POWERS: If you're collecting large amounts of data in real time and trying to act on it in real time, stream processing is definitely the paradigm for you, and tools like InfoSphere Streams is a powerful platform for you to use for those types of applications.

But in order to react in real time, you have to analyze the stream in real time, and that's where the TimeSeries Toolkit comes in to play. Welcome to this week on developerWorks. I'm your host, Calvin Powers.

My guests today are Bharath Kumar Devaraju and Dattaram Rao from the IBM India Labs. They're both software engineers and have published an article showing how to apply the TimeSeries Toolkit to a very important domain of problems. Bharath, welcome to This Week on developerWorks. First, tell us about the domain of problems you two are focusing on.

DEVARAJU: Hi, Calvin. Thanks for your kind introduction. As you rightly say, InfoSphere Streams is a powerful platform which can be used for analyzing large amounts of data in real time and detect events as they are happening.

So, in our article, we are concentrating on the domain of real-time anomaly detection. So, what is an anomaly? An anomaly is a deviation from the standard behavior, which is unexpected and which is not supposed to happen.

So, what do you mean by real-time anomaly detection? So, we need to detect such anomalies in real time. Consider this example, so, example in large systems where we have [INAUDIBLE] sensors which are monitoring various parameters, and each sensor is ingesting data at very high velocity and the data is of high volume.

So, we have to continuously analyze them looking for any unexpected deviations. So, if undetected, these deviations or anomalies can cause damage to the system. By damage, we mean degrading the performance of the system or even collapsing the entire system.

RAO: An anomaly detection system basically consists of four components. There are data preprocessing, data decomposition, data prediction; and finally, the last stage, anomaly detection stage.

The first one is the data preprocessing stage. As we know, the data is collected using sensors, and these sensors can also induce some noise in this data. And hence, it is necessary to remove this noise. If this noise is not removed, this can lead to errors in further analysis.

How can we implement this noise removal? There are two well known techniques: first one is the normalization; the second one is the filtering. What is normalization? Normalization is a process of converting or transforming the data into a zero mean and unit variance. The second one is the filtering. The well-known filtering method is using exponential smoothing algorithm or the DSP filter.

Now, how can we implement this in the TimeSeries Toolkit? Normalization can be achieved or implemented using the normalized operator in the TimeSeries Toolkit, while the exponential smoothing algorithm can be implemented using the DSP filter operator. In the article, we have mentioned the details of this.

The second stage, the data decomposition. Now, that we have removed the noise, what next? Now, we need to decompose the data or multi variant analysis. Now, what is the meaning of decomposition? Say a signal can be decomposed into OR and its magnitude components. How can we achieve this decomposition?

This decomposition can be achieved using a 50 [INAUDIBLE] transform or discrete cosign transform or wavelength transformation. How can we implement this in the TimeSeries Toolkit?

Now, we have an operator by name DWT. Now, this DWT -- which is the Discrete Wavelength Transform operator -- can be used to decompose the input TimeSeries data. Now, the decomposition will highlight the general trends in the input TimeSeries. Now, this general trend is what is important for detecting the anomaly.

Now, the third stage, the data prediction. Now that we have decomposed the data using DWT, what we can do is use this decomposed data as input and predict the future values or the future trends. The trend may be that it's going high always, or the trend may be that it's going down.
Now, we need to predict this well in advance. Now, how do we achieve this data prediction? While there are many options we can use, FMP filter, fading memory polynomial filter, as a process for predicting the future general transaction.

In the TimeSeries Toolkit, we have this modeling operator, FMP filter, which can be used for prediction purpose. Now we have two different values: one is the predicted; the second one is the current input [INAUDIBLE]. Now, we know what is the future predicted value and now we know what is the current input.

The last stage is the anomaly detection step. Now, this anomaly detection step makes use of these two data points; that is, the current input value and the predicted future value.

So, once we take a distance, say a Euclidian distance, we can say whether this predicted one is an anomaly or not. Now, which operator we can use? We have a GMM operator, Gaussian mixture model, in the TimeSeries Toolkit which can be used to measure the distance and detect an outlier which is nothing but an anomaly. In this way, these four components; namely, the preprocessing, decomposition, prediction and anomaly detection can be implemented using TimeSeries Toolkit.

DEVARAJU: In our article, we have used the example of the monitoring the memory and statistics of a data center, looking for any spikes in them which are anomalies. So in this example, we have actually simulated the input from three different machines in a datacenter so each of them is generating the usage statistics in real time.

So, these usage statistics are collected and analyzed through an anomaly detection system. And these statistics are processed through various phases of our system -- as explained previously, the preprocessing, data transformation, modeling; and finally, anomaly detection.

So, we have also illustrated how the data, the input data has transformed through various phases with the help of the charts. So it's visible from the article that the input [INAUDIBLE] statistics, when it passes through the filtering and normalization, we can visualize that the data which are at different ranges are filtered and brought on with a zero mean and variance.

So they're all brought on to the same range, facilitating the next step; that is, the data transformation, where you can actually see that the operative data transformation has an extra graph of just not presenting the input.

This is basically because of what transformation does is it exposes data [INAUDIBLE] trends and more details about the input which are not visible in the time domain, but it converts into a space where the trends and more details about the data are more prominent.

This transform data is again fed as input into the prediction state where we use these trends and more finer details and predict what is the expected memory usage trend we can expect next.

So, the next step, the anomaly detection step, we basically, it does a metric comparison between the predicted and what was the input. So, this metric comparison basically, it outputs with the probability that given the deviation from what we predicted and what has arrived, is anomaly or not. The more the deviation, with the more confidence the system says that it is an anomaly.

[END OF SEGMENT]