O'REILLY

Compliments of VOLTDB

Fast Data Use Cases for Telecommunications

How Fast Data Can Help Telcos Virtualize, Monetize, and Deal with the Data Deluge

Ciara Byrne


VOLTDB
5G: It's not an evolution, it's a revolution.
Are you ready?
LEARN MORE

Fast Data Use Cases for Telecommunications

How Fast Data Can Help Telcos Virtualize, Monetize, and Deal with the Data Deluge

Ciara Byrne

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Fast Data Use Cases for Telecommunications

by Ciara Byrne

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com/safari*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Tim McGovern
Production Editor: Nicholas Adams
Copyeditor: Octal Publishing, Inc.
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

September 2017: First Edition

Revision History for the First Edition

2017-09-06: First Release

978-1-491-99823-6

[LSI]

Table of Contents

Fast Data Use Cases for Telecommunications

Big data is data at rest. Fast data is data in motion: a relentless stream of events generated by humans and machines that must be analyzed and acted upon in real time. Data is fast before it becomes big through export to a long-term data store.

Fast data applications must ingest vast amounts of streaming data while maintaining real-time analytics and making instant decisions on the live data stream. A fast data application in a telco might enforce policies, make personalized real-time offers to subscribers, allocate network resources, or order predictive maintenance based on Internet of Things (IoT) sensor data.

This ebook covers not only *why* telcos need fast data but also the technical characteristics of several telco-specific fast data use cases and examples of real-life deployments. VoltDB is an in-memory, NewSQL database that became popular with telcos for its ability to handle the speed and scale of fast data. This ebook reflects the experiences of VoltDB engineers and customers who have deployed multiple telco fast data use cases.

"Telecom is really hard," says Michael Pogany, head of business development in VoltDB's Telecom Solutions Group. "Telecom is unique. Our telecom clients are the most demanding and the most visionary of our customers."

Why Telcos Need Fast Data

Telco networks have always generated fast data at line speed. In telco use cases like policy management, decisions are already made on that data in near real-time. In many other cases, network and customer data is backhauled into a data lake and analyzed over hours or days to gain insight into the subscriber experience or the quality of the network.

Two fundamental changes will bring fast data systems to the forefront at every telco operator: a massive increase in the volume of streaming-data service providers need to process, and the need to act on that data in milliseconds.

"Real time is making decisions on the data within milliseconds of the event happening," says Pogany. "There are elements of the Telecom network that operate that fast, but now the entire network and all of the systems supporting it are going to have to operate that fast."

Fast data applications will operate the agile, automated, virtualized network infrastructure created by *Network functions virtualization* (NFV), *Software-defined networking* (SDN) and eventually 5G. Fast data will enable telecom service providers to personalize services and deploy new ones like IoT to boost declining revenues. Fast data is the future of telco.

Fast OSS and BSS Systems

Service providers are facing a data deluge. Annual global IP traffic will reach 3.3 Zettabytes (ZB) per year by 2021, up from 1.2 ZB in 2016, according to a report from Cisco. Sixty-three percent of that data will come from wireless and mobile devices. Globally, mobile data traffic will increase sevenfold between 2016 and 2021. Cisco predicts that global IoT IP traffic—from devices like smart meters, home security and automation systems, connected cars, and healthcare monitors—will grow more than sevenfold by 2021. On top of this explosion in devices, faster network technology (the advent of 5G) is another major factor nudging data traffic toward exponential growth.

Operations Support Systems (OSS) and Business Support Systems (BSS), many of which rely on batch processes, are already creaking under the strain. Telco service providers don't just need flexible network infrastructure to deal with a massive increase in traffic while keeping costs under control, they need support systems that can keep up.

Use cases like *least-cost routing*, *subscriber management*, *policy management*, *real-time billing*, *authentication and authorization*, and *fraud detection* all require real-time decision making. OSS/BSS providers like Openet and Nokia are meeting the challenge by adding fast data support with real-time decision-making capabilities to their products.

New Services

Although the demand for data has exploded, average revenue per subscriber has fallen globally over the past decade, according to PwC's 2017 Telecommunications Trends report. Telco service providers face a continual decline in revenue unless they can launch revenue-generating new services and monetize customers more efficiently. According to Michael O'Sullivan, CTO of Openet:

> Over-the-top players can launch a new service very quickly, leveraging all of the infrastructure those service providers have built, leveraging the devices the service providers have often provided for free to the subscribers and the service providers, whose only return is a fixed monthly charge to lease the connectivity

PwC's report suggests that service providers pick a service vertical—branded content, financial services, lifestyle services—in which to specialize. Some service providers have already bought content companies to get a bigger slice of the content service business: Verizon acquired AOL in 2015, and AT&T recently announced that it wants to buy Time Warner for $85 billion.

