# Real-Time Big Data Analytics: Emerging Architecture

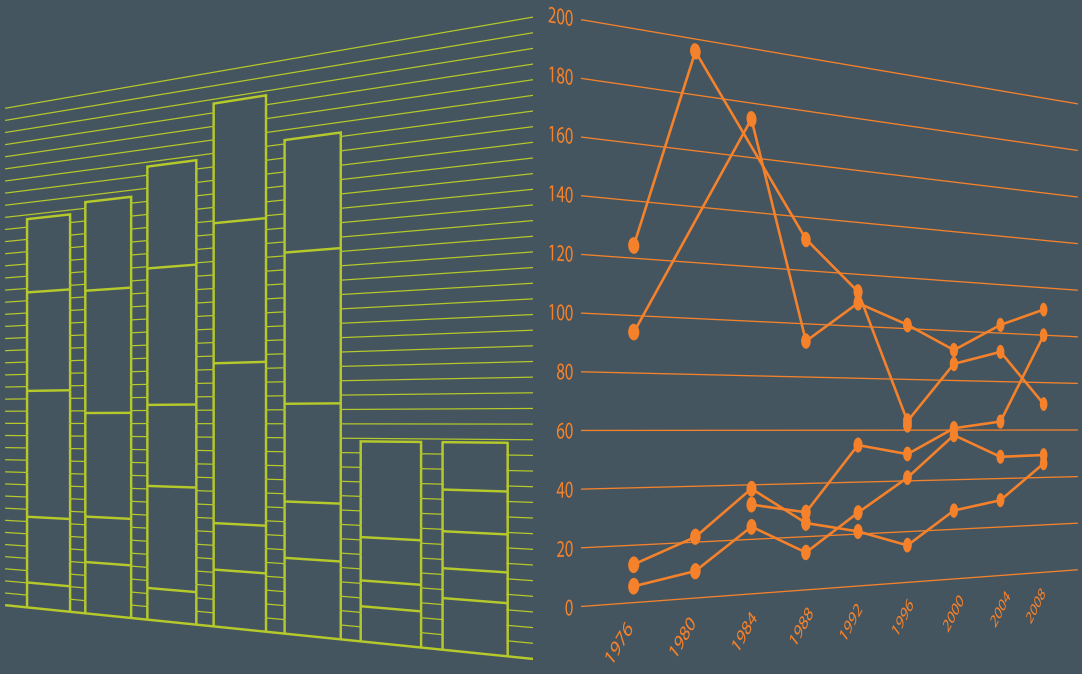
**Mike Barlow**


O'REILLY®

O'REILLY®
Strata
Making Data Work

# SLEEK & SMART

## Big Data Analytics




pentahobigdata.com

pentaho

# Real-Time Big Data Analytics: Emerging Architecture

*Mike Barlow*

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

**Real-Time Big Data Analytics: Emerging Architecture**

by Mike Barlow



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://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

February 2013: First Edition

**Revision History for the First Edition:**

2013-02-25 First release

See *http://oreilly.com/catalog/errata.csp?isbn=9781449364212* for release details.



ISBN: 978-1-449-36421-2

# Table of Contents

# CHAPTER 1
# Introduction

Imagine that it's 2007. You're a top executive at major search engine company, and Steve Jobs has just unveiled the iPhone. You immediately ask yourself, "Should we shift resources away from some of our current projects so we can create an experience expressly for iPhone users?" Then you begin wondering, "What if it's all hype? Steve is a great showman … how can we predict if the iPhone is a fad or the next big thing?"

The good news is that you've got plenty of data at your disposal. The bad news is that you have no way of querying that data and discovering the answer to a critical question: How many people are accessing my sites from their iPhones?

Back in 2007, you couldn't even ask the question without upgrading the schema in your data warehouse, an expensive process that might have taken two months. Your only choice was to wait and hope that a competitor didn't eat your lunch in the meantime.

Justin Erickson, a senior product manager at Cloudera, told me a version of that story and I wanted to share it with you because it neatly illustrates the difference between traditional analytics and real-time big data analytics. Back then, you had to know the kinds of questions you planned to ask before you stored your data.

