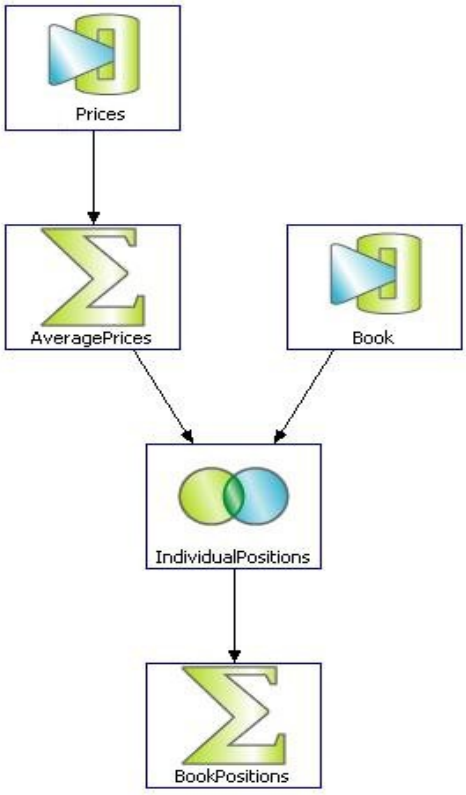
Go with the Flow: Dataflow in the Aleri Streaming Platform

By: Jon Riecke, Lead Platform Architect
7 February 2008

The Aleri Streaming Platform is a software infrastructure for processing and analyzing event data—e.g., stock trades, messages from equipment, orders from customers—into aggregated, correlated, and alerting data. It's designed to handle vast numbers of messages and process those messages in real-time. It's like a database turned sideways: instead of saving records into a data store, and querying them later, the Aleri Streaming Platform transforms the data as it enters the system. In essence, the queries are computed continuously, and the results are pushed out to subscribers.

Another way, though, to look at the system is as a dataflow programming environment. Dataflow is a particularly simple form of parallel computing, in which a program is a connected graph of computational elements. The connections specify which way the data flows. A node in the graph waits for messages from its inputs; when it receives an event, it computes one or more new events that are sent to downstream nodes. A node computes only when it has a waiting message; it sits idle otherwise. Conversely, when messages arrive too quickly, the node queues them up for later processing. It helps that the dataflow model fits the notion of streaming events too.

To make this more concrete, here's the graph of a program—what's called a *data model*—in the Aleri Streaming Platform.



Events enter the program at two source streams in this example: a Book stream, which has more static information about number of shares held, and a Prices

stream, which handles the current prices of stocks. A few derived streams process the data. The AveragePrices stream enhances the Prices with information about running averages. The IndividualPositions stream correlates data from Book and AveragePrices. The BookPositions stream rolls up the information across books. As events flow in, they get pushed along the arcs of the graph. Each node runs independently of the others.

The dataflow model of computation has a long history. The roots extend to the parallel computer hardware design of Jack Dennis at MIT, and the theoretical work of Gilles Kahn at INRIA in Paris. Experimental machines and programming languages were built around dataflow, but the idea has found its way into commercial microprocessors too. Most microprocessors now break up complex operations into simpler ones and feed those operations to parallel processing units, merging the results after processing.

What's the benefit of the dataflow model? In one word, simplicity. Programming parallel systems is notoriously difficult. Many mechanisms for coordinating the actions between parallel computations have been invented and implemented: mutexes, semaphores, monitors, barriers, condition variables, and so forth. Programs written using these mechanisms often exhibit subtle bugs. Moreover, finding such bugs is often painstaking and difficult, because they often cannot be repeated: they might arise only during certain schedules of the threads, or under certain loads. The dataflow model helps programmers avoid such bugs. First, it makes the parallelism explicit—each node can run as a separate thread—and makes it simple to control. Second, data gets passed between nodes only when the nodes are connected. There's no shared memory to protect and manage.

The Aleri Streaming Platform has one advantage over similar CEP systems: it takes the dataflow model of computation more seriously than the database model. Many other systems force the programmer to express the computational nodes with SQL primitives. Sometimes, this is natural, but sometimes it's just not flexible enough. The Platform therefore includes a special type of node, the FlexStream. These nodes can be programmed by specific event handlers, using a simple procedural language that looks like C or PL/SQL. And indeed, experience has shown that many of the real-world examples need precisely that flexibility. We'll explore the syntax and use of FlexStreams in another article.

With multi-core laptops and desktops becoming the norm, and multiprocessor server machines becoming ever cheaper, it's important to take advantage of the computing horsepower. The dataflow model, and its instantiation in the Aleri Streaming Platform, helps application developers take advantage of that horsepower, with elegance and simplicity.