

IBM Research Report

System S - Stream Computing at IBM Research

Stream Computing Research Team
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598



System S – Stream Computing at IBM Research



Highlights

- *Real-time, adaptive processing of enormous volumes of continuously streaming data — both structured and unstructured*
- *Automatic reconfiguration of system resources in response to changing data streams and user objectives*
- *Multiple programming models to suit the needs of a variety of developer and user communities*

The goal of the IBM Research System S Stream Computing System is to provide breakthrough technologies that enable aggressive production and management of information and knowledge from relevant data, which must be extracted from enormous volumes of potentially unimportant data. Specifically, the goal of **System S** is to radically extend the state of the art in information processing by simultaneously addressing several technical challenges, including:

- Responding quickly to events and changing requirements
- Continuously analyzing data at rates that are orders of magnitude greater than existing systems
- Adapting to rapidly changing data forms and types
- Managing high availability, heterogeneity, and distribution for the new stream paradigm
- Providing security and information confidentiality for shared information

While certain research and commercial initiatives endeavor to address the above technical challenges in isolation, no program – outside of **System S** – attempts to simultaneously address all.

The primary goal of **System S** is to break through a number of fundamental barriers to enable the creation of a system designed to meet these challenges. The project, which began in IBM Research in 2003, has now reached a level of maturity that has permitted it to be demonstrated in a variety of application environments and to embark on a path toward IBM productization.

This document provides an overview of the stream processing paradigm, surveys several of the applications on which **System S** currently runs, and describes some of the underlying technologies. Additional details will soon be available in a companion document.

Note that the capabilities of any product *derived* from IBM Research **System S** would differ from the capabilities described in this paper.

Stream Computing

Stream computing is a new paradigm. In “traditional” processing, one can think of running queries against relatively static data: for instance, “list all personnel residing within 50 miles of New Orleans,” which will result in a single result set. With stream computing, one can execute a process similar to a “continuous query” that identifies personnel who are currently within 50 miles of New Orleans, but get continuous, updated results as location information from GPS data is refreshed over time. In the first case, questions are asked of static data, in the second case, data is continuously evaluated by static questions. System S goes further by allowing the continuous queries to be modified over time. A simple view of this distinction is reflected in figure 1.

While there are other systems that embrace the

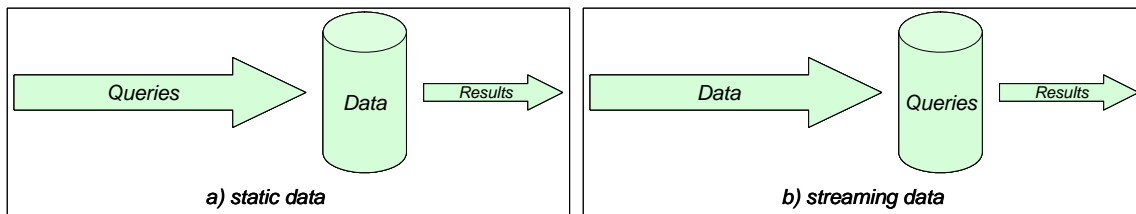


Figure 1 Static data vs. streaming data: conceptual overview

stream computing paradigm, System S takes a fundamentally different approach for continuous processing and differentiates with its distributed runtime platform, programming model, and tools for developing continuous processing applications. The data streams consumable by System S can originate from sensors, cameras, news feeds, stock tickers, or a variety of other sources, including traditional databases.

System S Pilot Studies

As System S is readied for deployment, a number of applications are being pursued. The following provides a summary of the pilots conducted by IBM, highlighting the types of usage that can be supported by System S.

Anomaly detection—One of **System S**’s key strengths is the 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 astronomy. There are a number of projects globally that receive continuous streams of telemetry from radio telescopes. For example, these radio telescopes might have thousands or tens of thousands of antennae, all routing data streams to a central supercomputer to survey a location in the universe. The **System S** middleware running on that supercomputer can provide a more flexible approach to processing these streams of data. We are working with the low frequency radio astronomy group of Uppsala University and the LOFAR Outrigger In Scandinavia (LOIS¹) project to develop analytics that identify anomalous and transient behavior such as high energy cosmic ray bursts. We are investigating expansion of this work to a similar effort with the Square Kilometre Array², with total data

rates in the range of terabits per second.

Energy Trading Services (ETS)—The ETS pilot was developed to demonstrate to an investment bank how System S can support energy trading. The demonstrated system provides energy traders with real-time analysis and correlation of events affecting energy markets, and allows them to make

¹ <http://www.lois-space.net/index.html>

² http://en.wikipedia.org/wiki/Square_Kilometre_Array

informed decisions faster than before. Analyses supporting energy traders include various heat maps, energy demand models, technical analyses of energy futures (Bollinger Band VWAP, etc), news feed analysis to identify and evaluate energy-relevant events, and a map view of the predicted impact of a hurricane on the assets of oil companies. The traders can leverage shared computing infrastructure to obtain information quickly and at a low cost. The system also provides context-sensitive guidance that helps the traders select the best available sources and analytics for the task.