"Fast forward to the present and technologies like Hadoop give you the scale and flexibility to store data before you know how you are going to process it," says Erickson. "Technologies such as MapReduce, Hive and Impala enable you to run queries without changing the data structures underneath."

Today, you are much less likely to face a scenario in which you cannot query data and get a response back in a brief period of time. Analytical processes that used to require month, days, or hours have been reduced to minutes, seconds, and fractions of seconds.

But shorter processing times have led to higher expectations. Two years ago, many data analysts thought that generating a result from a query in less than 40 minutes was nothing short of miraculous. Today, they expect to see results in under a minute. That's practically the speed of thought — you think of a query, you get a result, and you begin your experiment.

"It's about moving with greater speed toward previously unknown questions, defining new insights, and reducing the time between when an event happens somewhere in the world and someone responds or reacts to that event," says Erickson.

A rapidly emerging universe of newer technologies has dramatically reduced data processing cycle time, making it possible to explore and experiment with data in ways that would not have been practical or even possible a few years ago.

Despite the availability of new tools and systems for handling massive amounts of data at incredible speeds, however, the real promise of advanced data analytics lies beyond the realm of pure technology.

"Real-time big data isn't just a process for storing petabytes or exabytes of data in a data warehouse," says Michael Minelli, co-author of *Big Data, Big Analytics*. "It's about the ability to make better decisions and take meaningful actions at the right time. It's about detecting fraud while someone is swiping a credit card, or triggering an offer while a shopper is standing on a checkout line, or placing an ad on a website while someone is reading a specific article. It's about combining and analyzing data so you can take the right action, at the right time, and at the right place."

For some, real-time big data analytics (RTBDA) is a ticket to improved sales, higher profits and lower marketing costs. To others, it signals the dawn of a new era in which machines begin to think and respond more like humans.

# CHAPTER 2
# How Fast Is Fast?

The capability to store data quickly isn't new. What's new is the capability to do something meaningful with that data, quickly and cost-effectively. Businesses and governments have been storing huge amounts of data for decades. What we are witnessing now, however, is an explosion of new techniques for analyzing those large data sets. In addition to new capabilities for handling large amounts of data, we're also seeing a proliferation of new technologies designed to handle complex, non-traditional data — precisely the kinds of unstructured or semi-structured data generated by social media, mobile communications, customer service records, warranties, census reports, sensors, and web logs. In the past, data had to be arranged neatly in tables. In today's world of data analytics, anything goes. Heterogeneity is the new normal, and modern data scientists are accustomed to hacking their way through tangled clumps of messy data culled from multiple sources.

Software frameworks such as Hadoop and MapReduce, which support distributed processing applications across relatively inexpensive commodity hardware, now make it possible to mix and match data from many disparate sources. Today's data sets aren't merely larger than the older data sets — they're significantly more complex.

"Big data has three dimensions — volume, variety, and velocity," says Minelli. "And within each of those three dimensions is a wide range of variables."

The ability to manage large and complex sets of data hasn't diminished the appetite for more size and greater speed. Every day it seems that a new technique or application is introduced that pushes the edges of the speed-size envelope even further.

Druid, for example, is a system for scanning tens of billions of records per second. It boasts scan speeds of 33 million rows/second/core and ingest speeds of 10 thousand records/second/node. It can query 6 terabytes of in-memory data in 1.4 seconds. As Eric Tschetter wrote in his blog, Druid has "the power to move planetary-size data sets with speed."

When systems operate at such blinding velocities, it seems odd to quibble over a few milliseconds here or there. But Ted Dunning, an architect at MapR Technologies, raises a concern worth noting. "Many of the terms used by people are confusing. Some of the definitions are what I would call *squishy*. They don't say, *If this takes longer than 2.3 seconds, we're out*. Google, for instance, definitely wants their system to be as fast as possible and they definitely put real-time constraints on the internals of their system to make sure that it gives up on certain approaches very quickly. But overall, the system itself is not real time. It's pretty fast, almost all the time. That's what I mean by a *squishy* definition of real time."

