StreamPy:

Python Stream Processing for Real-Time Analytics

K. Mani Chandy, Julian Bunn, Rahul Bachal, Ker Lee Yap
California Institute of Technology

StreamPy is an open-source platform for developing real-time analytics (RTA) applications by processing streams of data generated by phones, sensors, social media, and monitoring systems. The platform leverages Python's open-source libraries. StreamPy is being developed by a group of students and staff at Caltech.

We prefer to use the phrase "right time analytics" because what matters is getting information to the right places at the right time. By "right time" we mean applications that typically respond in a few seconds or minutes.

Applications of stream processing [1, 2, 3, 4] and RTA have been described extensively in the literature [5, 6, 7] and include:

Examples of real-time analytics (RTA)

- Diagnose problems in machinery before breakdown.
- Identify environmental problems from sensor streams.
- Monitor computer logs to search for cyber attacks.
- Home and office security – intrusion detection, hazards
- Detect supply chain glitches before costly delays.
- Identify marketing opportunities in streams generated by social media and websites.

Challenges to stream processing and RTA

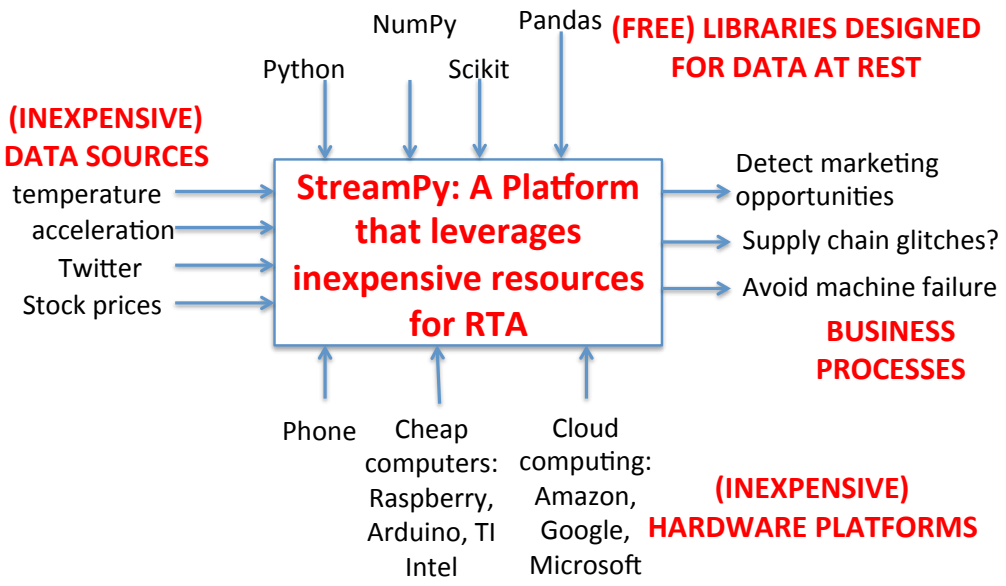
1. Business units have access to powerful Python libraries, but this software operates only on data at rest – snapshots of data. Businesses don't have tools to use this software to obtain RTA from data in motion – data as it is generated.
2. Business units don't have tools that give them a holistic view of their entire stream processing capability – processing in phones, smart sensors, local servers and the cloud.
3. IT staff in business units have deep expertise in their business operations, but they don't have tools that help them use this expertise to build RTA apps.

Enterprises, especially medium and small ones, don't use RTA for three main reasons:

1. Many examples of RTA, such as those in the previous slide, deal with operations in different lines of business rather than with corporate business intelligence. Some operating units in medium and small enterprises can benefit from RTA but don't have the budget or the time to acquire and integrate proprietary software into stream processing apps that generate RTA.
2. Many enterprises carry out analytics in databases and the cloud. Stream processing is most efficient, however, when processing is distributed across phones and smart sensors, local servers, and the cloud. Business units don't have tools that give them a holistic view of their entire distributed processing capability for stream processing.
3. IT staff have deep expertise in their businesses operations and they need tools that enable them to use their expertise to build stream-processing apps.

