Performance Modeling of Spatio-Temporal Algorithms Over GEDS Framework

Jonathan Cazalas, University of Central Florida, USA
Ratan Guha, University of Central Florida, USA

ABSTRACT

The efficient processing of spatio-temporal data streams is an area of intense research. However, all methods rely on an unsuitable processor (Govindaraju, 2004), namely a CPU, to evaluate concurrent, continuous spatio-temporal queries over these data streams. This paper presents a performance model of the execution of spatio-temporal queries over the authors’ GEDS framework (Cazalas & Guha, 2010). GEDS is a scalable, Graphics Processing Unit (GPU)-based framework, employing computation sharing and parallel processing paradigms to deliver scalability in the evaluation of continuous, spatio-temporal queries over spatio-temporal data streams. Experimental evaluation shows the scalability and efficacy of GEDS in spatio-temporal data streaming environments and demonstrates that, despite the costs associated with memory transfers, the parallel processing power provided by GEDS clearly counters and outweighs any associated costs. To move beyond the analysis of specific algorithms over the GEDS framework, the authors developed an abstract performance model, detailing the relationship of the CPU and the GPU. From this model, they are able to extrapolate a list of attributes common to successful GPU-based applications, thereby providing insight into which algorithms and applications are best suited for the GPU and also providing an estimated theoretical speedup for said GPU-based applications.

Keywords: Computation Sharing, Continuous Query, Graphical Processing Unit (GPU), kNN, Location-Based Services, Mobile Database Systems, Parallel Processing, Performance Model, Range Query, Spatio-Temporal Data Streams

INTRODUCTION

The ever-increasing popularity of mobile devices, coupled with the advances in wireless technologies and global positioning systems, has helped to create an environment where data access truly is anywhere at any time. The wireless communication market has grown by leaps and bounds, and these new technologies bring a flood of location-dependent applications where it is not only desirable, but often critical, to provide real-time query results to the respective users (Cai, Hua, Cao, & Xu, 2006). With the worldwide proliferation of GPS-enabled devices and the staggering growth of smartphones (Chang et al., 2009), it is only a matter of time until these applications, often referred
to as Location Based Services (LBS), truly become ubiquitous.

Traditionally, mobile object databases augment the standard database model of persistent data storage and complex querying by adding new models and index structures geared to store, track, and process the locations of moving objects efficiently (Guttman, 1984; Beckmann et al., 1990; Gedik & Liu, 2006). R-trees (Guttman, 1984) have been the most popular mechanism for spatial indexing, and many variants have been proposed, including the R*-tree (Beckmann et al., 1996), X-tree (Berchtold et al., 1996), Lazy Update R-tree (Kwon et al., 2002), and a plethora of other suggestions. Additionally, there is a large body of research focused on reducing the computational burden of continuously monitoring and evaluating real-time queries over these mobile objects. Such works include MQM (Cai, Hua, Cao, & Xu, 2006), MobiEyes (Gedik & Liu, 2006), Domino (Wolfson et al., 1999), and CAT (Trajcevski et al., 2004), to name a few. While these models and structures did initially extend the research in this area, the past few years have witnessed the emergence of a new class of data intensive applications that often require the continuous processing of potentially unbounded sequences of transient data, called data streams. Examples include financial tickers, internet traffic, sensor data, and transaction logs. The high arrival rates of these spatio-temporal data streams, coupled with their massive data sizes, makes it infeasible for traditional DBMS techniques to store, query, or index these streams and therefore dictates the need for better solutions.

In simplest terms, a data stream can be defined as “a sequence of characters or bits that is too large to be viewed in its entirety” (Hartman & Watters, 1990). Several works have convincingly argued that the two research fields of spatio-temporal data streams and the management of moving objects can naturally come together (Chandrasekaran & Franklin, 2003; Ghanem et al., 2007; Mokbel et al., 2004). For example, the output of a GPS receiver, monitoring the position of a mobile object, is viewed as a data stream of location updates. This data stream of location updates, along with those from the plausibly many other mobile objects, is received at a centralized server, which processes the streams upon arrival, effectively updating the answers to the currently active queries in real time. From this model, it becomes clear that additional applications could benefit from modeling location updates as streaming data, including, but not limited to, network traffic, time series data, telephone records, weather data, web click streams, and the list goes on.

Unfortunately, most of the recent research in data stream management systems (Adabi et al., 2003; Babu & Widom, 2001; Chandrasekaran & Franklin, 2003) is insufficient, as they overlook the spatial and temporal qualities of both the data streams and the continuous queries over these streams. And it is these two qualities, specifically, that distinguish continuous query processing in spatio-temporal data streams from traditional data streams. Because both queries and data can continuously change their locations, spatio-temporal data streams are viewed as a series of location updates rather than the append-only model of classical data streams (Mokbel & Aref, 2008). Additionally, the temporal quality stipulates that a mobile object may be added to or removed from the result set of the spatio-temporal query, an example being GPS-equipped vehicles moving in and out of a query region. Because these queries are continuous in nature, any delay would result in an obsolete response. Therefore, it is vital to procure scalable and efficient algorithms for the processing of continuous spatio-temporal queries over data streams. To this end, SINA was proposed to address this issue by exploiting shared execution and incremental evaluation (Mokbel et al., 2004). The main drawback of SINA was the reliance on physical disk-storage to perform its operations. SOLE was then proposed as a scalable, in-memory algorithm, which uses an incremental evaluation paradigm and a grid structure to evaluate concurrent, continuous spatio-temporal queries over data streams (Mokbel & Aref, 2008). SOLE avoids the slow, physical data storage, but is also limited based on memory. As a result, a load-shedding

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Location Update Improvement Using Fuzzy Logic Optimization in Location Based Routing Protocols in MANET
www.igi-global.com/chapter/location-update-improvement-using-fuzzy/69032?camid=4v1a