The difference between a hard definition and a "squishy" definition isn't merely semantic — it has real-world consequences. For example, many people don't understand that real-time online algorithms are constrained by time and space limitations. If you "unbound" them to allow more data, they can no longer function as real-time algorithms. "People need to begin developing an intuition about which kinds of processing are bounded in time, and which kinds aren't," says Dunning. For example, algorithms that keep unique identifiers of visitors to a website can break down if traffic suddenly increases. Algorithms designed to prevent the same email from being resent within seven days through a system work well until the scale of the system expands radically.

The Apache Drill project will address the "squishy" factor by scanning through smaller sets of data very quickly. Drill is the open source cousin of Dremel, a Google tool that rips through larger data sets at blazing speeds and spits out summary results, sidestepping the scale issue.

Dunning is one of the Drill project's core developers. He sees Drill as complementary to existing frameworks such as Hadoop. Drill brings big data analytics a step closer to real-time interactive processing, which is definitely a step in the right direction.

"Drill takes a slightly different tack than Dremel," says Dunning. "Drill is trying to be more things to more people — probably at the cost of some performance, but that's just mostly due to the different environment. Google is a well-controlled, very well managed data environment. But the outside world is a messy place. Nobody is in charge. Data appears in all kinds of ways and people have all kinds of preferences for how they want to express what they want, and what kinds of languages they want to write their queries in."

Dunning notes that both Drill and Dremel scan data in parallel. "Using a variety of online algorithms, they're able to complete scans — doing filtering operations, doing aggregates, and so on — in a parallel way, in a fairly short amount of time. But they basically scan the whole table. They are both full-table scan tools that perform good aggregation, good sorting, and good *top 40* sorts of measurements."

In many situations involving big data, random failures and resulting data loss can become issues. "If I'm bringing data in from many different systems, data loss could skew my analysis pretty dramatically," says Cloudera's Erickson. "When you have lots of data moving across multiple networks and many machines, there's a greater chance that something will break and portions of the data won't be available."

Cloudera has addressed those problems by creating a system of tools, including Flume and SQOOP, which handle ingestion from multiple sources into Hadoop, and Impala, which enables real-time, ad hoc querying of data.

"Before Imapala, you did the machine learning and larger-scale processes in Hadoop, and the ad hoc analysis in Hive, which involves relatively slow batch processing," says Erickson. "Alternatively, you can perform the ad-hoc analysis against a traditional database system, which limits your ad-hoc exploration to the data that is captured and loaded into the pre-defined schema. So essentially you are doing machine learning on one side, ad hoc querying on the other side, and then correlating the data between the two systems."

Impala, says Erickson, enables ad hoc SQL analysis "directly on top of your big data systems. You don't have to define the schema before you load the data."

For example, let's say you're a large financial services institution. Obviously, you're going to be on the lookout for credit card fraud. Some kinds of fraud are relatively easy to spot. If a cardholder makes a purchase in Philadelphia and another purchase 10-minutes later in San Diego, a fraud alert is triggered. But other kinds of credit card fraud involve numerous small purchases, across multiple accounts, over long time periods.

Finding those kinds of fraud requires different analytical approaches. If you are running traditional analytics on top of a traditional enterprise data warehouse, it's going to take you longer to recognize and respond to new kinds of fraud than it would if you had the capabilities to run ad hoc queries in real time. When you're dealing with fraud, every lost minute translates into lost money.

# CHAPTER 3
# How Real Is Real Time?

Here's another complication: The meaning of "real time" can vary depending on the context in which it is used.

"In the same sense that there really is no such thing as truly unstructured data, there's no such thing as real time. There's only near-real time," says John Akred, a senior manager within the data domain of Accenture's Emerging Technology Innovations group. "Typically when we're talking about real-time or near real-time systems, what we mean is architectures that allow you to respond to data as you receive it without necessarily persisting it to a database first."

