Events on the Edge

Shariq Rizvi Shawn R. Jeffery Sailesh Krishnamurthy Michael J. Franklin
Nathan Burkhart Anil Edakkunni Linus Liang

University of California, Berkeley
http://hifi.cs.berkeley.edu

ABSTRACT

The emergence of large-scale receptor-based systems has enabled applications to execute complex business logic over data generated from monitoring the physical world. An important functionality required by these applications is the detection and response to complex events, often in real-time. Bridging the gap between low-level receptor technology and such high-level needs of applications remains a significant challenge.

We demonstrate our solution to this problem in the context of HiFi, a system we are building to solve the data management problems of large-scale receptor-based systems. Specifically, we show how HiFi generates simple events out of receptor data at its edges and provides high-functionality complex event processing mechanisms for sophisticated event detection using a real-world library scenario.

1. INTRODUCTION

Widespread deployment of receptor devices such as wireless sensor networks and RFID technologies is enabling many new applications and enhancing existing ones.

These receptors are often deployed as a part of a much larger system, as in the case of a geographically distributed supply chain management system. The applications supported by these systems need to execute complex business logic, like event-oriented sense-and-respond functionality, on top of the underlying receptor data. Scenarios requiring real-time detection and processing of sophisticated events in such systems often occur at the edges of the system (as in the case of a store or warehouse in a supply chain). We call this class of events *edge events*. These events are often of a critical nature (e.g., shoplifting, fire detection) and demand timely detection and response.

As an example of sophisticated event-driven logic in an application, consider an RFID-enabled library scenario (that we use in our demonstration as detailed in Section 3). The library, equipped with a variety of receptors, like RFID readers and motion sensors, uses a receptor-based system to drive its main tasks:

1. Providing multiple ways for patrons to check out books. This may include a standard counter-based checkout as well as a fast path that allows patrons with RFID tags to simply walk out with books (that are also RFID-tagged). To make this possible, the system should provide the application with ways to specify complex events and associate many of them with the same logical event ("The person with ID 7 is checking out book 10").
2. Avoiding thefts based on the timely detection of complex events ("Book 12 is leaving the library without being checked out").

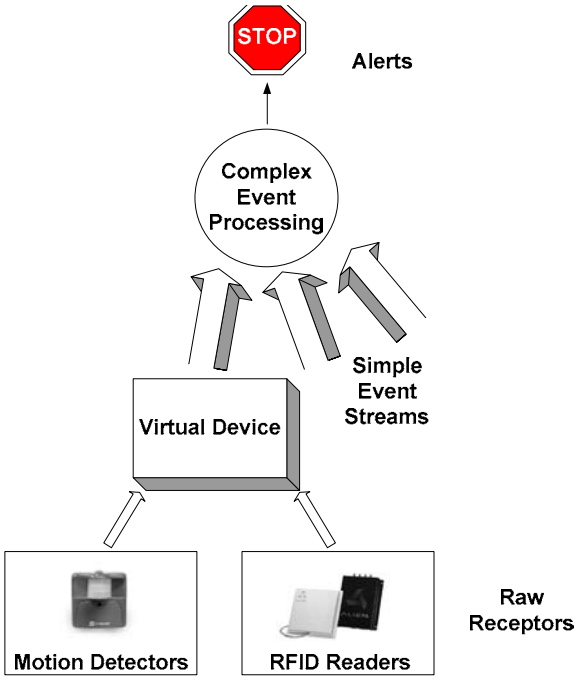

Figure 1: Edge Event Processing in HiFi

*This work was funded in part by NSF under ITR grants IIS-0086057 and SI-0122599, by the IBM Faculty Partnership Award program, and by research funds from Intel, Microsoft, and the UC MICRO program.

SIGMOD 2005 June 14–16, 2005, Baltimore, Maryland, USA

3. Enforcing policies like a limit on the maximum number of books a particular patron can check out at any point of time, and taking suitable actions when these policies are violated. This requires that the system provide ways to join real-time event information with stored data.

To process edge events in real-time in such scenarios, a receptor-based system must provide functionality that enables execution of sophisticated event-oriented logic. We demonstrate edge event processing in the context of *HiFi* [2], a data management infrastructure for receptor-based systems that uses data stream query processing at all levels of the system to acquire, filter, and aggregate data from multiple receptor devices.

Figure 1 depicts how our demonstration combines multiple types of receptors, including RFID readers and motion sensors, to do edge event processing, and highlights the two key challenges present in such scenarios: receptor interaction for the generation of reliable application-level simple events and techniques for powerful and flexible complex event processing over these simple event streams.

HiFi utilizes *Virtual Devices* (*VICEs*) to interact with heterogeneous receptor devices to produce reliable, application-level simple events. We discuss VICEs in Section 2.1. The complex event processing used to correlate and aggregate these simple events into more sophisticated application-level events using language extensions to SQL is described in Section 2.2.

2. EDGE EVENT PROCESSING

Edge events essentially fall into two broad categories:

- Simple events: These events are equivalent to a single tuple in a data stream. Examples of this type of events include the detection of an item designated by an RFID tag at a certain location or the observation that the temperature of some spatial region rises above a threshold.
- Complex events: These events correlate multiple simple events across time, either on a single stream or across disparate streams. Examples of these events include the simultaneous overheating of multiple shelves in a warehouse within a 30 second time window or shoplifting in a retail store (an item leaves the store without a preceding purchase event for that item).

Hence, in order to perform processing based on edge events the system must first interact with the relevant receptor(s) to generate simple events and then use that information to do complex event processing.

2.1 Virtual Devices

The first task of edge event processing in HiFi is the initial generation of simple events from receptor data. The data produced by physical receptors, however, are notoriously raw: they are dirty, unreliable, and tend to have little meaning to an application. Before this data can be used, they must be cleaned, processed, and transformed to a form that the application can use directly.