Video content is one of the immediate drivers of the data deluge. Global IP video traffic will grow threefold from 2016 to 2021, and video will by then account for 82 percent of all IP traffic. To extract the maximum business value from video customers, service providers must collect viewing data and analyze it in real-time to personalize video offerings and advertising. This is a classic fast data use case. Many other personalized services will have similar requirements.

IoT devices can provide a new source of both connectivity revenue and service revenue to service providers. IoT use cases like healthcare monitoring or predictive maintenance require real-time analysis and decision-making on incoming streams of sensor data. Fast data systems will be a key enabler for IoT.

Flexible Infrastructure

To launch new services while keeping costs down, service providers need flexible, automated network infrastructure. "You're going to need to deploy services within the speed of a marketing window, and to be able to do that, there's only one answer," says VoltDB's Pogany, "It's called the cloud."

Even service providers who were previously hesitant about virtualization are adopting NFV and SDN technologies to modernize their networks; for example, to deploy a virtualized Evolved Packet Core (vEPC), a framework for virtualizing the functions required to converge voice and data on 4G networks.

One IDC study showed that a flexible orchestration layer for vEPC can reduce the time to market for new services by 67 percent. "I know of three different service providers who told me around three years ago, 'Virtualization? No chance', because of the overhead of 15 to 20 percent of running a VM, who have all shifted to push forward on it now," says Openet's O'Sullivan.

McKinsey estimates that technologies like NFV and SDN will allow service providers to lower their capital expenditures by up to 40 percent (and operating expenditures by a similar amount), pushing these costs down to less than 10 percent of revenues as opposed to around 15 percent today. By 2020, AT&T expects to reduce operational expenses by up to 50 percent by virtualizing 75 percent of its network.

NFV uses real-time system metadata for orchestration. 5G networks will deploy network resources in real-time to address the Quality of Service (QoS) requirements of each service or application. Fast data is therefore a prerequisite to operating future network infrastructure.

The Four Functions of a Fast Data System

Interacting with fast data is fundamentally different from interacting with big data. Telco fast data applications need to not only capture streaming data, but also enrich that data with context and personalization, calculate real-time analytics, make decisions and act before the data comes to rest. Fast data systems must perform four basic functions within a telco: ingest, analyze, act, and export (see Figure 1-1). Let's look at each of these in more detail.

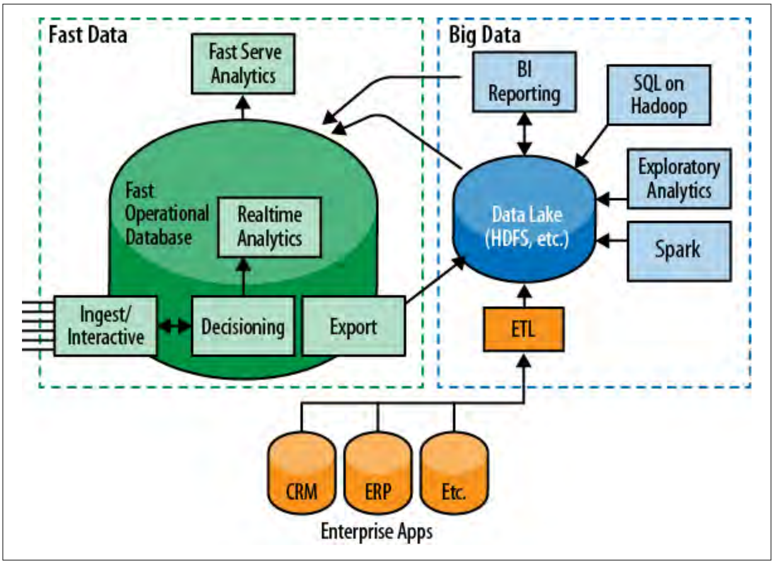

Figure 1-1. A fast data and big data architecture

Ingest

Streaming data often describes events or requests, as shown in Table 1-1. Each event in the data feed must be examined and might need to be validated, transformed, or normalized before it can be used by a fast data application.