In other words, real-time denotes the ability to process data as it arrives, rather than storing the data and retrieving it at some point in the future. That's the primary significance of the term — real-time means that you're processing data in the present, rather than in the future.

But "the present" also has different meanings to different users. From the perspective of an online merchant, "the present" means the attention span of a potential customer. If the processing time of a transaction exceeds the customer's attention span, the merchant doesn't consider it real time.

From the perspective of an options trader, however, real time means milliseconds. From the perspective of a guided missile, real time means microseconds.

For most data analysts, real time means "pretty fast" at the data layer and "very fast" at the decision layer. "Real time is for robots," says Joe

Hellerstein, chancellor's professor of computer science at UC Berkeley. "If you have people in the loop, it's not real time. Most people take a second or two to react, and that's plenty of time for a traditional transactional system to handle input and output."

That doesn't mean that developers have abandoned the quest for speed. Supported by a Google grant, Matei Zaharia is working on his Ph.D. at UC Berkeley. He is an author of Spark, an open source cluster computing system that can be programmed quickly and runs fast. Spark relies on "resilient distributed datasets" (RDDs) and "can be used to interactively query 1 to 2 terabytes of data in less than a second."

In scenarios involving machine learning algorithms and other multipass analytics algorithms, "Spark can run 10x to 100x faster than Hadoop MapReduce," says Zaharia. Spark is also the engine behind Shark, a data warehousing system.

According to Zaharia, companies such as Conviva and Quantifind have written UIs that launch Spark on the back end of analytics dashboards. "You see the statistics on a dashboard and if you're wondering about some data that hasn't been computed, you can ask a question that goes out to a parallel computation on Spark and you get back an answer in about half a second."

Storm is an open source low latency processing stream processing system designed to integrate with existing queuing and bandwidth systems. It is used by companies such as Twitter, the Weather Channel, Groupon and Ooyala. Nathan Marz, lead engineer at BackType (acquired by Twitter in 2011), is the author of Storm and other open-source projects such as Cascalog and ElephantDB.

"There are really only two paradigms for data processing: batch and stream," says Marz. "Batch processing is fundamentally high-latency. So if you're trying to look at a terabyte of data all at once, you'll never be able to do that computation in less than a second with batch processing."

Stream processing looks at smaller amounts of data as they arrive. "You can do intense computations, like parallel search, and merge queries on the fly," says Marz. "Normally if you want to do a search query, you need to create search indexes, which can be a slow process on one machine. With Storm, you can stream the process across many machines, and get much quicker results."

Twitter uses Storm to identify trends in near real time. Ideally, says Marz, Storm will also enable Twitter to "understand someone's intent in virtually real time. For example, let's say that someone tweets that he's going snowboarding. Storm would help you figure out which ad would be most appropriate for that person, at just the right time."

Storm is also relatively user friendly. "People love Storm because it's easy to use. It solves really hard problems such as fault tolerance and dealing with partial failures in distributed processing. We have a platform you can build on. You don't have to focus on the infrastructure because that work has already been done. You can set up Storm by yourself and have it running in minutes," says Marz.

# CHAPTER 4
# The RTBDA Stack

At this moment, it's clear that an architecture for handling RTBDA is slowly emerging from a disparate set of programs and tools. What isn't clear, however, is what that architecture will look like. One goal of this paper is sketching out a practical RTBDA roadmap that will serve a variety of stakeholders including users, vendors, investors, and corporate executives such as CIOs, CFOs and COOs who make or influence purchasing decisions around information technology.

Focusing on the stakeholders and their needs is important because it reminds us that the RTBDA technology exists for a specific purpose: creating value from data. It is also important to remember that "value" and "real time" will suggest different meanings to different subsets of stakeholders. There is presently no one-size-fits-all model, which makes sense when you consider that the interrelationships among people, processes and technologies within the RTBDA universe are still evolving.

