CHAPTER 1

What is big data? - O'Reilly Media

Big data is data that exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or doesn't fit the strictures of your database architectures. To gain value from this data, you must choose an alternative way to process it.

The hot IT buzzword of 2012, big data has become viable as cost-effective approaches have emerged to tame the volume, velocity and variability of massive data. Within this data lie valuable patterns and information, previously hidden because of the amount of work required to extract them. To leading corporations, such as Walmart or Google, this power has been in reach for some time, but at fantastic cost. Today's commodity hardware, cloud architectures and open source software bring big data processing into the reach of the less well-resourced. Big data processing is eminently feasible for even the small garage startups, who can cheaply rent server time in the cloud.

The value of big data to an organization falls into two categories: analytical use, and enabling new products. Big data analytics can reveal insights hidden previously by data too costly to process, such as peer influence among customers, revealed by analyzing shoppers' transactions, social and geographical data. Being able to process every item of data in reasonable time removes the troublesome need for sampling and promotes an investigative approach to data, in contrast to the somewhat static nature of running predetermined reports.

The past decade's successful web startups are prime examples of big data used as an enabler of new products and services. For example, by combining a large number of signals from a user's actions and those of their friends, Facebook has been able to craft a highly personalized user experience and create a new kind of advertising business. It's no coincidence that the lion's share of ideas and tools underpinning big data have emerged from Google, Yahoo, Amazon and Facebook.

The emergence of big data into the enterprise brings with it a necessary counterpart: agility. Successfully exploiting the value in big data requires experimentation and exploration. Whether creating new products or looking for ways to gain competitive advantage, the job calls for curiosity and an entrepreneurial outlook.

What does big data look like?

As a catch-all term, "big data" can be pretty nebulous, in the same way that the term "cloud" covers diverse technologies. Input data to big data systems could be chatter from social networks, web server logs, traffic flow sensors, satellite imagery, broadcast audio streams, banking transactions, MP3s of rock music, the content of web pages, scans of government documents, GPS trails, telemetry from automobiles, financial market data, the list goes on. Are these all really the same thing?

To clarify matters, the three Vs of *volume, velocity and variety* are commonly used to characterize different aspects of big data. They're a helpful lens through which to view and understand the nature of the data and the software platforms available to exploit them. Most probably you will contend with each of the Vs to one degree or another.

VOLUME

The benefit gained from the ability to process large amounts of information is the main attraction of big data analytics. Having

more data beats out having better models: simple bits of math can be unreasonably effective given large amounts of data. If you could run that forecast taking into account 300 factors rather than 6, could you predict demand better?

This volume presents the most immediate challenge to conventional IT structures. It calls for scalable storage, and a distributed approach to querying. Many companies already have large amounts of archived data, perhaps in the form of logs, but not the capacity to process it.

Assuming that the volumes of data are larger than those conventional relational database infrastructures can cope with, processing options break down broadly into a choice between massively parallel processing architectures — data warehouses or databases such as *Greenplum* — and *Apache Hadoop*-based solutions. This choice is often informed by the degree to which the one of the other "Vs" — variety — comes into play. Typically, data warehousing approaches involve predetermined schemas, suiting a regular and slowly evolving dataset. Apache Hadoop, on the other hand, places no conditions on the structure of the data it can process.

At its core, Hadoop is a platform for distributing computing problems across a number of servers. First developed and released as open source by Yahoo, it implements the *MapReduce* approach pioneered by Google in compiling its search indexes. Hadoop's MapReduce involves distributing a dataset among multiple servers and operating on the data: the "map" stage. The partial results are then recombined: the "reduce" stage.

To store data, Hadoop utilizes its own distributed filesystem, HDFS, which makes data available to multiple computing nodes. A typical Hadoop usage pattern involves three stages:

- loading data into HDFS,
- MapReduce operations, and
- retrieving results from HDFS.

