If you can't join them, JSON them

Adding NoSQL to Teradata

February 2014

A Point of View by
Dr. Barry Devlin, 9sight Consulting

barry@9sight.com

The rapidly emerging Internet of Things presents enormous opportunities for data warehousing and analytics. It also presents a number of challenges. Both aspects are addressed head-on by Teradata's embedding of JSON in its flagship database product.

Together we're better

As the Internet of Things (IoT) takes shape, one thing is certain. The nature of data in business is changing dramatically, fragmenting and expanding. The vast majority of event data will come from machines and sensors, rather than people. Their sources will be diverse and distributed, their provenance debatable. Their meanings and structures will become disparate and changeable. Most importantly, their uses and value will be mostly determined only after they are generated and received.

Therefore, the applications and the databases that process them must be very different from those used in traditional business processing, where data is modeled in advance and seldom repurposed. That was the belief of the NoSQL movement: we need to adopt new tooling and discard the old. But that belief was half-wrong. Yes, we do need new tools, but they must coexist with the old. They must benefit from existing technology, learn from well-exercised data management approaches, and build on long-standing skills.

By embedding JSON support in its existing relational database environment, Teradata has bridged the classic world of data warehousing to the IoT. The power and data management strengths of the RDBMS can now support the new world of machine-generated data. Analytic users can easily explore this diverse and changeable world with skills similar to those of traditional SQL.

JSON—what for and why now

The relational model for data structure and storage has been the technology of choice for all business application data—first informational and then operational—for almost thirty years. Its success stems from a strong mathematical foundation and strict logical structuring that ensures a logically complete and correct representation of modeled data for a wide variety of applications.

In the past decade, internet-based business and, more recently, the Internet of Things has operated in a rapidly evolving environment where business needs change rapidly and data needs are voluminous and highly volatile in structure. The formal relational approach is less well suited to such needs, and a number of new, NoSQL models for data structure and storage have emerged. JSON, Java Script Object Notation, is among the more widely adopted, especially for data transmitted by devices on the IoT, because of its simple lightweight nature and the agility it offers developers.

JSON data consists of nested name-value pairs, usually stored as text. It can be compared to XML in some respects, but is much simpler and looser in its structure and use. The following JSON *document* shows data from an imaginary weather sensor atop the Empire State Building in New York:

```
{       id: “1F36BB2A”
        GPS: { longitude: “-73.98556”, latitude: “40.748328”, altitude: “1295” },
        date_time: “2014-01-06 03:15:00”,  temp: “-35”   }
```

The example shows simple name-value pairs, such as id, date_time and temp, as well as nesting in the GPS entry. The structure is very loose and the parsing and interpretation of each entry depends fully on the application that receives the data. For example, the units for altitude and temperature are feet and Fahrenheit, respectively, but that clarifying metadata cannot be found in the data itself.

If the sensor is upgraded to measure wind speeds, we simply insert more name-value pairs. It is from these features that JSON’s agility emerges. Each JSON document is the logical equivalent of a row in relational terms, but unlike a relational database, the variable (column-equivalent) names appear in every document, allowing each document to contain different sets of fields. Although this redundancy may appear very wasteful of storage space, IoT data is often very sparse, a characteristic that can lead to enormous volumes of null values in normal relational formats.

Welcoming JSON to the data warehouse

Native JSON data stores, such as MongoDB and CouchDB, are now popular where data needs are voluminous and volatile. In pure operational applications, these databases provide an ideal environment for application development and ongoing upgrades. However, for informational needs, especially of the more complex variety found in enterprise data warehouses, the situation is problematical.

The data manipulation language used to access JSON data stores is procedural and Java-like in syntax, allowing the equivalent of many SQL statements to be constructed. That’s great for programmers but less so for business users. Furthermore, the agility of the JSON structure leads to the situation where the metadata (field names) are recorded independently in each document, as opposed to the relational model where a single, definitive catalog can be depended upon. Finally, we can see that most analytics of IoT data requires combining this data with existing data stores. In short, we need at least some IoT data to be available in the data warehouse environment.

A similar situation had previously arisen with the emergence of XML in the 1990s. Given the similarities between XML and JSON, the obvious solution to bringing JSON into the data warehouse was to adopt a comparable approach. This involves storing JSON documents in Large Objects (LOBs) in their original textual structure, with indexes on all field names. In most cases, it makes sense to partially shred key, always-occurring JSON fields into separate columns for ease-of-use and improved performance. In the above example, id might be stored as a separate column (called dev_id), while all the various sensor readings would be stored in the LOB (called sensor).