David Smith writes a popular blog for Revolution Analytics on open source R, a programming language designed specifically for data analytics. He proposes a four-layer RTBDA technology stack. Although his stack is geared for predictive analytics, it serves as a good general model:

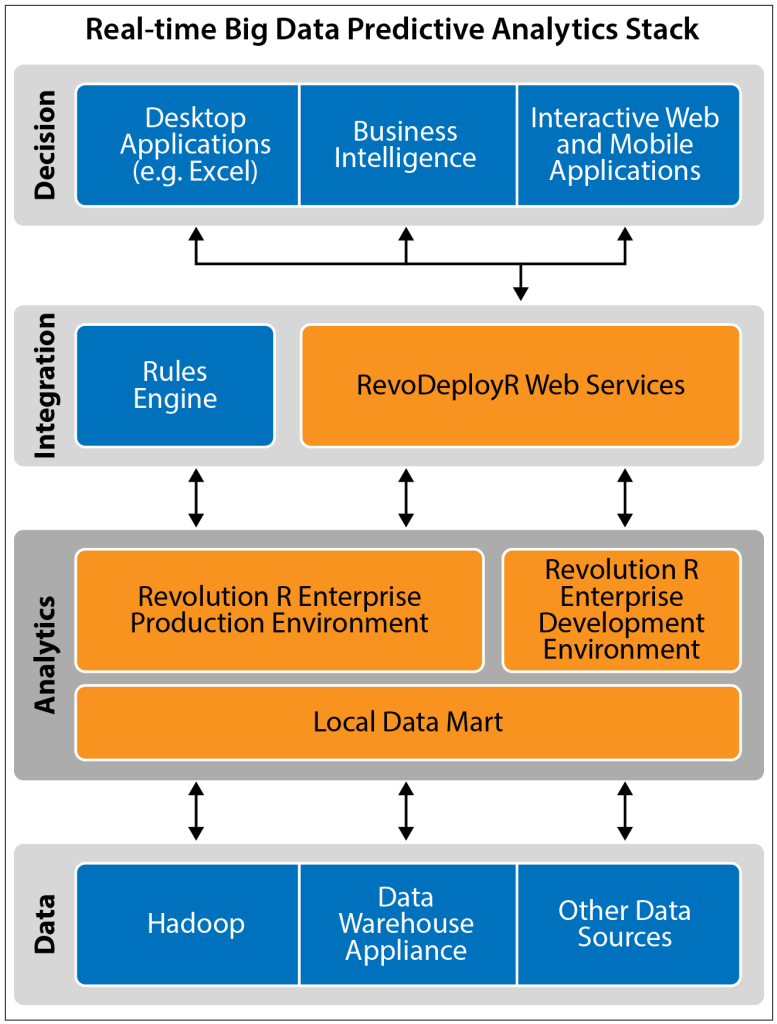



*Figure 4-1. From David Smith's presentation, "Real-Time Big Data Analytics: From Deployment To Production"*

At the foundation is the *data layer*. At this level you have structured data in an RDBMS, NoSQL, Hbase, or Impala; unstructured data in Hadoop MapReduce; streaming data from the web, social media, sensors and operational systems; and limited capabilities for performing descriptive analytics. Tools such as Hive, HBase, Storm and Spark also sit at this layer. (Matei Zaharia suggests dividing the data layer into two layers, one for storage and the other for query processing)

The *analytics layer* sits above the data layer. The analytics layer includes a production environment for deploying real-time scoring and dynamic analytics; a development environment for building models; and a local data mart that is updated periodically from the data layer, situated near the analytics engine to improve performance.

On top of the analytics layer is the *integration layer*. It is the "glue" that holds the end-user applications and analytics engines together, and it usually includes a rules engine or CEP engine, and an API for dynamic analytics that "brokers" communication between app developers and data scientists.

The topmost layer is the *decision layer*. This is where the rubber meets the road, and it can include end-user applications such as desktop, mobile, and interactive web apps, as well as business intelligence software. This is the layer that most people "see." It's the layer at which business analysts, c-suite executives, and customers interact with the real-time big data analytics system.

Again, it's important to note that each layer is associated with different sets of users, and that different sets of users will define "real time" differently. Moreover, the four layers aren't passive lumps of technologies — each layer enables a critical phase of real-time analytics deployment.