To this end, HiFi employs *Virtual Devices* (*VICEs*) to provide what we term *metaphysical data independence* [3]. Just as physical data independence protects applications from changes to the underlying data storage and layout, metaphysical data independence shields HiFi from unreliable data and other issues that arise when dealing with physical devices. Thus, HiFi can use traditional stream data processing techniques (and the complex event processing discussed below) over receptor data that has been appropriately processed by VICEs.

To provide metaphysical data independence, a VICE processes receptor data through a cascade of programmable stages that successively clean, aggregate, and convert raw receptor data into reliable, application-level simple events.

In this demonstration, we show the following features of virtual devices:

- Increased reliability of receptor-based data.
- Conversion of raw receptor data from multiple types of receptors into application-level simple events.

2.2 Complex Event Processing

To run sophisticated event-oriented logic over streaming data, the system should allow joining and aggregation of multiple streams under intricate notions of time, ordering and negation. For example, the detection of a (possibly suspicious) complex activity pattern at a library door translates into a complex event query over a data stream produced by an RFID receptor on the door.

Languages and techniques for such complex event processing have been proposed by the active database community [7] in the context of traditional non-streaming database systems. In contrast, SQL (and its extensions for querying streaming data), which is the accepted language for traditional declarative queries, does not provide natural ways of expressing sophisticated queries over ordered data (like time-ordered data streams)[4],[5]. In our demonstration, we present a unified language and system for both traditional stream query processing and event processing. Our mechanisms provide high flexibility in event specification. For example, in the library scenario described earlier, the system can associate the checkout of a book by a patron with multiple physical events over receptor streams – the book's tag being seen at the exit followed by the patron's tag within a short amount of time, or the patron's tag being seen at the exit followed by the book's tag within a short amount of time. Such expressive power is essential to capture real-world complex events.

In general, event specifications may span large scales of time and space (e.g., a CEO who wants an alert when the nation-wide sale of a product in the previous week goes below a threshold). However, in this demo we focus on edge event processing that involves real-time detection of events in a small area of interest (e.g., a library).

3. DEMONSTRATION SCENARIO

In this demonstration, we show the generation of simple events and complex edge event processing in HiFi through a library scenario. This demonstration showcases HiFi's ability to express and execute real-world application-level logic through two mechanisms: reliable simple event generation using VICEs, and complex event processing unified with data stream processing.

We place RFID tags on books and demo participants. RFID readers, motion detectors, and other receptors monitor various logical areas of the library, including a book shelf, a checkout desk, and an exit. Patrons may check out books through a traditional checkout desk or by wearing an RFID tag linked to the patron's account. In the latter case, the system automatically checks out books for the patron as he/she exits the library.

The data produced by the receptors are processed by VICEs to produce a stream of reliable simple events, such as:

- "The book with ID 10 is on the shelf"
- "The person with ID 7 is leaving the library"
- "The book with ID 4 is being checked out"

These events are fed into HiFi's complex event processing mechanisms to derive complex events. These complex events correlate and aggregate multiple simple events across time, providing the ability to express sequencing and negation, as well as the ability to join real-time simple events with stored data (like the number of books a patron has checked out in the past). Some examples of complex events in this demonstration include the following:

- "The person with ID 7 is checking out book 10". Detecting this event in a reliable fashion requires associating the logical checkout event with multiple physical patterns over receptor streams, as discussed in Section 2.2.
- "Book 12 is being taken out of the library without being checked out".
- "The person with ID 7 has checked out more books than her limit". This requires joining real-time event information with stored data.

The goal of this demonstration is to provide data management support for normal library operations (e.g., checkouts, inventory) as well as alerts (e.g., book theft, over-limit checkouts). This application underlines the HiFi theme of providing a highly functional platform for executing sophisticated event-oriented logic at the edges of large-scale receptor-based systems.

4. REFERENCES

[1] Owen Cooper, Anil Edakkunni, Michael J. Franklin, Wei Hong, Shawn R. Jeffery, Sailesh Krishnamurthy, Frederick Reiss, Shariq Rizvi, Eugene Wu. *HiFi: A Unified Architecture for High Fan-In Systems (System Demonstration)*. In VLDB 2004: 1357-1360.

[2] Michael J. Franklin, Shawn R. Jeffery, Sailesh Krishnamurthy, Frederick Reiss, Shariq Rizvi, Eugene Wu, Owen Cooper, Anil Edakkunni, Wei Hong. *Design Considerations for High Fan-in Systems: The HiFi Approach*. In CIDR 2005: 290-304.

[3] Shawn R. Jeffery, Gustavo Alonso, Michael J. Franklin, Wei Hong, Jennifer Widom: *Virtual Devices: An Extensible Architecture for Bridging the Physical-Digital Divide*. UC Berkeley Technical Report UCB-CS-05-1375, March 2005.

[4] Raghu Ramakrishnan, Donko Donjerkovic, Arvind Ranganathan, Kevin S. Beyer, Muralidhar Krishnaprasad. *SRQL: Sorted Relational Query Languag*e. In Statistical and Scientific Database Management 1998: 84-95.

[5] Reza Sadri, Carlo Zaniolo, Amir M. Zarkesh, Jafar Adibi. *Optimization of Sequence Queries in Database Systems*. In PODS 2001.

[6] Robbert van Renesse, Kenneth P. Birman, Werner Vogels. A*strolabe: A robust and scalable technology for distributed system monitoring, management, and data mining*. In ACM Trans. Comput. Syst. 21(2): 164-206, 2003.

[7] Narain H. Gehani, H. V. Jagadish, Oded Shmueli. *Composite Event Specification in Active Databases: Model & Implementation*. In VLDB 1992: 327-338