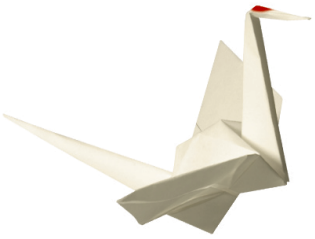
Information Management software

IBM

IBM InfoSphere Streams

Redefining Real Time Analytic Processing

Roger Rea
IBM Software Group



Contents

1. Introduction
2. Stream Computing
3. Selected Use Cases
4. Architectural Overview
5. What's new in Streams 2.0?
6. Summary

Executive Summary

Moore's Law, which describes the long term trend of doubling computing power every two years, powers two separate technology waves. Multicore chips used in servers continue to accelerate the performance in servers. Embedded chips in sensor devices, smart phones and tablets are doubling the volume of data every two years. These two technology trends make it harder for developers to deliver business solutions. They can't keep up with data volumes. They can't easily make use of multiple cores with multithreading (the so called many core challenge). Yet, global competition drives organizations to seek greater agility for operations and decision making. In the face of exploding data volumes and shrinking decision windows, these organizations are struggling to make 'truly' real time decisions and gain competitive advantage. Existing tools and technologies that aid decision making by the Line of Business first require data to be recorded on a storage device and run analytic queries after the fact to detect actionable insights. Savvy businesses are fast realizing that the time lost in this process leads to missed opportunities that might be the difference between success and failure.

InfoSphere Streams addresses this gap effectively by providing developer tools and execution platforms that can detect insights from real time data streams, before data are saved into databases.

Introduction

The goal for IBM InfoSphere Streams is to deliver breakthrough capabilities enabling aggressive analysis and management of information and knowledge from relevant data, extracted from huge volumes and varieties of potentially unimportant data. Specifically, InfoSphere Streams radically extends the state of the art in information processing by simultaneously addressing several technical challenges:

- Respond in real time to events and changing requirements
- Continuously analyze data at rates that are orders of magnitude greater than existing systems
- Adapt rapidly to changing data forms and types
- Manage high availability, heterogeneity, and distribution for the new stream paradigm
- Provide security and information confidentiality for shared information

While certain research, open source and commercial initiatives try to address these technical challenges in isolation, no program – outside of InfoSphere Streams – attempts to simultaneously address all of them. InfoSphere Streams breaks through a number of fundamental barriers to meet these challenges. The project, which began as collaboration between the United States Government and IBM in 2003, has been implemented by many organizations to build a variety of application in industries like Government, Telecommunications, Financial Markets, Energy, e-Science and Healthcare.

Stream Computing

Stream computing is a new paradigm. In "traditional" processing, one can think of running analytic queries against historic data: for instance – calculate the distance walked last month from a data set of subscribers who transmit Global Positioning System (GPS) location data while walking. With stream computing, one can execute a process similar to a "continuous query" that keeps running totals, as location information from GPS data is refreshed moment by moment. In the first case, questions are asked of historic data, in the second case, data is continuously evaluated by static questions. InfoSphere Streams goes further by allowing the continuous analysis to be modified over time.

A simple view of this distinction is shown below:

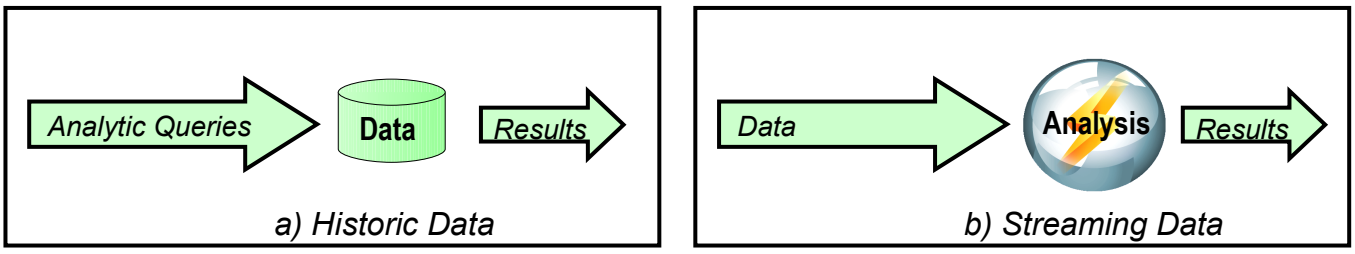


Figure 1: Historic data versus streaming data: conceptual overview.