# CHAPTER 5
# The Five Phases of Real Time

Real-time big data analytics is an iterative process involving multiple tools and systems. Smith says that it's helpful to divide the process into five phases: data distillation, model development, validation and deployment, real-time scoring, and model refresh. At each phase, the terms "real time" and "big data" are fluid in meaning. The definitions at each phase of the process are not carved into stone. Indeed, they are context dependent. Like the technology stack discussed earlier, Smith's five-phase process model is devised as a framework for predictive analytics. But it also works as a general framework for real-time big data analytics.

1. **Data distillation** — Like unrefined oil, data in the data layer is crude and messy. It lacks the structure required for building models or performing analysis. The data distillation phase includes extracting features for unstructured text, combining disparate data sources, filtering for populations of interest, selecting relevant features and outcomes for modeling, and exporting sets of distilled data to a local data mart.
2. **Model development** — Processes in this phase include feature selection, sampling and aggregation; variable transformation; model estimation; model refinement; and model benchmarking. The goal at this phase is creating a predictive model that is powerful, robust, comprehensible and implementable. The key requirements for data scientists at this phase are speed, flexibility,

productivity, and reproducibility. These requirements are critical in the context of big data: a data scientist will typically construct, refine and compare dozens of models in the search for a powerful and robust real-time algorithm.

3. **Validation and deployment** — The goal at this phase is testing the model to make sure that it works in the real world. The validation process involves re-extracting fresh data, running it against the model, and comparing results with outcomes run on data that's been withheld as a validation set. If the model works, it can be deployed into a production environment.
4. **Real-time scoring** — In real-time systems, scoring is triggered by actions at the decision layer (by consumers at a website or by an operational system through an API), and the actual communications are brokered by the integration layer. In the scoring phase, some real-time systems will use the same hardware that's used in the data layer, but they will not use the same data. At this phase of the process, the deployed scoring rules are "divorced" from the data in the data layer or data mart. Note also that at this phase, the limitations of Hadoop become apparent. Hadoop today is not particularly well-suited for real-time scoring, although it can be used for "near real-time" applications such as populating large tables or pre-computing scores. Newer technologies such as Cloudera's Impala are designed to improve Hadoop's real-time capabilities.
5. **Model refresh** — Data is always changing, so there needs to be a way to refresh the data and refresh the model built on the original data. The existing scripts or programs used to run the data and build the models can be re-used to refresh the models. Simple exploratory data analysis is also recommended, along with periodic (weekly, daily, or hourly) model refreshes. The refresh process, as well as validation and deployment, can be automated using web-based services such as RevoDeployR, a part of the Revolution R Enterprise solution.

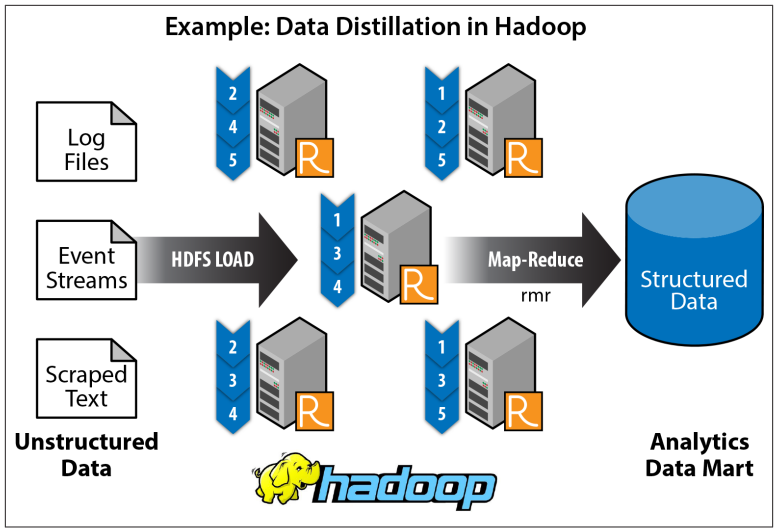



*Figure 5-1. From David Smith's presentation, "Real-Time Big Data Analytics: From Deployment To Production"*

A caveat on the refresh phase: Refreshing the model based on re-ingesting the data and re-running the scripts will only work for a limited time, since the underlying data — and even the underlying structure of the data — will eventually change so much that the model will no longer be valid. Important variables can become non-significant, non-significant variables can become important, and new data sources are continuously emerging. If the model accuracy measure begins drifting, go back to phase 2 and re-examine the data. If necessary, go back to phase 1 and rebuild the model from scratch.

