Parallel Processing Data Streams in Complex Event Processing Systems

Fuyuan Xiao[1], Cheng Zhan[1], Hong Lai[1], Li Tao[1]
1. School of Computer and Information Science, Southwest University, Chongqing, 400715, China
E-mail: xiaofuyuan@swu.edu.cn, zhanc@swu.edu.cn, hlai@swu.edu.cn, tli@swu.edu.cn

Abstract: For distributed complex event processing systems, handling high volume and continuous data streams with high throughput are required for further decision support. Due to the specific properties of pattern operators, it is difficult to process the data streams in parallel over complex event processing systems. To address the issue, a novel parallel processing strategy is proposed. The proposed method can keep the complex event processing system working stably and continuously via the elapsed time. Finally, the utility of our work is demonstrated through the experiments on the StreamBase system.

Key Words: Data streams, Complex event processing, Parallel processing, Pattern operators, Decision support

1 INTRODUCTION

Nowadays, there has been increasing interest in distributed applications for decision support which require processing continuously flowing data from geographically distributed sources at unpredictable rate to obtain timely responses to complex queries, such as data stream processing (DSP) systems [1–6] and complex event processing (CEP) systems [7–14]. In CEP systems, event streams are processed in or near real-time for a variety of purposes, from wireless sensor networks to financial tickers, from traffic management to click-stream inspection [15, 16]. In those application domains, continuous and highly-available event stream processing with high throughput is critical for dealing with real-world events.

Nevertheless to say, most of these strategies which exclusively focus on aggregate queries or binary equi-joins in DSP systems cannot be simply and directly used in CEP systems which focus on multi-relational non-equi-joins on the time dimension. Whereas, the volume and input rates of the data would become large as in event stream processing, especially in the big data applications [17, 18]. The increasing of the input rate of a stream may cause bottlenecks. Such a behavior gives rise to poor quality of query results and loses the QoS guarantees of the system.

To address these issues, we propose a parallel processing strategy which can be used to keep the system running stably and continuously. Specifically, the CEP system based on the proposed parallelization architecture can split the input stream into parallel sub-streams to realise a scalable execution of continuous pattern query over event streams. The parallel processing strategy can keep the CEP system working stably and continuously via the elapsed time. The utility of our work is substantiated through the experiments on the StreamBase [19] system.

The rest of this paper is organized as follows. Section 2 briefly introduces the preliminaries of this paper. After that, a parallel processing strategy is proposed for distributed complex event processing systems in Section 3. Section 4 demonstrates the utility of our proposal through the experiments on the StreamBase system. Finally, conclusions are given in Section 5.

2 PRELIMINARIES

In this section, we briefly introduce the basic event model, nested pattern query language, and the pattern operators based on related studies, e.g., [9–11, 14, 20].

2.1 Event Model

An event which represents an instance and an atomic, is an occurrence of interest at a point in time. Basically, events can be classified into primitive events and composite events. A primitive event instance is pre-defined single occurrence of interest that cannot be split into any small events. A composite event instance that occurs over an interval is created by composing primitive events.

Definition 2.1 *A primitive event e_i is typically modeled multi-dimensionally denoted as e_i=e($e_i.t$, ($e_i.st = e_i.et$), $< a_1, \ldots, a_m >$), where, for simplicity, we use the subscript i attached to a primitive e to denote the timestamp i, $e_i.t$ is event type that describes the essential features of e_i, $e_i.st$ is the start time-stamp of e_i, $e_i.et$ is the end time-stamp of e_i, $< a_1, \ldots, a_m >$ are other attributes of e_i and the number of attributes in $e(\cdot)$ denotes the dimensions of interest.*

Definition 2.2 *Based on Definition 2.1, a composite event is denoted as e=e($e.t$, (($e.st = \min_{1\le i\le n} e_i.st$) $<$ ($e.et = \max_{1\le i\le n} e_i.et$)), $< a_1, \ldots, a_g >$).*

This work is supported by Fundamental Research Funds for the Central Universities (Nos. XDJK2015C107, SWU115008, XDJK2016C043, SWU115091) and the Education Teaching Reform Program of Higher Education (No. 2015JY030). The paper is also partially supported by A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).