While there are other systems that embrace the stream computing paradigm, InfoSphere Streams takes a fundamentally different approach for continuous processing and differentiates with its distributed runtime platform, programming model, and tools for developing continuous analytic applications. The data streams consumable by InfoSphere Streams can originate from sensors, cameras, news feeds, stock tickers, or a variety of other sources, including traditional databases.

Selected Use Cases

Over the past several years, hundreds of applications have been developed for InfoSphere Streams. The following provides a summary of a few applications, highlighting the types of usage supported by InfoSphere Streams.

Telecommunications: The challenge of closing the technology and business gaps has been especially apparent for cellular service providers in Asia. Embedded chips in cell phones enable email, texting, pictures, videos and information sharing using social sites like Facebook. For each phone call, email, web browse or text message, cellular phone switches emit call detail records (CDRs). To ensure no data loss, the switches emit two CDRs for each transaction, which must later be de-duplicated for billing support systems. Rising volume caused mediation of CDRs to become ever more difficult to perform in a timely manner. Number portability enabled subscribers to move to a competitor at any time, with some sophisticated users even switching between multiple providers at different times of each day to take advantage of promotions. Providers not only needed to reduce the window of processing CDRs to near real time, but perform analytics at the same time to predict which customers might leave for a competitor. With this real time insight into customer behavior, providers could take action to retain a higher percentage of customers.

InfoSphere Streams with its agile programming model has enabled customers to handle their huge volume of CDRs with very low latency, while analyzing the data at the same time. At one company a peak rate of 100,000 CDRs per second is being processed, with end to end latency of under 1 second. Each CDR is checked against billions of existing CDRs with duplicates eliminated in real time, effectively cutting in half the amount of data stored in databases. This illustrates a key Streams use case – simultaneous processing, filtering and analysis in real time. High Availability, automated fault tolerance and recovery along with real time dashboard summaries are in use, improving IT operations. Real time analysis of CDRs is expected soon, leading to improved business operations.

Government: A key strength of InfoSphere Streams lies in its ability to perform analytics on data-intensive streams to identify the few items that merit deeper investigation. One example of this use case is in the

domain of cyber security. A Botnet is a network of software agents, or robots, that run autonomously and automatically. Botnets respond to Command and Control (C&C) machines, where an underground economy has sprung up to provide per pay access to the Botnets for criminal activity. These Botnets are evolving rapidly to make it vary difficult to detect the bots due to fast fluxing networks and encryption. The Shadowserver Foundation tracks known bots, and estimates there are thousands of C&C machines, and tens of thousands of bots. InfoSphere Streams was used to analyze over 100 megabytes per second of IP traffic and over 10 million domain name server (DNS) queries per hour to generate fast fluxing Botnet alerts. Streams not only uses machine learning models, but also models created using historic data in IBM InfoSphere Warehouse for Botnet alerts. SPSS Modeler is used create these historic models. Streams also monitors for model drift, and when the attack patterns are changing, and issue requests to have models updating against the historic data to ensure up to date detection models. This use case shows continuous updating of mining models for both historic and real time analysis of data.

Financial Services: Many segments of the financial services industry rely on rapidly analyzing large volumes of data in order to make near-real time business and trading decisions. Today these organizations routinely consume market data at rates exceeding one million messages per second, twice the peak rates they experienced only a year ago. This dramatic growth in market data is expected to continue for the foreseeable future, outpacing the capabilities of many current technologies. Industry leaders are extending and refining their strategies by including other types of data in their automated analysis; sources range from advanced weather prediction models to broadcast news. IBM developed an InfoSphere Streams based trading prototype running on eleven x86 blade computers that could host scalable trading applications capable of processing OPRA data feeds sped up 21 times the recorded rate.