The pilot uses MARIO (Mashup Automation with Runtime Invocation and Orchestration) to dynamically assemble applications needed by energy traders, deploying and operating the stream processing parts of these applications in a System S cluster. The set of 250 independent analytics, data sources and configuration descriptions that were built for the ETS pilot are dynamically composed and parameterized in different combinations to create thousands of applications that analyze and present data relevant to energy trading. The demonstrated applications analyze real-time and/or previously recorded data obtained from external sources such as NOAA and NYMEX. .

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/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 and TD Bank Financial Group are jointly exploring the application of System S and a Blue Gene supercomputer in a project to define a platform for streaming Financial Services applications that will continue to scale with industry growth for years to come.

Health monitoring–Stream computing can be used to perform better medical analysis with reduced workload on doctors. Privacy-protected streams of medical monitoring 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. A “First of a Kind” collaboration between IBM and the University of Ontario Institute of Technology will use System S to monitor premature babies in a neonatal unit.³

Manufacturing–We are conducting a pilot with IBM’s Burlington semiconductor chip fabrication line, in which **System S** performs multivariate monitoring for real-time process fault detection and classification. In this fashion, when process errors cause defects in manufactured chips, these errors can be detected within minutes rather than days or weeks. The defective wafers can then be potentially reworked prior to ensuing process steps which might render the wafers unusable, and more importantly, adjustments can be made before processing subsequent wafers. This pilot is in the prototyping stage.

Architectural Overview

The **System S** architecture represents a significant change in computing system organization and capability. It has some similarity to CEP systems, but it is built to support higher data rates and a broader spectrum of input data modalities. It also provides significant infrastructure support to address needs for scalability and dynamic adaptability such as scheduling, load balancing, and high availability.

In **System S**, continuous applications are composed of individual operators, which interconnect and operate on multiple data streams; that is, data streams can come from outside the system or be produced internally as part of an application. As an example, the flow diagram in figure 2 shows how multiple sources of varying types of streaming data can be filtered, classified, transformed, and correlated to inform 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

For the purposes of this overview it is not necessary to understand the specifics of Figure 2. Rather, its purpose is to

³ <http://biz.yahoo.com/iw/080723/0418488.html>

demonstrate how streaming data sources from outside **System S** can make their way into the core of the system, be analyzed in different fashions by different pieces of the

export results outside the system, and a facility to extend the underlying system with user-defined operators.

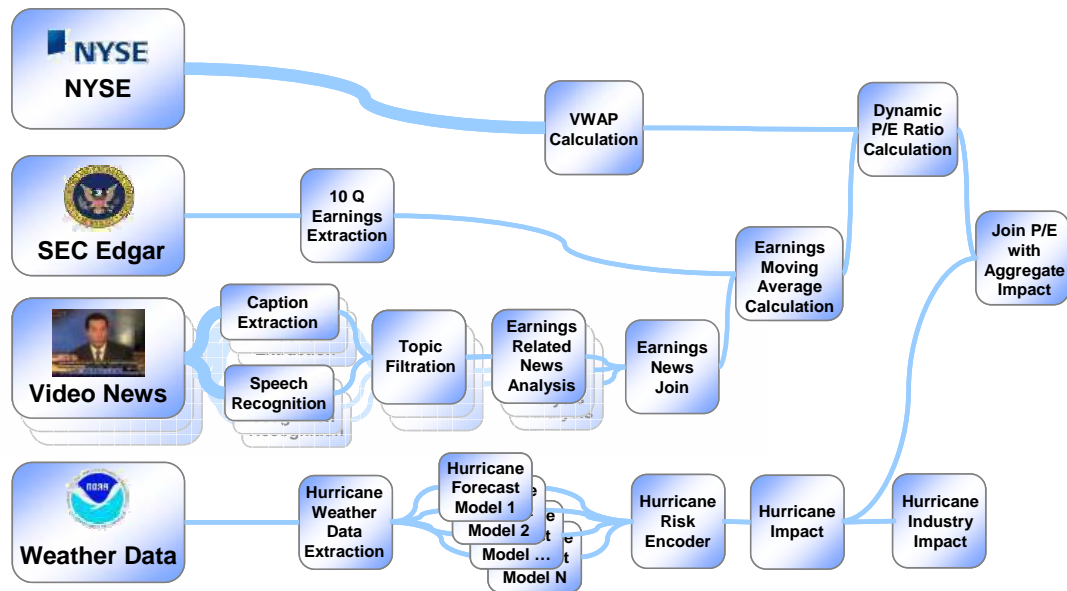


Figure 2 - Trading Example

application, flow through the system, and produce results. These results can be used in a variety of ways, including display within a dashboard, driving business actions, or storage in enterprise databases for further offline analysis.

Figure 3 illustrates the complete prototype infrastructure. As shown, data from input streams representing a myriad of data types and modalities flow into the system. The layout of the operations performed on that streaming data is determined by high-level system components that translate user requirements into running applications. System S offers three methods for end-users to operate on streaming data, as follows:

SPADE (Stream Processing Application Declarative Engine)⁴ provides a language and run-time framework to support streaming applications. Users can create applications without needing to understand the lower-level stream-specific operations. SPADE provides some built-in operators, the ability to bring streams from outside **System S** and

MARIO (Mashup Automation with Runtime Invocation & Orchestration)⁵ Users may pose *inquires* to the system to express their information needs and interests. These inquiries are translated by a *Semantic Solver* into a specification of how potentially available raw data and existing information can be transformed to satisfy user objectives. The runtime environment accepts these specifications, considers the library of available application components, and assembles a job specification to run the required set of components.

