execution, and specifies the set of atomic operations to be performed locally at the host node to execute its share of the global execution plan. The plan modification and migration techniques presented are for the case of updating the NOSs centralized at a single node and the case of updating them distributed at each node. The plan modification is triggered by the change of stream statistics and adjusts the join execution order and placement greedily to satisfy a cost invariant. The plan migration uses the distributed track strategy to accelerate the migration of window extents to new nodes. The migration of all window extents is synchronized. Experiments confirm the effectiveness of the developed adaptive framework on reducing the join execution cost and indicate a small additional adaptation-overhead for distributing the NOS update.

INTRODUCTION
Distributed data stream processing (Amini, Jain, Sehgal, Silber, & Verscheure, 2006; Cormode, Muthukrishnan, & Zhuang, 2006; Das, Ganguly, Garofalakis, & Rastogi, 2004; Kumar, Cooper, & Schwan, 2005; Kumar, Cooper, Cai, Eisenhauer, & Schwan, 2005; Olston, Jiang, & Widom, 2003; Seshadri, Kumar, & Cooper, 2006; Sharfman, Schuster, & Keren, 2006) is a fast growing research area in the data stream field. The driving force behind this growth is the widely deployed and utilized diverse distributed computing environments such as the telecommunication networks,
In this paper, we focus on multi-way window-based stream join query which is an important class of queries in distributed stream applications. For example, in network packet monitoring, the network administrator may want to monitor the traffic of data packets passing through different routers with the objective of finding packets with the same destination IP address. For this task, a distributed stream join query is needed to join the streams of packets from those routers. As another example, in building-monitoring using sensor networks, one may want to keep track of the temperature, humidity, and light intensity measured by sensors in a room. The sensor readings of each measurement type are sent to their respective sinks as a stream. The monitoring task in each room can be specified as a distributed stream join query that joins on the same room id from three sensor reading streams. Similar distributed join queries are also needed in many other stream applications such as financial stock ticker analysis, telephone call monitoring, and news article filtering.

An important aspect of query processing today is the adaptivity, that is, adjusting the query execution plan adaptively to the changing data profile and system environment. In light of data stream query processing, the fluctuations of stream statistics (e.g., stream rates, join selectivity) or available system resources (e.g., memory, CPU time) are the changes to adapt to. This paper focuses on the former, i.e., stream statistics.

To the best of our knowledge, all existing research on adaptive stream join processing have been done in the centralized environment (Babu, Motwani, Munagala, Nishizawa, & Widom, 2004; Babu, Munagala, Widom, & Motwani, 2005; Zhu, Rundensteiner, & Heineman, 2004) and none in the distributed environment. In the distributed environment, a different query processing model is needed because some or all join steps are performed at different nodes across the network and the communication overhead for these join steps should be taken into consideration in query execution planning, and thus the solutions developed in the centralized environment are not applicable.

Moreover, there is a division in the scope of the existing work. Adaptive query processing framework encompasses query plan modification and query plan migration. Query plan modification involves the process of updating current execution plan to a new, better plan, and query plan migration handles the switch from the current execution plan to the new plan. As far as we know, however, there does not exist any work done on adaptive stream query processing with both in one scope. All the existing work address either the query plan modification (Babu, Motwani, Munagala, Nishizawa, & Widom, 2004; Babu, Munagala, Widom, & Motwani, 2005) or the query plan migration (Zhu, Rundensteiner, & Heineman, 2004), not to mention they are not distributed. This disconnection naturally misses out the interaction between the two key aspects of adaptive query processing.

This paper aims to advance the state of the art by providing a solution to the distributed plan modification and migration problem for executing stream join queries adaptively within the same framework as the stream statistics change in a distributed environment.

The adaptation of query processing in this paper is triggered by an event defined as a significant change of stream statistics such as stream rate, join selectivity, and tuple size. One challenge in achieving this adaptivity in a distributed environment comes from the fact that the query execution plan specifies only the steps for executing a query but not specifically what each node needs to do. Thus, each node has to make its own local decision on what part of the distributed global plan it needs to execute, when to adjust its own part, and how to adjust it.