Making sense of IoT data

Storing IoT data, analyzing IoT data, and applying rules to act on your IoT data

Anna Gerber

January 03, 2018
(First published December 19, 2017)

IoT solutions are most successful when they do something with all that data that is generated, when they make sense of all that data. In this article, learn best practices in storing, analyzing, and applying rules to take action on your IoT data.

IoT 301: Mastering IoT development

This article is part of the IoT 301 learning path, an advanced developer guide for IoT.

- Get serious
- IoT security challenges
- IoT device management
- IoT analytics (this article)
- Tutorial: Extend an IoT system

As more and more things are connected to the Internet of Things, the volume of data associated with and generated by IoT devices, including device statuses, metadata, and sensor readings, is increasing exponentially. Managing and making sense of this data is essential if IoT solutions are to deliver value.

Data analytics can be applied to IoT data to generate dashboards, reports, visualizations, and alerts, to monitor the health and status of connected devices, and to provide visibility for sensor readings. Analytics are used to identify patterns, detect anomalies, and predict outcomes from the data, as well to trigger actions through the application of rules.

In this article, I'll describe some of the approaches for dealing with IoT data, including storing data, processing and analyzing data, and applying rules.

Storing data

IoT devices typically have limited data storage capabilities, so the bulk of the data acquired by IoT devices needs to be communicated using communication protocols such as MQTT or CoAP, and

then ingested by IoT services for further processing and storage. More data is being stored and accessed by IoT apps and services than ever before.

Dealing with the increased volume of data is not the only concern with managing stored IoT data. Other considerations include:

- Dealing with heterogeneous data
- Transforming, aggregating, and integrating data in order to prepare it for analytics while keeping track of data provenance
- Securing the data to maintain the integrity and privacy of the data
- Selecting storage technologies to ensure a balance between high performance, reliability, flexibility and scalability and cost.

Read more tips for storing IoT data.

Heterogeneous data

The data captured by IoT devices is produced in a mix of data formats, including structured, semi-structured, and unstructured data. This data might include discrete sensor readings, device health metadata, or large files for images or video. Because IoT data is not uniform, no one-size-fits-all approach exists for storing IoT data.

Transforming data

Data is usually transformed on the device or at device gateways to perform normalization. Events may need to be re-ordered if they have been received out of order, and if data is time sensitive, stale data might be dropped.

Provenance information about the sensor that captured the data, as well as location and timestamp for the data, is often attached at this point too. It is useful to store raw data for debugging and historical analytics purposes, but storing the pre-processed data will avoid having to repeat expensive transformations if the data should need to be analyzed more than once.

Be sure to transform your data and then be selective about which data to transmit and store to save on bandwidth and storage costs.

Securing data

Data must be transmitted and stored securely in order to maintain data integrity and privacy, using secure protocols and encryption. Read more in "Securing IoT data over the network" on IBM developerWorks.

Data storage technologies

Because IoT solutions are complex, you'll need to use a combination of data storage strategies.

Data storage technologies that you use for data stored temporarily at the edge while it is being processed on devices and gateways will need to be physically robust (to withstand the often harsh operating environments in which devices are installed), but also fast and reliable so that pre-

processing of data can be performed quickly and accurately before data is sent upstream. Device data storage can take advantage of next-generation non-volatile RAM (NVRAM) technologies such as 3D X Point and ReRAM, which are up to 1000 times faster than NAND flash but consume less power than DRAM.

Once IoT data is transmitted from the device that generates it, it can be stored in the cloud, on premises, or a hybrid of the two.

Data is likely destined for different purposes. Data that is intended for archival purposes rather than real-time analytics can be stored using different approaches that complement the analytics approach that will be adopted. Data access needs to be fast and support querying for discrete real-time data analytics. Storage technologies adopted for this data should support concurrent reads and writes, be highly available, and include indexes that can be configured to optimize data access and query performance.

High volume archival data can be stored on cloud storage that may be slower to access but it has the advantage of being lower cost and elastic so it will scale as the volume of data increases. Access to large file format data like video is usually sequential, so this data might be stored using object storage (such as OpenStack Object Storage known as Swift), or written directly to a distributed file system like Hadoop HDFS or IBM GPFS. Data warehouse tools like Apache Hive can assist with managing reading, writing, and accessing data that is stored on distributed storage.

Data storage technologies that are often adopted for IoT event data include NoSQL databases and time series databases.