Workflow Development Tool Environment, enabling users to develop components and applications using 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. This development model will evolve over time to directly operate on SPADE operators rather than the base, low-

⁴ <http://portal.acm.org/citation.cfm?id=1376616.1376729>

⁵ <http://portal.acm.org/citation.cfm?id=1367497.1367602>

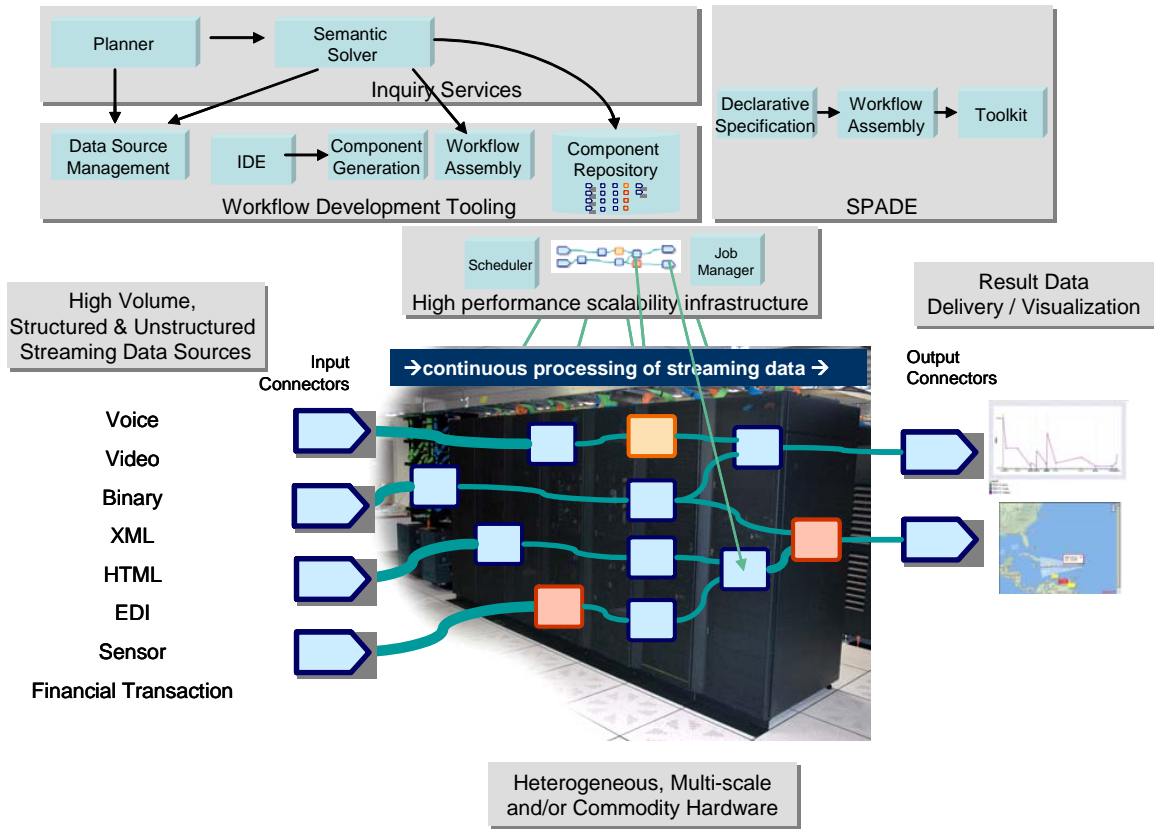


Figure 3 - System Overview

level applications components, but will still allow new operators to be developed.

All three of these methods are supported by the underlying runtime system. As new jobs are submitted, the **System S Scheduler** determines how it might reorganize the system in order 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 **System S**. 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.

Summary

In over five years since **System S** first began as an IBM research project, it has demonstrated

initial successes with a number of commercial and scientific applications. It provides an infrastructure to support mission-critical data analysis with exceptional performance and interoperability with existing application infrastructures. The anticipated adoption of technologies from IBM Research **System S Stream Computing System** into the IBM Software Group product line is expected to further increase the scale and diversity of its infrastructure, tools, support, and potential applications.

For more information about **System S**, please contact Nagui Halim, halim@us.ibm.com.



© Copyright IBM Corporation 2008

IBM Corporation
New Orchard Road
Armonk, NY 10504
U.S.A.

Produced in the United States of America

10-01

All Rights Reserved

IBM, the IBM logo, Blue Gene and Cell
Broadband Engine are trademarks of
International Business Machines Corporation in
the United States, other countries or both.

Other company, product or service names may
be trademarks or service marks of others.

References in this publication to IBM products or
services do not imply that IBM intends to make
them available in all countries in which IBM
operates.