Table 1-1. Types of data

Data set	Temporality	Example
Input feed of events	Stream	Click stream, tick stream, sensor outputs, M2M, gameplay metrics
Event metadata	State	Version data, location, user profiles, point-of-interest data
Big data analytic outputs	State	Scoring models, seasonal usage, demographic trends
Event responses	Events	Authorizations, policy decisions, triggers, threshold alerts
Output feed	Stream	Enriched, filtered, correlated transformation of input feed

In fact, many fast data applications need to handle both fast/streaming and big/stateful data. Incoming data is streaming. Metadata about the events in the stream is stateful, as are profiles, models and other big data analytics. Relevant stateful data is often cached by a fast data system so that it can be accessed in real time. Event responses like alerts or authorizations, which are the result of decisions, need to be pushed to downstream systems.

Analyze

Real-time analytics like counters, aggregations, and leaderboards summarize the data on the live feed. For example, a policy management application might maintain usage metrics for individual users.

Traditionally, analytics were calculated after data came to rest in a data warehouse. Real-time analytics can be performed on live data streams as a transaction takes place, and the results streamed off to a data warehouse to be used to update big data analytics like predictive and machine learning models.

Act

Fast data applications must make per-event decisions on incoming data and then act on those decisions. In telco, real-time decisions might be authorizations, policy evaluations, network resource allocations or personalized responses to customers.

To make efficient decisions, streaming data first needs to be enriched with stateful data such as the following:

- Real-time analytics calculated on the incoming stream.
- Batch analytics from a warehouse or data lake; for example, customer segmentation reports for personalization.
- Contextual metadata about the events in the stream; for example, IoT device version numbers or location data.

A rules engine making automated decisions needs to transact against each event as it arrives, to access relevant stateful data and to save results and decisions. Enrichment data is often hosted in a fast, scalable query cache.

Export

Fast data applications must export data to backend systems. Rules engines generate event responses such as alerts, alarms, and notifications, which need to be pushed downstream; for example, to a distributed queue like Kafka.

A subset of the incoming data stream may also need to be exported to a big data store for further analysis. A fast data system should therefore enable real-time extract, transform, and load (ETL) of the feed to a big data store like OLAP storage or Hadoop/HDFS clusters.

Nonfunctional Requirements for Telco

A fast data system for telco must not only implement the functions of fast data but also must conform with telco's stringent nonfunctional requirements on speed, scale, and cost.

Speed
: Telco service providers need a high-performance, low-latency data store that can keep up with the speed and scale of a telco network. When a subscriber tries to make a mobile call, the policy management and charging system must access all relevant data, make a decision to let the call through or deny it, and respond in milliseconds.

Scale
: Data management is more difficult to scale than computation. Applications like IoT will exceed the scale of traditional tools and techniques, so service providers need to be able to scale-out on commodity hardware.

Cloud ready
: Service providers are virtualizing network infrastructure with technologies like NFV and need to scale-out as required to deal with the data deluge.

Immediately consistent
: Eventual consistency means that multiple replicas of the same value in a distributed database might differ temporarily but will eventually converge to a single value. However, this single value is not guaranteed to be the newest or most correct value. Telco use cases like real-time billing and authentication require 100 percent accuracy. Telcos need a database with immediate consistency, where all replicas of the same data are guaranteed to have the same value.

Cost effective
: Service providers need to manage hardware costs, software licensing costs, and operational costs while dealing with the data deluge.

Use Case: Mediation, Policy, and Charging

Mediation collects network and usage data across a wide variety of networks for business intelligence as well as for charging, billing, and policy management. Mediation has traditionally been a batch process executed regularly on massive amounts of data but is moving toward real time.

Policy-management systems control subscriber access to an increasingly virtualized network that offers multiple services, charging and policy rules, and QoS levels. Policy systems must make real-time decisions at the network edge. Service providers moving to Evolved Packet Core (EPC), Long-Term Evolution (LTE), and IP Multimedia Subsystem (IMS) require evolved charging systems to collect and rate data transactions in real-time. Policy and charging have always had strict latency requirements with responses expected in less than 50 milliseconds.

A huge increase in the volume of data to be processed, strict latency requirements and the need to make instant decisions on real-time data make mediation, policy, and charging attractive use cases for fast data.

Openet Case Study

Openet is a leading supplier of OSS and BSS systems, including mediation, policy, and charging products. "Openet processes more transactions per second for a single operator in the United States than Google does searches worldwide," says Michael O'Sullivan, global vice president at Openet, "It's somewhere in the region of 18 billion transactions a day." In 2016, Google was processing 3.5 billion searches per day. Openet's charging products typically need to respond to a request in less than 10 milliseconds.