Many students and others, familiar with Python, want to build streaming applications that leverage Python libraries such as NumPy, SciPy and SciKit. They too don't have tools that help them to do so easily.

StreamPy: Python for stream processing and RTA



Today, more than ever, the resources for building RTA apps are free or inexpensive. Streaming data sources such as sensors and phones are inexpensive. Hardware platforms, including microcomputers such as the Raspberry Pi and Intel Galileo, connected to sensors, are inexpensive and getting more powerful. Many powerful Python libraries for analytics and science are freely available. StreamPy is a platform that integrates inexpensive streaming data sources, open-source software that operates on data at rest, and inexpensive hardware, to obtain RTA apps that operate on data in motion.

StreamPy is designed for those familiar with Python who want to build streaming applications. Students can use StreamPy to explore the development of streaming apps. Operations groups in many lines of businesses, particularly in small and medium enterprises, can benefit by using StreamPy to leverage Python's powerful free libraries.

StreamPy: Python stream processing for RTA

1. Challenge: Businesses can use powerful Python libraries for analytics, but this software usually operates only on data at rest – snapshots of data.

StreamPy encapsulates this software to obtain RTA from data in motion – data as it is generated.

2. Challenge: Business units don't have tools that give them a holistic view of their entire distributed processing capability.

StreamPy gives businesses a holistic view of distribute processing across multiple stages from sensors to the cloud.

3. Challenge: IT staff want to use their expertise to build RTA apps.

StreamPy enables IT staff to build libraries of plug-and-play RTA software modules by encapsulating Python programs, and to build complex apps by plugging modules together.

StreamPy is a platform that addresses challenges faced by enterprises in building RTA apps.

1. Operating units, especially in medium and small businesses, don't have large data-analytics staff and budgets. StreamPy enables them to leverage extensive Python libraries to obtain RTA by analyzing data streams.
2. Businesses often obtain intelligence by processing in the cloud and in corporate databases. RTA processing is most efficient when it occurs at multiple stages from initial data acquisition to decision support. Processing in hundreds or thousands of small computers, such as the Raspberry Pi and Intel Galileo, connected to sensors, reduces the time and cost of computation. StreamPy helps provide a holistic view of stream processing across the entire distributed system.
3. Businesses can benefit from a rich and growing collection of free Python libraries that cover analytics, business, mathematics, science, engineering and displays. StreamPy enables IT staff to encapsulate Python programs to obtain modules that operate on data streams. They can then build complex RTA apps by connecting modules together.

StreamPy is intended for enterprises that want to use powerful Python libraries for RTA. StreamPy is also for students and others who use Python and who want to explore streaming applications that leverage familiar Python libraries.

References

1. K. Mani Chandy and Roy Schulte, Designing IT Systems for Agile Companies, McGraw Hill, 2010
2. K. Mani Chandy, Opher Etzion and Rainer von Ammon (eds.) "The Event Processing Manifesto", Dagstuhl Online Research Publication Server, 2011,http://drops.dagstuhl.de/opus/volltexte/2011/2985/,
3. Opher Etzion, Peter Niblett, Event Processing in Action, ACM Digital Library, 2010
4. Stefan Appel, Pascal Kleber, Sebastian Frischbier, Tobias Freudenreich, Alejandra Buchmann, "Modeling and Execution of Event Stream Processing in Business Processes," ACM Digital Library and Journal of Information Systems, Vol. 46, 2014
5. Maricel Rivera, "Real Time Business Intelligence in the Real World,"https://tdwi.org/Articles/2015/02/03/Real-Time-BI-Real-World.aspx?Page=1
6. Ron Anderson-Lehman, Hugh J. Watson, Barbara H. Wixom, Jeffrey A. Hoffer, "Continental Airlines Flies High with Real Time Business Intelligence"
7. Byron Ellis, Real Time Analytics: Techniques to Analyze and Visualize Streaming Data, John Wiley, 2014