# CHAPTER 6
# How Big Is Big?

As suggested earlier, the "bigness" of big data depends on its location in the stack. At the data layer, it is not unusual to see petabytes and even exabytes of data. At the analytics layer, you're more likely to encounter gigabytes and terabytes of refined data. By the time you reach the integration layer, you're handling megabytes. At the decision layer, the data sets have dwindled down to kilobytes, and we're measuring data less in terms of scale and more in terms of bandwidth.

The takeaway is that the higher you go in the stack, the less data you need to manage. At the top of the stack, size is considerably less relevant than speed. Now we're talking about real-time, and this is where it gets really interesting.

"If you visit the Huffington Post website, for example, you'll see a bunch of ads pop up on the right-hand side of the page," says Smith. "Those ads have been selected for you on the basis of information generated in real time by marketing analytics companies like Upstream Software, which pulls information from a mash up of multiple sources stored in Hadoop. Those ads have to be selected and displayed within a fraction of a second. Think about how often that's happening. Everybody who's browsing the web sees hundreds of ads. You're talking about an incredible number of transactions occurring every second."

# CHAPTER 7
# Part of a Larger Trend

The push toward real-time big data analytics is part of a much larger trend in which the machines we create act less like machines and more like human beings, says Dhiraj Rajaram, Founder and CEO of Mu-Sigma, a provider of decision sciences and analytics solutions.

"Today, most of our technology infrastructure is not designed for real time," says Rajaram, who worked as a strategy consultant at Booz Allen Hamilton and Pricewaterhouse Coopers before launching Mu Sigma. "Our legacy systems are geared for batch processing. We store data in a central location and when we want a piece of information, we have to find it, retrieve it and process it. That's the way most systems work. But that isn't the way the human mind works. Human memory is more like flash memory. We have lots of specific knowledge that's already mapped — that's why we can react and respond much more quickly than most of our machines. Our intelligence is distributed, not highly centralized, so more of it resides at the edge. That means we can find it and retrieve it quicker. Real time is a step toward building machines that respond to problems the way people do."

As information technology systems become less monolithic and more distributed, real-time big data analytics will become less exotic and more commonplace. The various technologies of data science will be industrialized, costs will fall and eventually real-time analytics will become a commodity.

At that point, the focus will shift from data science to the next logical frontier: decision science. "Even if you have the best real-time analytics, you won't be competitive unless you empower the people in the

organization to make the right decisions," says Rajaram. "The creation of analytics and the consumption of analytics are two different things. You need processes for translating the analytics into good decisions. Right now, everyone thinks that analytics technology is sexy. But the real challenge isn't transforming the technology — the real challenge is transforming the people and the processes. That's the hard part."

## About the Author

**Mike Barlow** is an award-winning journalist, author and communications strategy consultant. Since launching his own firm, Cumulus Partners, he has represented major organizations in numerous industries.

Mike is coauthor of *The Executive's Guide to Enterprise Social Media Strategy* (Wiley, 2011) and *Partnering with the CIO: The Future of IT Sales Seen Through the Eyes of Key Decision Makers* (Wiley, 2007).

He is also the writer of many articles, reports, and white papers on marketing strategy, marketing automation, customer intelligence, business performance management, collaborative social networking, cloud computing, and big data analytics.

Over the course of a long career, Mike was a reporter and editor at several respected suburban daily newspapers, including *The Journal News* and the *Stamford Advocate*. His feature stories and columns appeared regularly in *The Los Angeles Times*, *Chicago Tribune*, *Miami Herald*, *Newsday*, and other major U.S. dailies.