Openet is evolving its mediation product to deal with the data deluge, in particular IoT data, and to make decisions on that data in real-time. Openet recently demonstrated internally that the new solution can process 1 trillion events per day. "VoltDB was very key as part of the overall solution stack in enabling that," remarks O'Sullivan.

In 2012, Openet began to evaluate databases to support fast data applications. Speed and scale weren't the only considerations. O'Sullivan elaborates:

> We were heavy users of Oracle at the time and we had challenges with that. One of the challenges was total cost of ownership [TCO]. The Oracle platform was rather expensive to operate both from a licensing point of view and a hardware footprint point of view, and it wasn't really friendly to a world where the telcos were advancing to NFV.

Telco charging systems deal with billions of dollars. Charging can't be close enough; it must be accurate. Immediate consistency was essential. "The eventual consistency model just doesn't work when you're dealing in cash," said O'Sullivan.

Openet's policy and charging systems make real-time decisions, so the company also needed SQL transactions and stored procedures. VoltDB processes each incoming event or request as a discrete ACID (Atomicity, Consistency, Isolation, Durability) transaction. Rules can be encapsulated in a VoltDB stored procedure combining SQL and code.

"Stored procedures run on server side close to the data, which meant that we didn't have to do round trips over and back (to get the data)," remarks O'Sullivan, "In a world where milliseconds count, and they do in our world, that became an issue for us."

VoltDB is now used in all of Openet's products. The main advantages for Openet of switching from Oracle to VoltDB were the following:

Speed
: VoltDB can meet Openet's stringent latency requirements.

Scale
: VoltDB has been demonstrated to handle up to 1 trillion transactions per day. Oracle struggled to handle complex calculations on high levels of transactions.

Cloud ready
: VoltDB is a completely virtualizable database that fits into the infrastructure of operators moving to NFV.

Immediately consistent
: Charging requires accuracy; thus, a data store with eventual consistency was not an option.

Cost
: Openet has saved an average of $500,000 per installation due to lower software licensing fees, a smaller hardware footprint, and the operational simplicity of VoltDB.

Use Case: NFV and 5G

NFV aims to virtualize entire classes of network function currently running on dedicated hardware. Service providers are deploying NFV in order to cost-effectively scale their networks up and down to deal with the data deluge and to launch new services faster.

SDN can support NFV efforts by providing a centralized view of the distributed network for more efficient orchestration and automation of network services. NFV and SDN are complementary but don't necessarily need to be deployed together.

"NFV is a paradigm shift," explains Dheeraj Remella, director of solutions architecture at VoltDB, "Everything needs to be virtualized. Everything needs to be software driven. Policies and decisions that are being made at the hardware level need to move into the software layer."

Those policies needed to be automated and implemented in real time in software. NFV orchestration uses real-time utilization data from compute, network, and storage elements to make decisions about where to place Virtual Network Functions (VNFs) and whether resources need to be scaled up or down. SDN requires a data-driven representation of policies, network metadata, and route cost information.

"To do NFV orchestration, you need metadata about the system itself," says Openet's Michael O'Sullivan, which has launched a community version of its VNF manager. "If an operator rolls out the ultimate stage NFV, which is that the SDN network can be reshaped based on traffic and new VNFs can be spun up as needed on an on-demand basis, you need a lot of data to make sure that you're making the right decision." In other words, NFV needs fast data.

5G

NFV is an essential step toward 5G, and 5G is the on-ramp to IoT. With 5G, service providers must support latencies as low as a millisecond and 10 Gbps data rates. But 5G is not just about more speed and scale. 5G network slicing allows service providers to split a single physical network into multiple virtual networks and apply different policies to each slice to offer optimal support for different types of services.

A service provider could, for example, partner with a content provider to offer higher QoS on a particular network slice or connect smart meters on a network slice that offers a high availability, data-only service with guaranteed latency, data rate, and security levels.

VoltDB's Pogany points out expands on the point:

> 5G is not one more G. It is not a little bit faster or a little bit more data; it turns the entire business proposition of a telco on its head. Every telco now must open its network and differentiate itself at every layer in the stack to enable multiple sources of data, multiple kinds of revenue streams, and multiple kinds of partnering schemes. Running 5G, your customer could be a car, a house, or a tea kettle.

Data-driven policy management will be extremely important in 5G because every data slice will have its own set of policy rules. A fast data rules engine will therefore be an essential enabler of 5G applications. That rules engine must be able to support billions of messages in real time to quickly deploy the necessary network resources to address the QoS requirements of each service or application.

