Agenda

- Introductions
- Why stream computing matters
- Overview of InfoSphere Streams
- Topics for future meetups
- Questions
Sponsors

PROTEUS TECHNOLOGIES

IBM
Presenters

Andrew Skene - Senior Member of Technical Staff at Proteus. 6+ years of stream computing experience

Brian Williams - IT Architect with IBM Software Group Services. Many years of experience with IBM InfoSphere Streams
Maryland Stream Computing

• Goals of the meetup group
  o Establish a local community around Stream computing
  o Eat pizza and talk
Stream computing

Stream computing VS Map Reduce

- More data than can be stored on disk
- Real-time computing need
- Correlate multiple data sources in real-time

- “Traditional” form of data mining utilizing large amounts of historic data
- Modeling based on historic data
Stream computing frameworks

- Infosphere Streams
- Apache Spark Streaming
- Amazon Kinesis
- Apache Storm
- AKKA
IBM InfoSphere Streams: Quick Start Edition

- **What is Streams Quick Start?**
  - No charge, downloadable edition to allow you to experiment with stream computing
  - No time or data limitations for use on your unique use cases in non-production systems
  - Sophisticated analytics for streaming data - quickly ingest, analyze and correlate data
  - Comprehensive development tools and scale-out architecture to get up and running quickly, support available through forums & communities**

STREAMSDEV: http://www.ibmdw.net/streamsdev

**IBM InfoSphere Streams Quick Start**

Download Now!  Download Link

Tutorials!  Videos and labs

** no formal IBM support is available
Big Data: Real-time Data Processing & Analytics

Big Data & Analytics Strategy, Integration & Managed Services

Big Data & Analytics Platform

Real-time Data Processing & Analytics

Operational data zone

Landing, Exploration and Archive data zone

Deep Analytics data zone

EDW and data mart zone

Information Integration & Governance

Big Data & Analytics Infrastructure

New / Enhanced Applications

Outcome-based Program Mgt.
Risk Determination
Case Management
Real-time Fraud Detection
Real-time Threat & Crime Detection
Patrol Deployment
Audit & Tax Compliance
Budget & Finance Optimization

All Data

Claims
Tax & Income
Threat & Crime
Case Worker
Social Media
Sensor
Images & Video

What is happening?
Discovery and exploration

What action should I take?
Decision management

What did I learn, what’s best?
Cognitive

Why did it happen?
Reporting and analysis

What could happen?
Predictive analytics and modeling

What is happening?
Discovery and exploration
**Big Data: Real-time Data Processing & Analytics**

**New / Enhanced Applications**

- Outcome-based Program Mgt.
- Risk Determination
- Case Management
- Real-time Fraud Detection
- Real-time Threat & Crime Detection
- Patrol Deployment
- Audit & Tax Compliance
- Budget & Finance Optimization
Big Data need - Analyze Streaming Data

- **Customer Need**
  - Harness and process live data sources
  - Quickly process and analyze perishable data, and take timely action
  - Select most valuable data and insights to be stored for further processing

- **Value Statement**
  - Significantly reduced processing time and cost – process and then store what’s valuable
  - React in real-time to capture opportunities before they expire

- **Customer examples**
  - Ufone – Telco Call Detail Record (CDR) analytics for customer churn prevention
Stream Computing

Traditional Computing

- Historical fact finding
- Find and analyze information stored on disk
- Batch paradigm, pull model
- Query-driven: submit queries to static data

Stream Computing

- Current fact finding
- Analyze data in motion – before it is stored
- Low latency paradigm, push model
- Data driven – bring data to the analytics
What Is Stream Computing

Continuous Ingestion

Continuous Analysis in Microseconds
InfoSphere Streams - Massively Scalable Stream Analytics

Linear Scalability
• Clustered deployments – unlimited scalability

Automated Deployment
• Automatically optimize operator deployment across nodes

Performance Optimization
• Parallel & pipeline operations
• Efficient multi-threading

Analytics on Streaming Data
• Analytic accelerators for a variety of data types
• Optimized for real-time performance
How is Streams Being Used?