Teradata Database uses the highly popular access method called *JSONPath Dot.notation*, which is based on the XPath (XML Path Language), but with a simpler and lighter approach. Thus, we can write a simple, combined SQL / JSONPath statement to retrieve the current location and temperature:

```
SELECT sensor.GPS.longitude AS Long,
        sensor.GPS.latitude AS Lat,
        sensor.curr_temp AS Temp
        WHERE dev_id = “1F36BB2A”;
```

where the Dot.notatation names provide access to specific JSON fields according to their names.

If, as previously mentioned, the device is upgraded to also measure wind speeds, the data warehouse table and load routines remain the same and the above query will continue to work on new documents which now contain additional information. A new query that retrieves wind speed will also work on all documents, simply returning null values from old-style documents that do not include the new data.

This approach provides the familiarity and ease-of-use of traditional BI usage while retaining the agility of the JSON data structure. This latter flexibility is also known as "schema-on-read", with such late binding supporting evolving and changing schemata in the data. In a traditional data warehouse, the approach is "schema-on-write", which favors data management and governance over flexibility. Now both data handling styles are available in the data warehouse.

Data warehousing in the Internet of Things

Name-value pairs, whether in JSON or other formats, have become the *lingua franca* of the world of machine-generated data and the Internet of things. Industrially and personally, society is rapidly accepting and implementing a pervasive and heterogeneous network of sensors that connect the physical world to the digital environment. As this occurs, analytics becomes central to monitoring what happening and deciding how to influence outcomes. However, the volumes of data are such that detailed human oversight is impossible, as seen in the following three areas:

1. In the manufacturing and distribution environments, sensors in machines, trucks and even on shelves in stores create a fully automated overview of the entire supply chain. This offers opportunities to improve product quality, reduce distribution costs and errors, and ensure optimum stock levels in store. These outcomes depend on the in-depth analysis of machine-generated data in combination with the process-mediated data from traditional business applications. Such combining of data from diverse sources has been a fundamental component of data warehousing since the earliest days.
2. Environmentally, whether dealing with weather patterns, energy usage or traffic on city streets, an explosion of sensors is allowing ever growing volumes of data to be gathered. Analysis of such data is vital to drive greener behavior, reducing consumption of energy and valuable resources, and enabling citizens to make better choices, save time and reduce costs. Initial analysis of such high volume and velocity data may often be performed by data scientists on commodity, Hadoop-based systems. Even so, long term retention, management and trending analysis for ongoing, mainstream reporting and querying is likely to be better supported in an expanded relational environment.
3. At a personal level, 2014 is seeing wearable devices becoming increasingly common and even fashionable, monitoring all aspects of health, location and activity. Despite very real security and privacy concerns, this trend is likely to grow rapidly. Consolidation and analysis of this data will drive a plethora of new business opportunities and applications. For example, we can envisage that the analysis of health and exercise data will enable new applications in areas ranging from emergency services through medical treatment to insurance underwriting. Initial collection and analytics may be done via both traditional and NoSQL (including Hadoop) technologies. However, hardening this data into secure production environments for both operational and informational business use demands the reliability, availability and serviceability of systems like relational databases.

While the initial industry focus may be on the new data from the IoT and its very different characteristics, these examples demonstrate that longer-term use and value will derive from the combination of this novel data with existing operational and informational data and its integration into robust production systems.

Conclusions

The Internet of Things is producing a new and extensive data ecosystem of measures and events from every aspect of the physical world. This data is often characterized by a name-value pair structure, and is widely implemented using JSON as a foundation. Its value and new uses emerge initially through analytics and then through embedding it in real-time production use. IoT data must become part of the historical and reporting environment to track and optimize its longer-term value.

A wide variety of use cases across all industries show that deep operational and predictive analytics is the foundation of all significant business and societal opportunities arising from IoT data. As a result, machine-generated data must be managed with the same care and performance granted to traditional process-generated sources. Both types of data must be combined in a dedicated analytic environment, without losing the flexibility and agility characteristic of the Internet of Things. The Teradata data warehouse, with the addition of JSON support, provides an ideal environment to pursue these opportunities.

Dr. Barry Devlin is among the foremost authorities on business insight and one of the founders of data warehousing, having published the first architectural paper on the topic in 1988. With over 30 years of IT experience, including 20 years as an IBM Distinguished Engineer, he is a widely respected analyst, consultant, lecturer and author of the seminal book, "Data Warehouse—from Architecture to Implementation" and numerous White Papers. His new book "Business unIntelligence—Insight and Innovation Beyond Analytics and Big Data" was published in October 2013.

Barry is founder and principal of 9sight Consulting. He specializes in the human, organizational and IT implications of deep business insight solutions that combine operational, informational and collaborative environments. A regular contributor to BeyeNETWORK, Focus, SmartDataCollective and TDWI, Barry is based in Cape Town, South Africa and operates worldwide.