Fast data also can help reduce the cost of managing the huge amount of operational data generated by 5G services. "The cost of hauling an entire telecom network into a data lake and then processing it is enormous," says Pogany, "We can ameliorate this huge investment by handling some of that data in real time."

Nokia Case Study

NFV is often deployed in parallel with traditional hardware network functions to gradually move toward full virtualization. 5G is still some way off, so service providers are first using virtualization to improve the efficiency of their existing core packet networks; for example, in the vEPC environment.

Nokia's Cloud Packet Core is designed to help deliver converged broadband and IoT communication while creating an evolution path to 5G. Cloud packet core products like the Cloud Mobility Manager and the Cloud Mobile Gateway can be deployed on servers or as cloud-native virtualized VNFs, enabling Nokia customers to seamlessly transition to NFV and SDN. VoltDB will be integrated into both products and is already deployed in the Nokia Telecom Application Server, another component of the Nokia Telco Cloud.

The Cloud Mobility Manager performs the MME/SGSN (Mobility Management Entity/Serving GPRS Support Node) functions within the packet core network. MME is the main signaling node in the evolved packet core. SGSN handles all packet switched data within the GPRS network.

The Cloud Mobility Manager also supports the Cellular IoT-Serving Gateway Node (C-SGN) function within narrowband IoT networks. The NarrowBand IoT low-power wide-area network radio technology standard has been developed to enable a wide range of devices and services to be connected using cellular telecommunications bands.

The Cloud Mobile Gateway performs gateway functions within the packet core. This gateway will help mobile service providers provision for the growth of mobile broadband, deliver new IoT services, and provide a foundation for 5G.

Nokia chose VoltDB to provide the fast data layer in these products for a number of reasons:

Speed
: Consistent average latency of around one millisecond. VoltDB can make decisions close to the data, via transactions and stored procedures, reducing round trips.

Scale
: Predictable scalability due to a linear relationship between transactions, node count, and CPU core count.

Cloud ready
: A completely virtualizable database to fit into Nokia's telco cloud and NFV infrastructure.

Cost
: The total cost of ownership was lower than with traditional databases.

Use Case: Personalized Services and Offers

Personalization is crucial to the success of many Over-the-Top (OTT) players like Netflix and Hulu. To increase customer satisfaction and reduce churn, service providers need to deliver a real-time, personalized user experience to every subscriber on any device.

Real-time user targeting allows service providers to build new service offerings and promotions to increase revenue. Openet, for example, helps a top US cable provider use audience data to tailor ads to location and content in real-time. Latency needs to be in the millisecond range.

To personalize services and offers, service providers need real-time analytics to monitor and analyze the user session data of millions of users in real time on a per-event, per-person basis. Real-time decision engines must combine streaming data with customer profiles or contextual data to generate personalized responses.

Emagine Case Study

Emagine International is a leading provider of real-time, contextual, and adaptive campaign management software solutions to telecom service providers. Emagine's RED.cloud platform detects events like customers going out of bundle, onto higher rates or running out of credit. It can determine whether a customer is experiencing network latency when downloading an app, dropping calls, or exceeding bandwidth limits while viewing a YouTube video. All this information can be used to trigger personalized offers, rewards, and notifications.

"Our vision was to build a platform that delivers the best interaction possible, aligned to each individual customer in real-time to drive customer engagement and maximize business results," explains Emagine CEO David Peters.

Many of Emagine's current mobile telecommunications prospects already have a Multichannel Campaign Management system that sits on top of a data warehouse and relies on batch processes. Those prospects were averaging a 10-minute response time for a typical near real-time campaign. Emagine wanted to complete the ingest-analyze-decide cycle in less than three milliseconds and deliver customized offers to subscribers in less than 250 milliseconds.

Emagine adopted a Lambda architecture, with VoltDB serving as the fast frontend. RED.cloud ingests real-time transactions such as customer data records (CDRs), network events, URL data, Home Location Register/Visitor Location Register (HLR/VLR) states, and end-of-call events. VoltDB provides real-time analysis of subscriber data based on event triggers such as the end of a call, use of the mobile device in a particular location, or a user hitting a data usage threshold.

Emagine conducted a proof-of-concept with a Tier 1 mobile service provider to quantify whether moving from near-real-time to real-time interaction would increase revenue or reduce churn. Two use cases were analyzed for which Emagine ingested 1.5 billion call and event detail records per day.

In the data bundle resign use case, subscribers were offered a customized new data bundle when they were about to go out of bundle. High out-of-bundle rates were leading to customer dissatisfaction and churn. Real-time offers reduced out-of-bundle usage by over 500 percent over near-real-time offers, and real-time data bundle sales increased by 50 percent.