Health monitoring: Stream computing can be used to better perform medical analysis with reduced workload on nurses and doctors. Privacy-protected streams of medical device data can be analyzed to detect early signs of disease, correlations among multiple patients, and efficacy of treatments. There is a strong emphasis on data provenance in this domain, in tracking how data are derived as they flow through the system. The "First of a Kind" collaboration between IBM and the University of Ontario Institute of Technology uses InfoSphere Streams to monitor premature babies in a neonatal unit. Data has been collected for over a year and a half from a hospital in Toronto, Canada. Remote telemetry from a US hospital has been operational for a year using the same analytic routines. And earlier this year, additional hospitals in China and Australia began implementation.

Transportation: IBM is working on an application in the IBM Smarter Cities Technology Centre in Dublin, Ireland, where InfoSphere Streams receives GPS data once per minute from buses. A real time display shows all buses as they move through the city. The pilot is working to extend this with realtime predictions about arrival times for each bus for each bus stop. This will enable bus riders to better plan when to arrive at the bus stops, reducing waiting times. In the future, a personal travel planner could allow riders to receive recommendations based on real time traffic monitoring.

In addition, other use cases of InfoSphere Streams are fast emerging in domains such as environmental monitoring and control (wildfire detection, water flow monitoring etc), energy and utilities industry (syncrhophasor monitoring of smart grids, prediction of wind power generation), radio astronomy, x-ray diffraction using synchrotrons, fraud prevention etc.,

Architectural Overview

The InfoSphere Streams architecture represents a significant change in computing system organization and capability. While it has some similarity to Complex Event Processing (CEP) systems, it supports higher data rates and a more data types. It also provides infrastructure support to address the needs for scalability and dynamic adaptability, like scheduling, load balancing, and high availability.

In InfoSphere Streams continuous applications are composed of individual operators, which interconnect and operate on one or more data streams. Data streams normally come from outside the system or can be produced internally as part of an application. The following flow graph shows how multiple sources and varieties of streaming data can be filtered, classified, transformed, correlated, and/or fused to make equities trade decisions, using dynamic earnings calculations, adjusted according to earnings-related news analyses, and real-time risk assessments such as the impact of impending hurricane damage:

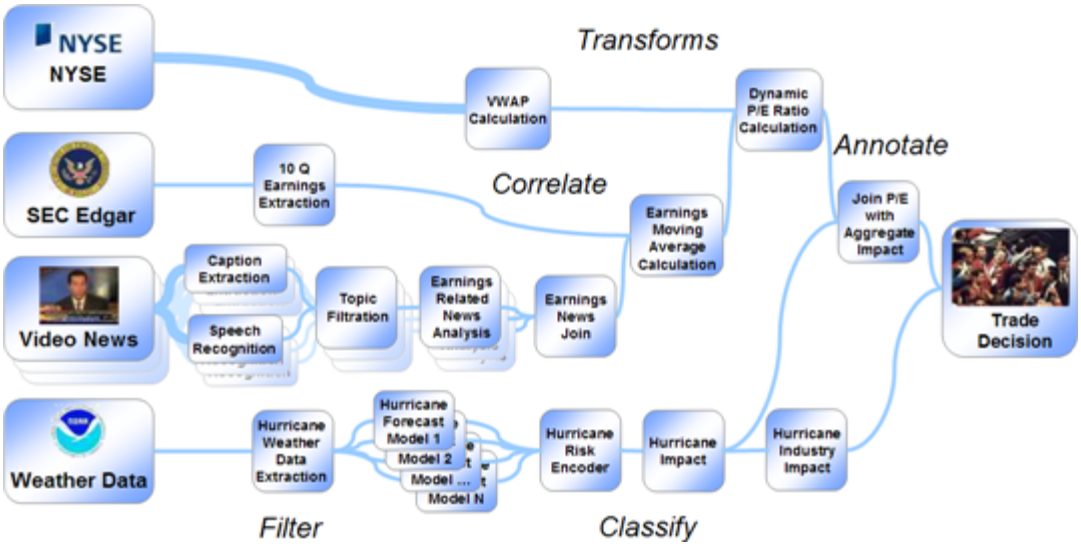


Figure 2: Trading Example.