- **NoSQL databases** are a popular choice for IoT data used for analytics because they support high throughput and low latency, and their schema-less approach is flexible, allowing new types of data to be added dynamically. Open source NoSQL databases used for IoT data include Couchbase, Apache Cassandra, Apache CouchDB, MongoDB and Apache HBase, which is the NoSQL database for Hadoop. Hosted NoSQL cloud storage solutions include IBM's Cloudant database and AWS DynamoDB.
- **Time series databases** can be relational or based on a NoSQL approach. They are designed specifically for indexing and querying time-based data, which is ideal for most IoT sensor data, which is temporal in nature. Time series databases used for IoT data include InfluxDB, OpenTSB, Riak, Prometheus and Graphite.

Read more about Apache Cassandra in this IBM developerWorks tutorial, "Set up a basic Apache Cassandra architecture."

Analyzing data

IoT data needs to be analyzed in order to make it useful, but manually processing the flood of data produced by IoT devices is not practical. So, most IoT solutions rely on automated analytics.

Analytics tools are applied to IoT data including device status data and sensor readings, to generate descriptive reports, to present data through dashboards and data visualizations, and to trigger alerts and actions.

Many different open source analytics frameworks or IoT platforms can be used to provide IoT data processing and analytics to your IoT solutions. Analytics can be performed in real-time as the data is received or through batch processing of historical data. Analytics approaches include distributed analytics, real-time analytics, edge analytics, and machine learning.

Distributed analytics

Distributed analytics is necessary in IoT systems to analyze data at scale, particularly when dealing with historical data that is too vast to be stored or processed by a single node. Data can be spread across multiple databases; for example, device data might be bucketed into databases for each device per time period, such as hourly, daily, or monthly, like the IBM Watson IoT Historian Service that connects to Cloudant NoSQL database that stores the IoT data. Analytics may involve aggregating results which are distributed across multiple geographical locations. You'll want to adopt a storage driver or analytics framework that bridges distributed storage and compute infrastructure to allow seamless querying across distributed databases.

Of particular note for processing of distributed data are the ecosystem of frameworks arising from the Hadoop community. Apache Hadoop is a batch processing framework that uses a MapReduce engine to process distributed data. It was one of the first open source frameworks to take off for big data analytics. Hadoop is ideal for historical IoT data analytics where time sensitivity is not an issue, such as performing analysis over a complete set of data and producing a result at a later time.

Real-time analytics

Analytics for high-volume IoT data streams is often performed in real-time, particularly if the stream includes time-sensitive data, where batch processing of data would produce results too late to be useful or any other application where latency is a concern.

Real-time analytics are also ideal for time series data, because unlike batch processing, real-time analytics tools usually support controlling the window of time analysis, and calculating rolling metrics, for example, to track hourly averages over time rather than calculating a single average across an entire dataset.

Frameworks that are designed for real-time stream analytics include Apache Storm and Apache Samza (usually used with Kafka and Hadoop YARN). Hybrid engines that can be used for either stream or batch analytics include Apache Apex, Apache Spark, and Apache Flink. Apache Kafka acts as an ingestion layer that can sit over the top of an engine like Spark, Storm or Hadoop. For a guide to selecting between these open source frameworks, read Choosing the right platform for high-performance, cost-effective stream processing applications.

Edge analytics

IoT analytics is not usually applied to raw device data. The data is pre-processed to filter out duplicates or to re-order, aggregate or normalize the data prior to analysis. This processing

typically occurs at the point of acquisition, on the IoT devices themselves or on gateway devices that aggregate the data, to determine which data needs to be sent upstream.

Analytics applied at the edges of the network, as close as possible to the devices generating the data is known as *edge analytics*. IBM's Watson IoT Platform includes an Edge Analytics SDK. Apache Edgent (previously known as Quarks), is one example of an open source, lightweight embedded analytics engine designed to be used on IoT devices and gateways. Analytics is also on the roadmap for the Linux Foundation's EdgeX Foundry, an open source IoT edge computing framework.

Edge analytics is low-latency and reduces bandwidth requirements because not as much data needs to be transmitted from the device. However, constrained devices have limited processing capacity, so most IoT solutions use a hybrid approach involving edge analytics and upstream analytics.