The airtime advance use case identified customers who were about to run out of credit on prepaid services and offered them an airtime advance—an IOU credit—to encourage those customers to continue to use the network. Subscribers receiving tailored, real-time offers bought 253 percent more airtime advance services than those who received near real-time offers. As the operator implements this use case across its entire subscriber base, it is projected to generate incremental airtime advance fees of $30,000 per month.

VoltDB offered Emagine as well as its service provider customers several advantages:

Speed
: VoltDB allowed Emagine to complete the ingest-analyze-decide in less than 250 milliseconds, moving offer generation from near real time to real time.

Scale
: VoltDB could deal with the scale of data required to generate real-time offers; for example, 1.5 billion call and event detail records per day, in a single proof of concept.

Cost
: Emagine generates new revenue for service providers. In the airtime advance use case alone, the operator could generate $30,000 per month more than with near-real-time offers.

Use Case: IoT

Gartner predicts 20.4 billion IoT devices will be in the field by 2020. IoT will affect almost every sector of the economy, from health care to automotive, smart cities to transportation, energy to farming. Whether it's speeding up a production line or instructing vendors to increase stock in a distribution warehouse, IoT applications need the ability to automate real-time decisions.

"Most of your readings are going to be similar and within a safe range," says VoltDB's Dheeraj Remella, "But to detect the anomaly, it's the needle in the haystack problem. You have to look at every piece of hay and perhaps compare the incoming data with some KPI. Once you make a decision that something anomalous has happened, or something interesting has happened, you need to act on it."

IoT fast data applications must perform all four functions of fast data at massive scale. Incoming events must be enriched with static metadata like the current device state, the last known device location, the last valid reading, the current firmware version or installed location. Big data analytics like thresholds, profiles, and models are combined with incoming sensor data and contextual metadata in order to make decisions. Alerts, alarms, and policy decisions from decisions must be exported to downstream systems and incoming IoT sensor data to big data systems (see Table 1-2).

Table 1-2. Data types in IoT fast data applications

Type	Real-time decisions	Real-time ETL	Real-time analytics/SQL caching
Input feed	Personalization, real-time scoring requests	Sensor data, M2M, IoT	Real-time feed being observed for operational intelligence

Type	Real-time decisions	Real-time ETL	Real-time analytics/SQL caching
Event metadata	Policy parameters; POI, user profiles	Metadata about the sensors infrastructure (versions, locations, and so on)	
Big data analytic outputs	Scoring rubrics; user segmentation profile	Interpolation parameters; min/max threshold validation parameters	OLAP report results in "SQL Caching" use cases.
Event responses and alerts	Decisions and customization results	Alerts/notifications on exceptional events (or exceptional sequences of events)	Dashboard and BI query responses. Counters, leaderboards, aggregates, and time-series groupings for operational monitoring
Output feed	Archive of transaction stream for historical analytics	Enriched, filtered, processed event feed handed downstream	

IoT also raises the issue of *where* computation happens. In fog and edge computing, computation moves from the cloud to the edge of the network, whereas big data analysis is still performed in the cloud. Edge computing pushes intelligence, processing power, and communication capabilities directly into IoT devices. Using edge computing, industrial IoT systems could use device sensors and actuators to monitor production environments, initiate processes, and respond to anomalies locally.

Fog computing pushes intelligence down to the local area network–level of network architecture, processing data in a fog node or IoT gateway. At the fog level, fast data applications often need to find correlations at the plant level between multiple incoming sensor streams. For example, in a power plant use case, all devices reside in a single location and act cohesively, so they influence each other.

Nimble Storage Case Study

Nimble storage is a flash storage vendor that in 2017 was acquired by Hewlett Packard Enterprise (HPE) for $1.09 billion. Nimble's InfoSight Predictive Analytics platform predicts, diagnoses, and prevents latency and performance problems across host, network, and storage layers, as well as identifying future capacity needs. It can resolve the detected problems automatically.

InfoSight collects and analyzes billions of sensor data points from each storage array. It also gathers data on the IT technology stack above the storage array all the way up to the virtual machine. According to Nimble's findings, 54 percent of application performance problems identified by InfoSight do not in fact come from the storage.

Infosight uses HPE's big data analytics solution, Vertica, to perform machine learning on sensor, log, and configuration data and build predictive maintenance models. VoltDB applies those models to correlated time–series events from multiple sensor streams in order to identify potential problems in real time. This is an example of fog computing.