This process is by nature a batch operation, suited for analytical or non-interactive computing tasks. Because of this, Hadoop is not itself a database or data warehouse solution, but can act as an analytical adjunct to one.

One of the most well-known Hadoop users is Facebook, whose model follows this pattern. A MySQL database stores the core data. This is then reflected into Hadoop, where computations occur, such as creating recommendations for you based on your friends' interests. Facebook then transfers the results back into MySQL, for use in pages served to users.

VELOCITY

The importance of data's velocity — the increasing rate at which data flows into an organization — has followed a similar pattern to that of volume. Problems previously restricted to segments of industry are now presenting themselves in a much broader setting. Specialized companies such as financial traders have long turned systems that cope with fast moving data to their advantage. Now it's our turn.

Why is that so? The Internet and mobile era means that the way we deliver and consume products and services is increasingly instrumented, generating a data flow back to the provider. Online retailers are able to compile large histories of customers' every click and interaction: not just the final sales. Those who are able to quickly utilize that information, by recommending additional purchases, for instance, gain competitive advantage. The smartphone era increases again the rate of data inflow, as consumers carry with them a streaming source of geolocated imagery and audio data.

It's not just the velocity of the incoming data that's the issue: it's possible to stream fast-moving data into bulk storage for later batch processing, for example. The importance lies in the speed of the feedback loop, taking data from input through to decision. *A commercial from IBM* makes the point that you wouldn't cross the road if all you had was a five-minute old snapshot of traffic

location. There are times when you simply won't be able to wait for a report to run or a Hadoop job to complete.

Industry terminology for such fast-moving data tends to be either "streaming data," or "complex event processing." This latter term was more established in product categories before streaming processing data gained more widespread relevance, and seems likely to diminish in favor of streaming.

There are two main reasons to consider streaming processing. The first is when the input data are too fast to store in their entirety: in order to keep storage requirements practical some level of analysis must occur as the data streams in. At the extreme end of the scale, the Large Hadron Collider at CERN generates so much data that scientists must discard the overwhelming majority of it — hoping hard they've not thrown away anything useful. The second reason to consider streaming is where the application mandates immediate response to the data. Thanks to the rise of mobile applications and online gaming this is an increasingly common situation.

Product categories for handling streaming data divide into established proprietary products such as IBM's *InfoSphere Streams*, and the less-polished and still emergent open source frameworks originating in the web industry: Twitter's *Storm*, and Yahoo *S4*.

As mentioned above, it's not just about input data. The velocity of a system's outputs can matter too. The tighter the *feedback loop*, the greater the competitive advantage. The results might go directly into a product, such as Facebook's recommendations, or into dashboards used to drive decision-making.

It's this need for speed, particularly on the web, that has driven the development of key-value stores and columnar databases, optimized for the fast retrieval of precomputed information. These databases form part of an umbrella category known as *NoSQL*, used when relational models aren't the right fit.

VARIETY

Rarely does data present itself in a form perfectly ordered and ready for processing. A common theme in big data systems is that the source data is diverse, and doesn't fall into neat relational structures. It could be text from social networks, image data, a raw feed directly from a sensor source. None of these things come ready for integration into an application.

Even on the web, where computer-to-computer communication ought to bring some guarantees, the reality of data is messy. Different browsers send different data, users withhold information, they may be using differing software versions or vendors to communicate with you. And you can bet that if part of the process involves a human, there will be error and inconsistency.

A common use of big data processing is to take unstructured data and extract ordered meaning, for consumption either by humans or as a structured input to an application. One such example is entity resolution, the process of determining exactly what a name refers to. Is this city London, England, or London, Texas? By the time your business logic gets to it, you don't want to be guessing.

The process of moving from source data to processed application data involves the loss of information. When you tidy up, you end up throwing stuff away. This underlines a principle of big data: *when you can, keep everything*. There may well be useful signals in the bits you throw away. If you lose the source data, there's no going back.