For the purposes of this overview it is not necessary to understand the specifics of Figure 2, rather, its purpose is to demonstrate how streaming data sources from outside InfoSphere Streams can make their way into the core of the system, be analyzed in different fashions by different pieces of the application, flow through the system, and produce results. These results can be used in a variety of ways, including display on a dashboard, driving business actions, or storage in enterprise databases or warehouses for further historic analysis.

A college professor once filled a large glass jar with rocks, and then asked students if the jar was full. Most replied yes, it was full. The professor proceeded to add small pebbles, and after shaking the jar was able to add a significant number of pebbles. The professor again asked if the jar was full. Having learned, most students replied no. The professor next added sand, and after again shaking the jar had sand filling all the small openings up to the top of the jar. Again, the professor asked, "Is the jar full?" Several students answered that yes, the jar was full; thinking no more rocks, pebbles or sand could be added. The professor next pulled out a pitcher of water and poured it into the jar. The professor explained that the jar was analogous to life and that if people didn't fill their life with important things first, just like the big rocks, they would never find time to add them later.

But the story illustrates another principle. Using the smallest of building blocks, there is less wasted space. Water molecules easily fill nearly invisible spaces in the sand filled jar. Streams uses this principle to optimize performance and latency. Very fine grained operators are used in streaming applications, which are then fused together into Processing Elements for deployment onto Streams for execution. An advanced compiler converts the high level declarative Streams Processing Language (SPL) into

machine language, ready for execution. The compiler also detects stateless and stateful operators, enabling the use of multiple threads on multicore computers. This advanced capability not only facilitates developer agility to solve the many core programming challenge, but enables outstanding performance and very low latency processing.

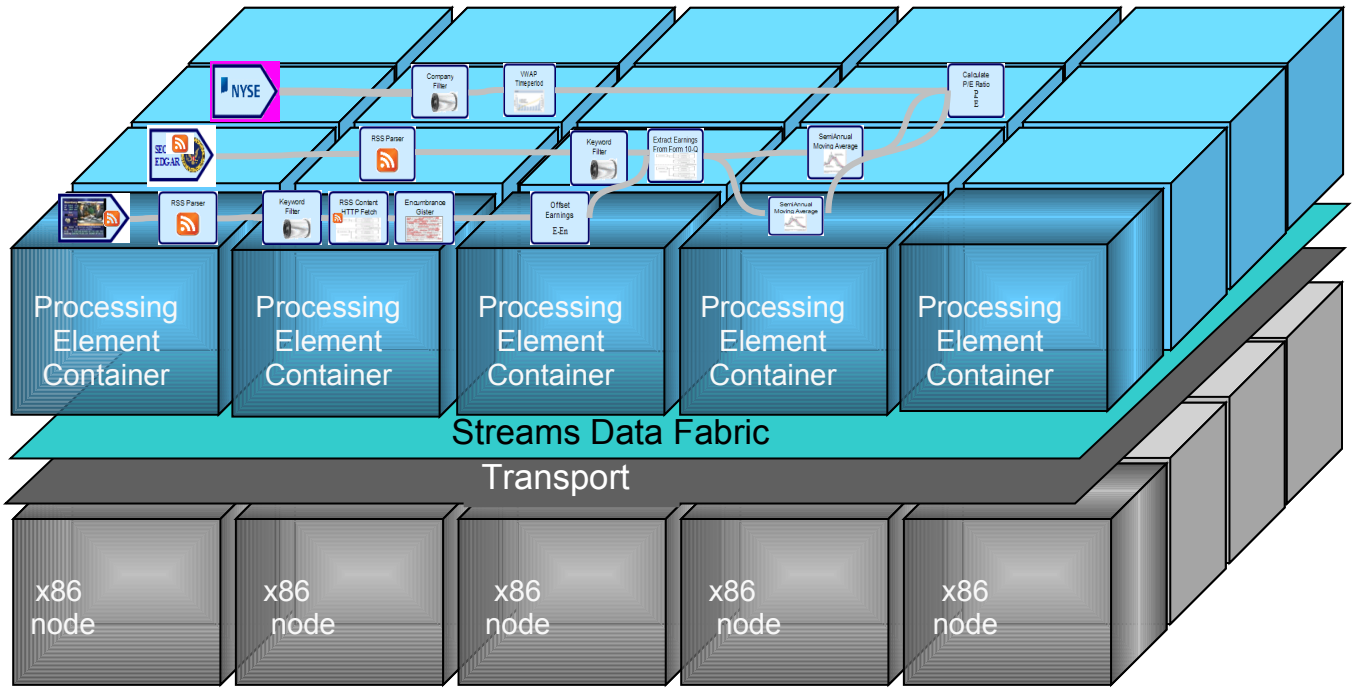


Figure 3: System overview.