Dheeraj Remella of VoltDB expands on this:

> All of these individual arrays are reporting their separate readings. We help them correlate on a time basis and on a model basis, and make decisions on what is happening. You have complex policies codified into VoltDB to orchestrate between several segments of your IoT deployment. That kind of decision making needs to happen locally, not in the cloud.

After a problem is identified, InfoSight generates a support ticket and recommends actions. InfoSight automatically detects 90 percent of all issues within a customer's infrastructure and resolves more than 80 percent of them in an automated fashion.

Nimble Storage selected VoltDB for the following reasons:

Speed
: Fast performance and high throughput was critical for Nimble Storage.

Scale
: IoT use cases like Nimble's involve huge volumes of data.

Integration with big data
: Tight integration with Vertica was essential for Nimble's use case. VoltDB cached predictive models from Vertica in a fast SQL query cache.

Building a Fast Data Stack for Telco

"The problem of real time computing which a lot of developers fail to appreciate, at least initially, is you've only got so much control over events," says David Rolfe, director of solutions engineering EMEA at VoltDB, "The world is happening all around you."

A fast data stack (Figure 1-2) must ingest, analyze, act upon, and export fast data while meeting the stringent nonfunctional requirements of telco fast data use cases. Three categories of technologies have been proposed as possible solution components for the fast data stack.

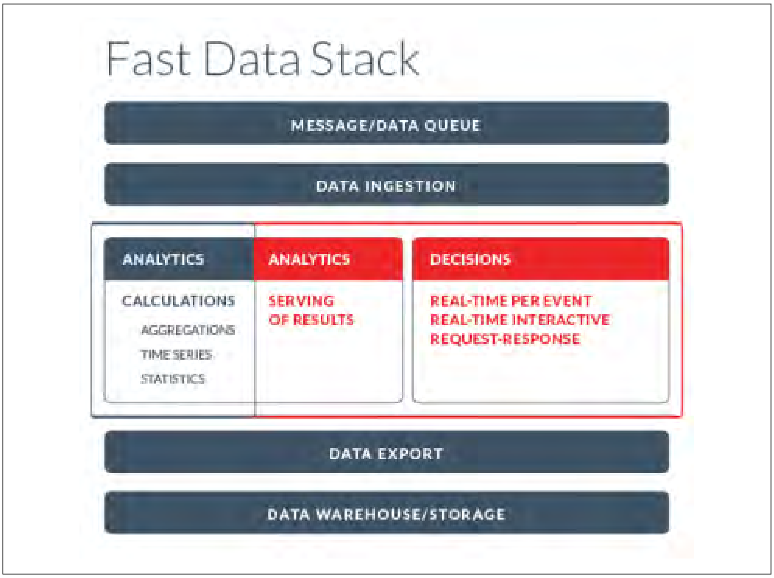

Figure 1-2. The fast data stack

Fast OLAP Systems

OLAP solutions enable fast queries against data at rest. Fast OLAP systems organize data to enable efficient queries across multiple dimensions of terabytes to petabytes of stored data. OLAP solutions can perform analytics on data at rest, but cannot generate real-time responses and decisions on streaming data (Figure 1-3).

FAST OLAP	CEP/STREAMING	FAST OLTP
• Cannot generate real-time responses and decisions. • An evolution of OLAP capability to in-memory; not a sustainable strategic value-add vs. columnar incumbents (Vertica, Redshift, etc.) • May have poor SQL support – not a substitute to incumbent column stores (MemSQL)	• Good at data capture. • Good at ingest to OLAP and pre-defined read-only analytics. • Other operations introduce complexity and lack of reliability. May require development work; needs a back-end database.	• Fast Data is message/event oriented and databases are query/transaction oriented. Put messages first and support with DB where necessary. • Suitable for analytics with pre-defined queries. • No compression: too expensive to store large datasets. May require complex OLAP integrations.

Figure 1-3. Fast data solution components

Stream Processing Systems

Streaming systems are optimized for running computations across a stream of incoming events. They can calculate real-time analytics and enable real-time Extract, Transform, and Load (ETL) operations. However, real-time analytics results like counts, aggregations, and leaderboards still need to be stored in an external backend storage system.

Stream processing systems integrate well with big data systems—they often are used as on-ramps to OLAP—but they cannot enrich streaming data with the context and state needed to make decisions. For this reason, stream processing systems are often combined with a backend database but bolting on a database results in lower performance and higher latency.