**Telephony**
- CDR processing
- Social analysis
- Churn prediction
- Geomapping

**Transportation**
- Intelligent traffic management
- Automotive Telematics

**Energy & Utilities**
- Transactive control
- Phasor Monitoring Unit
- Down hole sensor monitoring

**Health & Life Sciences**
- ICU monitoring
- Epidemic early warning system
- Remote healthcare monitoring

**Law Enforcement, Defense & Cyber Security**
- Real-time multimodal surveillance
- Situational awareness
- Cyber security detection

**Stock market**
- Impact of weather on securities prices
- Analyze market data at ultra-low latencies
- Momentum Calculator

**Fraud prevention**
- Detecting multi-party fraud
- Real time fraud prevention

**e-Science**
- Space weather prediction
- Detection of transient events
- Synchrotron atomic research
- Genomic Research

**Natural Systems**
- Wildfire management
- Water management

**Other**
- Manufacturing
- Text Analysis
- ERP for Commodities

**Telephony**
- CDR processing
- Social analysis
- Churn prediction
- Geomapping
Predictive Analytics in Neonatal ICU

- **Real-time analytics and correlations on physiological data streams**
  - Blood pressure, Temperature, EKG, Blood oxygen saturation etc.,
- **Early detection of the onset of potentially life-threatening conditions**
  - Up to 24 hours earlier than current medical practices
  - Early intervention leads to lower patient morbidity and better long term outcomes
- **Technology also enables physicians to verify new clinical hypotheses**
TerraEchos – Covert Intrusion Detection

- State-of-the-art covert surveillance system based on Streams platform
- Acoustic signals from buried fiber optic cables are monitored, analyzed and reported in real time for necessary action
- Currently designed to scale up to 1600 streams of raw binary data
- Dr. Alex Philp, TerraEchos Inc.
How does Streams Work?

Stream of records (messages, events, tuples) flowing through an operator producing a new stream of records (messages, events, tuples)

In-Memory Analytics
From Essential Elements to Running Jobs

- Streams application graph:
  - A directed, possibly cyclic, graph
  - A collection of sources, operators, and sinks (the graph’s vertices)
  - Connected by streams (the graph’s edges)

- Each complete application is a potentially deployable job
From Essential Elements to Running Jobs

• Streams application graph:
  – A directed, possibly cyclic, graph
  – A collection of sources, operators, and sinks (the graph’s vertices)
  – Connected by streams (the graph’s edges)

• Each complete application is a potentially deployable job

• Jobs are deployed to a Streams runtime environment, known as a Streams Instance (or simply, an instance)
From Essential Elements to Running Jobs

• Streams application graph:
  – A directed, possibly cyclic, graph
  – A collection of sources, operators, and sinks (the graph’s vertices)
  – Connected by streams (the graph’s edges)

• Each complete application is a potentially deployable job

• Jobs are deployed to a Streams runtime environment, known as a Streams Instance (or simply, an instance)

• An instance can include a single processing host (hardware)
From Essential Elements to Running Jobs

- Streams application graph:
  - A directed, possibly cyclic, graph
  - A collection of sources, operators, and sinks (the graph’s vertices)
  - Connected by streams (the graph’s edges)

- Each complete application is a potentially deployable job

- Jobs are deployed to a Streams runtime environment, known as a Streams Instance (or simply, an instance)

- An instance can include a single processing host (hardware)

- Or multiple processing hosts (no coding changes required)
Big Data in Real Time with InfoSphere Streams

→ continuous ingestion  → continuous analysis
Big Data in Real Time with InfoSphere Streams

- Filter / Sample
- Modify
- Annotate
- Analyze
- Fuse
- Classify
- Score
- Windowed Aggregates

→ continuous ingestion
→ continuous analysis
IBM InfoSphere Streams 3.2

Comprehensive Tooling

- Eclipse IDE
- Web console
- Drag & Drop editor
- Instance graph
- Streams visualization
- Streams debugger

Scale-out Architecture

- Clustered runtime for near-limitless capacity
- RHEL v5.5 and above
- CentOS v6.0 and above
- SLES v11.2 and above
- X86 & Power hardware
- InfiniBand support
- Ethernet support

Sophisticated Analytics with Toolkits & Accelerators

- Big Data, CEP, Database, DataStage, Finance, Geospatial, Internet, Messaging, Mining, R, SPSS, Business Rules, Standard, Text & Timeseries toolkits
- Telco & Social Media accelerators
Overview of InfoSphere Streams Programming Model

Streams Processing Language

Input (Source)  Process  Output (Sink)

Cleanse  Filter  Model
Meters  Company Filter  Usage Model
Usage Contract  Text Extract  Season Adjust
Weather Data  RSS Parser  Daily Adjust

Classify  Analyze  Fuse  Act

Temp Action

Persist

Platform optimized compilation
SPL (Streams Processing Language)

- **Stream processing language**
  - Distributed data flow composition language
- **Modular**
  - Ease of assembly and data exchange
- **Procedural support**
  - Custom logic in operators and functions
- **Extensible**
  - User-defined data types
  - Custom operators in SPL
  - User-defined operators in C++/Java
  - Toolkits of operators

```cpp
composite NumberedCat {
  graph
    stream<rstring contents> Lines = FileSource() {
      param format : line;
      file: "catFood.txt";
    }

    stream<rstring contents> Numbered = Functor(Lines) {
      logic state : mutable int32 i = 0;
      onTuple Lines : i++;
      output
        Numbered : contents = (rstring)I + " " + contents;
    }
  }

  () as Sink = FileSink(Numbered) {
    param format : line;
    file: "result.txt";
  }

...}
```
InfoSphere Streams Studio