Figure 3 illustrates the InfoSphere Streams infrastructure, which supports very large clusters of x86 architecture computers. As shown, data from input data streams representing a myriad of data types and modalities flow into the system. Management services communicate using the Streams Data Fabric over a physical transport layer. The physical transport can be high speed Ethernet, such as 10G Ethernet, or even faster InfiniBand transport. Streams management services continually monitors performance of each operator, processing element, job and node on the runtime to optimize deployment of jobs. This information is especially important for the development phase of applications when saved performance information is used by the compiler to optimize operator fusion.

Figure 3 also shows that multiple jobs can be deployed to Streams. These new jobs can be added dynamically, and import or export data streams between the jobs. This flexibility creates tremendous agility for businesses as new jobs can be added to analyze data in new ways to existing applications. To meet growing capacities, new input streams, output streams and nodes can be dynamically added or removed from the Streams runtime without restarting the system.

InfoSphere Streams offers multiple methods for end-users to operate on streaming data, as follows:

- The Stream Processing Language provides a language that works with the Streams run-time framework to support streaming applications. Users can create applications without needing to understand the lower-level stream-specific operations. The Stream Processing Language provides many built-in operators, the ability to bring streams from outside InfoSphere Streams and export results outside the system, and a facility to extend the underlying system with user-defined operators.

- Users can develop applications using Streams Studio, an Eclipse-based Integrated Development Environment (IDE). These users can program low-level application components that can be interconnected via streams, and specify the nature of those connections. Each component is "typed" so that other components can later reuse or create a particular stream. Existing analytic routines, developed in Java or C++ can be incorporated into these Streams applications to potentially shorten development time, and reuse existing algorithms.

- Users can also deploy existing data mining scoring models in Streams applications for real time insights as opposed to running those models on persistent, or stored, data. In addition, Streams applications can detect the need for a new model, trigger the generation of an updated model and deploy the new model in a running application, as highlighted in the cyber security use case.

All these features are supported by the underlying runtime system. As new jobs are submitted, the InfoSphere Streams scheduler determines how to best meet the requirements of both newly submitted and already executing specifications, and the Job Manager automatically effects the changes required. The runtime continually monitors and adapts to the state and utilization of its computing resources, as well as the information needs expressed by the users and the availability of data to meet those needs.

Results that come from the running applications are acted upon by processes (such as web servers) that run external to InfoSphere Streams. For example, an application might use TCP connections to receive an ongoing stream of data to visualize on a map, or it might alert an administrator to anomalous or "interesting" events.

What's new in Streams Version 2.0

IBM InfoSphere Streams V2.0 delivers a range of new features and functions to speed development of applications to enhance performance, availability, and administration of Streams, by delivering a range of new features and functions described below.

Simplified development of streaming applications

IBM InfoSphere Streams Studio is an Eclipse-ready tool that enables you to create, edit, visualize, test, debug, and run Streams Processing Language (SPL) and SPL mixed-mode applications. The integrated development environment (IDE) of Streams Studio consists of the following major features:

- Streams Explorer view that helps you set up and manage your Streams development environment.
- SPL Project and SPL Application Set Project support for organizing and building SPL applications and toolkits.
- SPL editor with auto-completion, code templates, code-folding, continuous build, in-line error reporting, refactoring, and outline view:
- Split-view SPL mixed-mode editor
- Toolkit model, operator model, and function model editors.
- Streams Explorer that can manage runtime instances.
- Visualizer with expand/collapse support for composite operators.
- Project Explorer view to visualize your SPL resources in a logical manner.
- Graph view that displays topology of application.
- Metrics view that allows you to view and analyze metrics from running applications.
- Log viewing support that allows you to gather and examine logs from a running instance.
- Launchers for running stand-alone and distributed applications.
- Debugger for testing and debugging Streams applications.
- Integrated Help system that provides information needed during the development of SPL applications and toolkits.