Online Transaction Processing Database Systems

Online Transaction Processing (OLTP) systems are operational databases: traditional SQL systems, some NoSQL offerings, and NewSQL architectures. Traditional database systems support per-event decision-making that is informed by other stored data, but historically have been unable to meet the performance requirements of fast data.

Both NewSQL and NoSQL solutions supply the speed, scale, and availability required by fast data applications. However, NoSQL solutions generally lack transactionality and query capabilities. In addition, most NoSQL databases are eventually consistent by design, so they are not suitable for telco use cases like charging, which require total accuracy. NewSQL solutions, on the other hand, offer SQL queries and strong consistency.

DIY and Open Source Stacks

To build a fast data stack, developers often stitch together a number of open source projects like Kafka, Storm, Spark, and a NoSQL database with glue code or use open source stacks like SMACK (Spark, Mesos, Akka, Cassandra, Kafka).

Latency performance problems often occur when you have a multi-tiered architecture. The entire system needs to recover quickly and gracefully from failures. For example, with the SMACK stack, what happens if Cassandra fails but Kafka keeps streaming in data? What if both Cassandra and Kafka fail? What should Spark do to ensure that you don't lose data in-flight?

For telco service providers who are used to highly reliable, built-for-purpose traditional OSS/BSS stacks, the maintenance work required can come as a shock. Each time one of the open source components releases a new version, at a minimum the entire system needs to be retested and glue code may need to be updated.

"Supporting open source is most definitely not free," says VoltDB's Rolfe, "We use the phrase 'the previously embittered', where people show up and know exactly what they want because they've tried it three times before."

A Unified Solution for Telco Fast Data

VoltDB is an example of a unified approach, providing all four fast data functions within a single product. "We're able to ingest data, process data, make decisions, aggregate KPIs, store that data, export notifications, and export to your archival storage," says Dheeraj Remella, "These are all the things that need to be done on fast data, and we put them all together in one platform."

Ingest

VoltDB can ingest multiple streams of data at wire speed and transact on each event.

Analyze

VoltDB can maintain real-time analytics on the incoming stream, store results, and make them accessible to decision engines.

Act

VoltDB can store the stateful metadata that decision engines need to combine with streaming data. VoltDB can cache big data analytics and make them accessible to decision engines.

Policies and decision rules can be encapsulated in stored procedures to speed up processing.

Export

VoltDB integrates well with big data systems. VoltDB is often used as a fast frontend for Hadoop, for example, to filter, dedupe, aggregate, enrich, and denormalize streaming data before it comes to rest. Identifying the most valuable subset of streaming data helps to reduce data management costs further down the line.

VoltDB integrates easily with a messaging queue like Kafka to export event responses and the results of decisions.

Nonfunctional telco requirements

VoltDB supports the scale, speed, and accuracy that Telcos require in a cloud-native, cost-effective system. Here's how:

Speed
: VoltDB combines the performance of an in-memory database with ACID transaction support to create a processing environment capable of fast, per-event decisions. Moving transaction processing into memory and eliminating client round trips with stored procedures reduces the running time of transactions in the database, further improving throughput.

Scale
: VoltDB can deal with the scale of telco data, as shown in production deployments and demonstrations like Openet's one trillion transactions per day.

Cloud ready
: VoltDB is the only virtualizable relational database that meets very strict carrier grade requirements, such as average one milli-

second response time, predictable low latency, and optimal resource allocation.

Immediately consistent
: VoltDB is immediately consistent, so it's suitable for telco use cases like charging and policy management.

Cost effective
: VoltDB has more flexible licensing than traditional database vendors, and has a much smaller hardware footprint than both traditional vendors and open source NoSQL databases like Cassandra. VoltDB's simplicity also lowers operational costs compared with both traditional database systems and open source fast data stacks.

Fast Data for All

VoltDB's ultimate purpose is to enable a telco service provider to do business better: to build a flexible, cost-effective network, to delight customers with personalized services, and to enable transformative new services like IoT.

"We don't make the software; we make the software better," declares Pogany, "We don't make the analytics; we make the analytics better. We don't make the network; we make the network better. That's really how you would think about VoltDB. We're a magic ingredient inside of a technology stack."

About the Author

Lapsed software developer, current tech journalist, and wannabe data scientist, **Ciara Byrne** started her career in academic machine learning research, was the CTO of a security startup, and managed suites of software products as well as building her own.

Her writing has appeared in *Fast Company*, *Forbes*, *MIT Technology Review*, *VentureBeat*, *O'Reilly Radar*, *TechCrunch*, and the *New York Times Digital*.