Despite the popularity and well understood nature of relational databases, it is not the case that they should always be the destination for data, even when tidied up. Certain data types suit certain classes of database better. For instance, documents encoded as XML are most versatile when stored in a dedicated XML store such as *MarkLogic*. Social network relations are graphs by nature, and graph databases such as *Neo4J* make operations on them simpler and more efficient.

Even where there's not a radical data type mismatch, a disadvantage of the relational database is the static nature of its schemas. In an agile, exploratory environment, the results of computations will evolve with the detection and extraction of more signals. Semi-structured NoSQL databases meet this need for flexibility: they provide enough structure to organize data, but do not require the exact schema of the data before storing it.

In practice

We have explored the nature of big data, and surveyed the landscape of big data from a high level. As usual, when it comes to deployment there are dimensions to consider over and above tool selection.

CLOUD OR IN-HOUSE?

The majority of big data solutions are now provided in three forms: software-only, as an appliance or cloud-based. Decisions between which route to take will depend, among other things, on issues of data locality, privacy and regulation, human resources and project requirements. Many organizations opt for a hybrid solution: using on-demand cloud resources to supplement in-house deployments.

BIG DATA IS BIG

It is a fundamental fact that data that is too big to process conventionally is also too big to transport anywhere. IT is undergoing an inversion of priorities: it's the program that needs to move, not the data. If you want to analyze data from the U.S. Census, it's a lot easier to run your code on Amazon's web services platform, which *hosts such data locally*, and won't cost you time or money to transfer it.

Even if the data isn't too big to move, locality can still be an issue, especially with rapidly updating data. Financial trading systems crowd into data centers to get the fastest connection to

source data, because that millisecond difference in processing time equates to competitive advantage.

BIG DATA IS MESSY

It's not all about infrastructure. Big data practitioners consistently report that 80% of the effort involved in dealing with data is cleaning it up in the first place, as Pete Warden observes in his *Big Data Glossary*: "I probably spend more time turning messy source data into something usable than I do on the rest of the data analysis process combined."

Because of the high cost of data acquisition and cleaning, it's worth considering what you actually need to source yourself. *Data marketplaces* are a means of obtaining common data, and you are often able to contribute improvements back. Quality can of course be variable, but will increasingly be a benchmark on which data marketplaces compete.

CULTURE

The phenomenon of big data is closely tied to the emergence of *data science*, a discipline that combines math, programming and scientific instinct. Benefiting from big data means investing in teams with this skillset, and surrounding them with an organizational willingness to understand and use data for advantage.

In his report, "*Building Data Science Teams*," D.J. Patil characterizes data scientists as having the following qualities:

- Technical expertise: the best data scientists typically have deep expertise in some scientific discipline.
- Curiosity: a desire to go beneath the surface and discover and distill a problem down into a very clear set of hypotheses that can be tested.
- Storytelling: the ability to use data to tell a story and to be able to communicate it effectively.

- Cleverness: the ability to look at a problem in different, creative ways.

The far-reaching nature of big data analytics projects can have uncomfortable aspects: data must be broken out of silos in order to be mined, and the organization must learn how to communicate and interpet the results of analysis.

Those skills of storytelling and cleverness are the gateway factors that ultimately dictate whether the benefits of analytical labors are absorbed by an organization. The art and practice of visualizing data is becoming ever more important in bridging the human-computer gap to mediate analytical insight in a meaningful way.

KNOW WHERE YOU WANT TO GO

Finally, remember that big data is no panacea. You can find patterns and clues in your data, but then what? Christer Johnson, IBM's leader for advanced analytics in North America, gives this advice to businesses starting out with big data: first, decide what problem you want to solve.

If you pick a real business problem, such as how you can change your advertising strategy to increase spend per customer, it will guide your implementation. While big data work benefits from an enterprising spirit, it also benefits strongly from a concrete goal.