Chapter 2.5
An Autonomous Agent Approach to Query Optimization in Stream Grids

Saikat Mukherjee
International Institute of Information Technology, India

Srinath Srinivasa
International Institute of Information Technology, India

Krithi Ramamritham
Indian Institute of Technology, India

ABSTRACT

Stream grids are wide-area grid computing environments that are fed by a set of stream data sources, and Queries arrive at the grid from users and applications external to the system. The kind of queries considered in this work is long-running continuous (LRC) queries, which are neither short-lived nor infinitely long lived. The queries are "open" from the grid perspective as the grid cannot control or predict the arrival of a query with time, location, required data and query revocations. Query optimization in such an environment has two major challenges, i.e., optimizing in a multi-query environment and continuous optimization, due to new query arrivals and revocations. As generating a globally optimal query plan is an intractable problem, this work explores the idea of emergent optimization where globally optimal query plans emerge as a result of local autonomous decisions taken by the grid nodes. Drawing concepts from evolutionary game theory, grid nodes are modeled as autonomous agents that seek to maximize a self-interest function using one of a set of different strategies. Grid nodes change strategies in response to variations in query arrival and revocation patterns, which is also autonomously decided by each grid node.

DOI: 10.4018/978-1-4666-0879-5.ch2.5

INTRODUCTION

Stream grids are grid computing environments that are fed with streaming data sources from instrumentation devices like cameras, RFID (radio-frequency identification) sensors, network monitoring or other applications. Queries by users or applications seek to tap into one or more such streams. The main costs for such queries include bandwidth costs and bookkeeping costs at each grid node. In such scenarios, there are conflicting optimization requirements. While end-users prefer reduced latency, individual grid nodes prefer reduced book-keeping costs and the grid as a whole seeks to minimize bandwidth consumption.

Queries in such grids may originate on any node and seek data from any stream or a set of streams. Such queries are typically long lived, but not necessarily infinitely long lived.

Traditionally, query optimization has been addressed for two classes of queries: transient or "one-shot" queries, and infinite or "standing" queries (Cormode & Garofalakis, 2007). One-shot queries are transient in nature and have very short life spans. In such environments, the speed of query processing takes precedence over computing the globally optimal execution plan. On the other hand, for standing queries whose lifetimes are practically infinitely long, it is desirable to invest time and resources to obtain optimal execution plans. Queries on stream grids however, are of a third interim type that we call long-running continuous (LRC) queries or "open-world" queries. These queries are "open" in the sense that, the system does not have control on when and where a query appears, seeking which stream, and when it is revoked. Since queries are typically long lived, ignoring query plan optimization would not be a good idea; at the same time optimizing query execution for the best possible plan is also undesirable, since queries may terminate or new queries may enter the system at any time.

An example of the kind of challenges faced in LRC queries is illustrated in Figure 1 (a). Grid node SN1 is a stream data source and the three other nodes CN1, CN2 and CN3 are nodes responsible for answering user queries. There is also a distance function d(x; y) defined between pairs of nodes that calculates the latency in shipping a data stream between pairs of nodes. Each query has to be answered with as little latency as possible. Assume that the nodes are arranged such that d(CN1;CN2) > d(CN2;CN3). Now if a query for S1 arrives at CN1 at time t1, it is optimal for CN1 to request for the stream at the source node SN1 (Figure 1 (a)). Suppose a second query and third query for SN1 arrive at time t2 and time t3 on compute nodes CN3 and CN2 respectively. When a query appears on a node, it is apparent that latency can be minimized by fetching the required data from the nearest available source. Given this, the routing of the data streams would be as shown in Figure 1 (b). It is immediately apparent that the routing of the data streams as shown in Figure 1 (b) is not optimal from the global (grid-wide) perspective. The optimal strategy would be as shown in Figure 1 (c). Now, if the query at node CN3 is revoked as shown in Figure 1 (d), the routing of the data streams would remain the same, as node CN3 is still active given the need to serve node CN2. This again is sub-optimal. It is clear that arrival and revocation of queries create a need for re-optimization. However determining the globally optimal query plan on every new query arrival or revocation, and enforcing it over the entire grid is infeasible.

In this paper, we explore the notion of *emergent* optimization where grid nodes act as self-interested autonomous agents and optimize on local properties. Local optimization is facilitated by a set of *strategies* using which nodes connect to other nodes. However, the choice made by each node affects not only its own optimality, but also the global optimality of the grid. In order to reconcile mismatches between local and global optimality, the choice of strategy is changed in an evolutionary fashion. The evolutionary dynamics are derived from Axelrod's now classic model of