Machine learning

Using traditional mathematical statistical models for analytics provides value as they can be used to track goals, create reports and insights, predict trends, and create simulations that are used to predict and optimize for specific outcomes. For example, you can predict the outcome of applying a specific action, predict the time to failure for a given piece of equipment, or optimize the configuration of an IoT system in terms of cost or performance. However, the value of statistical analytics models diminishes when applied to dynamic data that contains many variables that change over time, when you don't know what factors to look for, or what variables to change to achieve a desired outcome like reducing cost or improving efficiency. In these cases, instead of using a statistical model, machine learning algorithms that learn from the data can be applied.

Machine learning can be applied to historic or real-time data. Machine learning techniques can be used to identify patterns, identify key variables and relationships between them to automatically create and refine analytics models, and then use those model for simulations or to produce decisions. Machine learning approaches have the advantage over static statistical analytics models that as new data comes in, the models can be improved over time, which leads to improved results.

Machine learning approaches include:

- **Neural networks.** Neural networks are networks of nodes that take data as input and produce a decision or result as output. The neural network is trained with data to produce a desired outcome.
- **Deep learning.** Deep learning is like a neural network, but with more hidden layers between the inputs and outputs.
- **Long short term memory (LSTM).** LSTM networks use feedback loops to include some memory to give them some context when processing time-series data.

Read this use case of applying SystemML and Apache Spark to IoT data to track retail customer behaviour and improve the customer experience.

Watch the IoT Data Analysis in real time using Deep Learning keynote to find out more about these different approaches, or read more about detecting anomalies in IoT time-series data by using deep learning in this developerWorks tutorial.

Standalone open source deep learning frameworks for IoT include TensorFlow, PyTorch, Theano or Cognitive Toolkit CNTK. Machine learning tools are also available that run over the top of Apache Spark, for example, the H20 and DeepLearning4J deep learning engines, and SystemML.

Acting on the data by applying rules

Rule engines use the insights discovered from IoT data through analytics, and apply rules to produce accurate, repeatable decisions. There are a variety of approaches to implementing rules for IoT:

- Decision trees. Decision points are represented using a tree (that is, a hierarchical graph-based structure), to represent decision points. Each leaf in the tree structure corresponds with a decision.

Decision trees grow exponentially as the possible number of variable states being tracked increases.

- Data flow graphs. Data flow graphs (also known as pipes) are used to represent the data dependencies and flow between functions. An example of a framework that is built on data flow graphs is TensorFlow.
- Complex Event Processing (CEP). CEP is often used for processing time series data, and it is a good fit for applying rules to real-time sensor data.
- Inference Engines. implement If-then-style rules.
- Business Process Management. In the context of a traditional BPM system, rules are often implemented as decision tables.

Regardless of the engine's implementation, rules encode conditions that are used to trigger actions that autonomously make adjustments to a system to work around issues, maximize performance, or optimize costs.

Open source analytics and rules engines can be self-managed; however, many IoT platforms offer hosted analytics and rules solutions based around open source offerings, such as IBM's Analytics for Apache Spark. IoT Platforms also typically include custom integrated analytics solutions, incorporating both batch and real-time analytics as well as machine learning and rules. Some examples of IoT platforms that include these capabilities include:

- Oracle Edge Analytics and Stream Analytics
- Azure Stream Analytics
- AWS IoT
- Google Cloud Dataflow
- IBM Watson IoT Platform Analytics

Conclusion

I've provided an overview of tools and approaches for making sense of IoT data, including managing data, using analytics to gain insights, and applying rules to perform actions.

IoT data analytics is essential to the management of large scale, complex IoT systems like connected cities, where analytics are used to predict demand and rules are applied to adjust services in response, such as to control adaptive traffic signals or manage smart lighting.

IoT data analytics improve efficiency in smaller-scale IoT systems too. For example, for predictive maintenance of industrial equipment or automotive fleets, sensor data installed on assets is used to predict when they will need to be serviced which allows maintenance to be scheduled at optimal times, which improves reliability, and which avoids the expense of unnecessary maintenance and reduces downtime and productivity loss.

Related topics

- Watson Data Platform: Solutions for storing and analyzing your IoT data
- developerWorks series: Developing cognitive IoT solutions for anomaly detection using deep learning
- Using IBM Watson Analytics to visualize data from Watson IoT Platform
- Visualizing and understanding data from IBM Watson IoT Platform by using IBM Data Science Experience
- developerWorks recipes for storing and visualizing historical data using IBM Watson IoT Platform and IBM Cloudant