Tutorial: Event Processing Using Database Technology

Mani Chandy
California Institute of Technology
mani@cs.caltech.edu

Dieter Gawlick
Oracle Corporation
dieter.gawlick@oracle.com

ABSTRACT

This tutorial deals with applications that help systems and individuals respond to critical conditions in their environments. The identification of critical conditions requires correlating vast amounts of data within and outside an enterprise. Conditions that signal opportunities or threats are defined by complex patterns of data over time, space and other attributes. Systems and individuals have models (expectations) of behaviors of their environments, and applications notify them when reality – as determined by measurements and estimates – deviate from their expectations. Components of event systems are also sent information to validate their current models and when specific responses are required. Valuable information is that which supports or contradicts current expectations or that which requires an action on the part of the receiver. A major problem today is information overload; this problem can be solved by identifying what information is critical, complementing existing pull technology with sophisticated push technology, and filtering out non-critical data.

Categories and Subject Descriptors

H.2.4 [**Database Systems**]: Rule Based Databases, Textual Databases, Distributed databases, Query Processing.

General Terms

Design, Economics, Languages, Theory.

Keywords

Events, streams, distributed systems, errors, false positives, false negatives, statistics.

1. OVERVIEW

This tutorial deals with applications that help systems and individuals respond to critical conditions in their environments. A major problem today is information overload; this problem can be solved by identifying what information is critical, complementing existing pull technology with sophisticated push technology, and filtering out non-critical data. Rick Hayes-Roth captured this requirement with the term VIRT (Valuable Information at the Right Time).

Event Driven Applications help individuals and systems take timely action to respond to environmental conditions. Such applications have many facets including: acquisition of streams of data both by push and pull, data storage, rapid business intelligence and correlation, messaging networks and alerting systems. The enterprise software stack includes many of the components required for EDA including rules engines, ESB (Enterprise Service Bus), publish/subscribe systems, BI tools, Business Process Management and Business Application Monitoring tools, and of course, databases. Database researchers have been at the forefront in designing systems dealing with continuous queries and continuous analytics.

Many commercial databases with their complementary enterprise software stacks provide all, or almost all the components required for event-driven applications. These systems have capabilities for triggers and online log-mining, message queuing, publish/subscribe, rules engines, and (increasingly) support for continuous queries. Enterprise software, whether for event-driven or more traditional applications, has requirements dealing with stability, maintainability, security and other aspects. Commercial databases and complementary software makes them ideal for developing event-driven applications for the enterprise.

The tutorial has two parts. Part 1 provides a conceptual view of event driven applications. Part 2 discusses existing support in commercial databases and complementary software. Both parts will be illustrated through shared use cases.

2. TOPICS

The outline of the tutorial is as follows.

2.1 A conceptual view of Event Driven Applications

1. Characterization of event-driven applications
    a. Essential characteristics of event-driven applications
    b. What types of applications do event-driven architectures support? Not support?
    c. Examples of applications from different fields: Sensor networks and RFID; customer relationship management; supply chain and logistics; finance.
    d. Formal specification of event-driven applications
    e. Specifying information event stream processing for monitoring and business intelligence.
    f. Specifying sense and respond systems. Management by exception; specifying expected behavior by models; identifying when reality deviates from expectation; updating models.

2.2 Support in commercial databases

a. Data base as message source. Databases allow users (subscribers) to define what they define as an event, what message(s) should be created if an event occurs and who should receive a specific message.

SIGMOD'07, June 12–14, 2007, Beijing, China.
ACM 978-1-59593-686-8/07/0006.

i. Capturing events using database triggers.

ii. Capturing events using journals.

iii. Capturing events using queries.

1. If queries reference the current state the change of the result set is perceived as an event.

2. If queries reference the current and previous states the occurrence of a specified pattern is an event.

b. Support of message storage (staging areas/queues).

i. Storing captured messages.

1. Accepting messages from clients using an extended INSERT interface.

2. Accepting messages that are created in foreign systems and delivered to a database message storage.

3. Storing internally created messages; there are significantly opportunities for optimization.

ii. Operational Characteristics.

1. Security, auditing, tracking.

2. Performance, scalability.

3. Recoverability, availability, transactional support.

c. Evaluating critical conditions.

i. Rules technologies for.

1. Support for publish/subscribe as well as subscribe-to-publish is a natural extension of database technology.

2. Supporting *expressions as data* allows databases to significantly extend traditional publish/subscribe technology.

3. The support for continuous queries provides a comprehensive base for CEP.

4. (Continuous) Analytics provide the technology to identify valuable Continuous Queries.

ii. Rules technologies can be used to evaluate external data; e.g., data can be presented to a rules service and the rules service will identify interested consumers.

iii. The second use of rules technology is the evaluation of internal data; e.g., messages in queues or data in the database. The evaluation of internal data can significantly be optimized

iv. Operational characteristics.

1. Security, auditing, tracking,

2. Performance, scalability.

a. Large rule sets and Continuous Queries.

b. Frequently changing rules sets and Continuous Queries.

d. Message consumption.

i. Messages may be consumed locally to the message store. To do this the message store may have to activate applications as needed.

ii. Distribution of messages.

1. Forwarding messages to other staging areas.

2. Forwarding messages to external services.

iii. Operational Characteristics.

1. Security, auditing, tracking.

2. Performance, scalability.

3. Recoverability, availability, transactional support.

e. Use Cases.

i. Financial Services require event processing for to execute online transaction, to react to opportunities and threads and to identify new opportunities and threads.

ii. Utilities use event processing for monitoring current usage and usage patterns.

iii. ChemSecure, a NASA project to manage hazardous material. Any thread has to be known to the people who are authorized and able to respond most efficiently.

iv. SensorNet, a US government project to capture a wide variety of data and deliver them to first responders who are authorized, available and able to respond most efficiently.

f. Discussion of selected features of commercial database products. Some features are chosen based on their high level of innovation other features are chosen because they are used in highly visible mission critical applications.