InfoSphere Streams V2.0 includes a wide array of enhancements to significantly increase business agility through improved developer productivity. The newly improved Streams Processing Language (SPL) delivers major improvements, including:

- Greater consistency in syntax, APIs, and configurations
- A type system that allows hierarchical representation of the data items
- Nested types that allow hierarchical representation of data
- Easy-to-use extension mechanisms for writing primitive operators and native functions
- Composite operators for addressing large-scale development

- Support for updating behavior of operators on-the-fly
- A common windowing library implementation which is available to operator developers
- Common container types, such as lists, sets, and maps
- Composition-level constructs for creating multithreaded flows
- Modularity support, via namespaces and multiple files
- Versioned toolkit support for operators and functions
- A new standard toolkit, with new relational, adapter, and utility operators
- Improved compile-time performance (compile-time folding and function evaluation minimize code generation and even improve performance)

Improved performance enables handling higher volumes at increased velocity

InfoSphere Streams runtime ingests both structured and unstructured information at high rates, and analyzes the information to provide business insights. Portions of Streams programs are distributed across one or more nodes of the runtime computing cluster to achieve volumes in the millions of messages per second with velocities of well under a millisecond. Streams V2.0 continues to improve performance to help companies deal with increased volume of information and need to increase velocity through:

- New multi multithreading support makes better use of advanced multithreaded hardware
- Improved Java-support allows for shared Java Virtual Machines to improve memory usage
- Improved intranode and internode communications improve date speed through Streams Runtime (operator implementations, runtime fusion, transport integration)

Improved systems administration lowers administrative costs and helps improve runtime capacity

InfoSphere Streams offers both command line and graphical interfaces to administer the Streams runtime and maintain optimal performance and availability of applications. The web-based administrative console provides ability to create, start, and stop instances on a set of nodes in a cluster, and can cancel applications on these instances. Many new and improved capabilities can simplify administrator workloads as they strive to maintain optimal usage of the Streams runtime.

- New health metrics available programmatically to facilitate application performance and availability improvements
- More granular health metrics and operator capabilities, down to the Streams operator level instead of at the Processing Element level
- Submission time values to customize applications without recompilation
- Instance agnostic and relocatable applications to help improve runtime capacity
- Dynamic allocation of host pools to add agility to the runtime platform
- New ex-location and isolation constraints via host tagging to simplify job deployment
- Runtime APIs for programmatic manipulation of import subscriptions

- Runtime and code-generation APIs for generic windowing manipulations
- Streams Administrative console with new operator-level information

High availability features

InfoSphere Streams has several features that enhance high availability and redundancy. Administrators can add or remove processor nodes to or from the cluster. The ability to seamlessly add or remove nodes to or from the cluster lets administrators perform necessary maintenance on the operating system or on data streams without shutting down the InfoSphere Streams application. This can help improve overall availability of the environment. Also, when the management nodes (that control the system) fail, the application nodes will continue to execute the InfoSphere Streams applications and the management nodes can be restarted by the administrators at a later time. The new programmatic manipulation of streaming data makes it easier to create highly available applications using redundant application components. New dynamic allocation of host pools and additional location and isolation constraints of nodes in the runtime cluster can be used to isolate redundant application components to separate nodes in the runtime. Features such as these are aimed at enhancing high availability aspects related to InfoSphere Streams.

Summary

In the short time since its announcement, InfoSphere Streams has demonstrated far reaching successes with over 150 customer installations. InfoSphere Streams provides an infrastructure that supports mission-critical data analysis with exceptional performance. IBM continues to heavily invest in this growth area and make this technology more powerful, scalable, secure, relevant and interoperable with existing Information infrastructures.

For more information about InfoSphere Streams or InfoSphere Streams, please contact your IBM Marketing Representative or Authorized IBM Business Partner.

For more information

To learn more about IBM InfoSphere Streams and associated products to build them, visit:

http://www.ibm.com/software/data/infosphere/streams/

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
04-11