- **Eclipse-based**
  - Linux and Windows Remote
- **Development Tools**
  - Drag and Drop
    - SPL Graphical Editor
  - SPL Source Code Editor
  - Syntax highlighting
  - Error detection
  - Support for complex application sets
  - Compilation and deployment support
  - Team computing support (e.g. SVN)
- **Runtime Support**
  - Starting and Stopping Instances
  - Starting and Stopping Jobs
- **Monitoring Support**
  - Streams Instance Graph for visualizing running jobs and metrics
  - Live data visualization
Streams Studio - SPL Development

Development Environment Supports:
• Drag and Drop Development
• Palette of operators provided by toolkits
• SPL Text Editor Development
• Round-tripping between the two editors
Streams Studio – Monitor Live Applications
Streams Console – Web GUI

- Web-based GUI used to monitor and manage instances and applications
  - View and administer your instance
  - View and monitor graph of applications
Operators, Toolkits and Accelerators

**Standard Toolkit**

**Relational Operators**
- Filter
- Functor
- Puncctor

**Adapter Operators**
- FileSource
- FileSink
- DirectoryScan
- TCPSource
- TCPSink

**Utility Operators**
- Custom
- Beacon
- Throttle
- Delay
- Barrier
- Pair
- JavaOp
- Parse
- Decompress

**XML Operator**
- XMLParse

**IBM Included Toolkits**
- Database
- Big Data
- Messaging
- Text Analytics
- SPSS
- Time Series
- Financial

**Open-Source Toolkits**
- JSON
- HTTP/REST
- OpenCV
- Accumulo
- HBase

**Big Data Accelerators**
- Telco Event Data Analytics
- Social Data Analytics

**User-Defined Toolkits**

Extend the language by adding user-defined operators, types, and functions (Java and C++)
Differentiating Features

• Capabilities that separate IBM InfoSphere Streams from the competition:
  – Drag and Drop Editor and Monitoring (previously discussed)
  – Dynamic Job Composition and Deployment
  – Deployment Flexibility
  – Runtime Environment – Streams Instance
  – Optimization (Fusion and Transport Selection)
  – User Defined Parallelism (single operator and pipeline)
  – Performance
  – ...and more
Static vs. Dynamic Composition

- Static connections
  - Fully specified at application development-time and do not change at run-time

- Dynamic connections
  - Partially specified at application development-time (Name or Properties)
  - Established at run-time, as new jobs come and go
  - Specifications can also be updated at run-time

- Dynamic application composition
  - Incremental deployment of applications
  - Dynamic adaptation of applications
Static vs. Dynamic Composition

• Static connections
  – Fully specified at application development-time and do not change at run-time

• Dynamic connections
  – Partially specified at application development-time (Name or Properties)
  – Established at run-time, as new jobs come and go
    • Specifications can also be updated at run-time

• Dynamic application composition
  – Incremental deployment of applications
  – Dynamic adaptation of applications
User Defined Parallelism

- Replicates a region of a Streams application to run in parallel
- Automatically splits and merges streams
- Not just multiple threads
- Not just multiple copies of an operator
- Can include:
  - sources and sinks
  - imports and exports
  - multiple parallel regions
  - the whole application
**User Defined Parallelism**

- Flexible Parallel Region Instance Graph views
  - Logical single channel or Physical parallel channels
  - Collapsed or Expanded Composites

![Logical View](image1.png)

![Physical View](image2.png)
IBM InfoSphere Streams: Quick Start Edition

- **What is Streams Quick Start?**
  - No charge, downloadable edition to allow you to **experiment** with **stream computing**
  - No time or data limitations for use on **your unique use cases** in non-production systems
  - **Sophisticated analytics** for streaming data - quickly ingest, analyze and correlate data
  - Comprehensive **development tools** and **scale-out architecture** to get up and running quickly, support available through forums & communities**

STREAMSDEV: http://www.ibmdw.net/streamsdev

**Download Now!**

- **Download Link**

**Tutorials!**

- **Videos and labs**

**IBM InfoSphere Streams Quick Start**

**no formal IBM support is available**
Future Topics
Maryland Streams Computing Group
Future topics

- Live coding of a simple end-to-end Streams application
- Java integration with Streams
- Case studies of existing Streams applications and lessons learned.
Questions?