6 Big Data

What the reader will learn:

- that Big Data is not just about data volumes
- that analysing the data involved is the key to the value of Big Data
- how to use tools like Hadoop to explore large data collections and generate information from data
- that the structured data traditionally stored in a RDBMS is not the only valuable data source
- that a data scientist needs to understand both statistical concepts and the business they are working for

6.1 What Is Big Data?

1 Terabyte = 1024 Gigabytes
1 Petabyte = 1024 Terabytes
1 Exabyte = 1024 Petabytes
1 Zettabyte = 1024 Exabytes

And what does a zettabyte of information look like?

According to a Cisco blog (http://blogs.cisco.com/news/the-dawn-of-the-zettabyte-era-infographic/) a zettabyte is equivalent to about 250 billion DVDs, and that would take one individual a very long time to watch. And the DVD is a good measure since Cisco go on to predict that by 2015 the majority of global Internet traffic (61 percent) will be in some form of video.

So we do mean BIG!

EMC^2 suggest that 1.8 Zettabytes is the amount of data estimated to be created in 2011. Their site has a growth ticker on it, allowing you to see the amount of data created since January 2011 (http://uk.emc.com/leadership/programs/digital-universe.htm). Of course these are estimates, but it helps us get a feel for the scale involved. They go on to suggest that the world's information is more than doubling every two years.

P. Lake, P. Crowther, *Concise Guide to Databases*,
Undergraduate Topics in Computer Science, DOI 10.1007/978-1-4471-5601-7_6,

At the beginning of IBM's guide to what Big Data is they say (http://www01.ibm.com/software/data/bigdata):

> Every day, we create 2.5 quintillion bytes of data—so much that 90 % of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. This data is big data.

But the most obvious trap to fall into is to believe that Big Data, a new term, is only about large volumes of data.

Roger Magoulas from O'Reilly media is credited with the first usage of the term 'Big Data' in the way we have come to understand it, in 2005. But as a distinct, well defined topic, it is younger even than that.

However Springer's Very Large Databases (VLDB) Journal has been in existence since 1992. It

> Examines information system architectures, the impact of technological advancements on information systems, and the development of novel database applications.

Whilst early hard disk drives were relatively small, Mainframes had been dealing with large volumes of data since the 1950s.

So handling large amounts of data isn't new, although the scale has doubtless increased in the last few years. Perhaps it isn't really the actual size, but more to do with whether or not we can meaningfully use and interact with the data? This is what Forrester seem to have in mind with the definition suggested on Mike Gualtieri's blog http://blogs.forrester.com/mike_gualtieri/12-12-05the_pragmatic_definition_of_big_data:

> Big Data is the frontier of a firm's ability to store, process, and access (SPA) all the data it needs to operate effectively, make decisions, reduce risks, and serve customers.

They go on to suggest three questions about big data:

> Store. Can you capture and store the data?
> Process. Can you cleanse, enrich, and analyze the data?
> Access. Can you retrieve, search, integrate, and visualize the data?

The Process question is also part of our definition of Big Data. It is often a presumption that Big Data cannot be handled by the standard, RDBMS-based data systems.

Many references to Big Data also come with the word Analytics tagged along somewhere nearby. Analytics may be quite a new term, but again, it has, in reality been with us for decades, sometimes called Business Analysis, sometimes Data Analysis. We will look at some of the exciting ways Analytics has changed business decisions later in the chapter, but we are really just talking about applying tools and techniques to the large volumes of data now available to an organisation and making some sense of it.

So there seems to be evidence that Big Data is more than just a new buzzword. We can see it as a loose label which covers the storage and accessibility of large volumes of data from multiple sources in a way that allows new information to be gleaned by applying a variety of analytic tools. Interestingly, whilst the phrase "Big Data" appears all over the place on the web, it isn't used much in the jobs market. Instead, people with the required skills are tending to be called Data Scientists.

And now may well be the time for career minded Data Scientists to make a good living. In one McKinsey report (http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation) they suggest:

> There will be a shortage of talent necessary for organizations to take advantage of big data. By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions.

The rest of this chapter is in three sections. We will look at Big Data from a datacentric perspective, then from an analytics perspective, and then finally quickly review some of the tools being used by Data Scientists.

6.2 The Datacentric View of Big Data

The cost of HDD storage has dropped dramatically over the last few decades, from thousands of dollars to fractions of a dollar per Gbyte. The era when data professionals spent much of their time trying to reduce the amount of data stored in an organisation is fast coming to an end. And Cloud will accelerate the trend as it provides an always available, infinitely flexible store of data.

This super-availability of storage is probably one of the key drivers in the upsurge in Big Data. It is certainly true that data production itself has also grown exponentially over the past few years, but if the cost/Gbyte were currently the same as it was in 1990 there can be little doubt that much of this generated data would be discarded as "not worth keeping".

As we shall see in the Analytics section, significant advances in analysis techniques have allowed useful information to be retrieved from a seemingly meaningless pile of raw data. This means that keeping data just in case it is useful is becoming the norm.

To put that into perspective, and remembering that we are talking about zettabytes of data, IMC suggest that there is a massive gap between the total data stored and that being analysed. They suggest that the global picture may be that 23 % of the data stored would be of some use for analysis, but that currently only 0.5 % is actually analysed (http://www.emc.com/collateral/analystreports/idc-the-digital-universe-in-2020.pdf).

6.2.1 The Four Vs

There are many new data specialists currently helping us get an understanding of big data, and they sometimes disagree. It isn't, of course, unusual for experts to disagree, particularly when a topic is relatively new, but it can make it more difficult for the newbie to come to their own understanding.

An example in point is the number of "V's" that should be considered. Some experts hold with three, others four. On the grounds only that it is easier to ignore one, we have decided to go with the four "V's" (Fig. 6.1). Such acronyms are only useful as an aide-memoir after all, so it is what they are describing that matters.

1. **V is for Volume**
   As we have seen, the most obvious characteristic of Big Data is that it is BIG. It may even be so big that an organisation needs to look for new tools and techniques to store and query it. This volume therefore is likely to be the biggest challenge for data professionals in the near future. It may result in the need for scalable distributed storage and querying, and many will turn to Cloud computing to meet those needs.
   Organisations will also recognise that they have historic data in archives that may help provide analysts with longer time-lines of data to look for trends in. These archives may be tape-based and hard to access and making potentially valuable data available will also be an important part of the data professional's job.
2. **V is for Velocity**
   The devices we carry with us everyday, like iPhones, have the ability to stream vast quantities of data of different digital types including geolocation data. A travelling salesman can use an App to report visit information back to head office instantly. Only a couple of decades ago that return of information might have been weekly. Not only is there more data being collected, but it is also being collected instantly.
   But it is the time taken by the decision making cycle that really matters. Gathering the information quickly is of no benefit if we only analyse it once a week. Real-time analytics is discussed later, but is about using very current data to provide information that will help an organisation improve a service, or respond to demand, in a much swifter way.

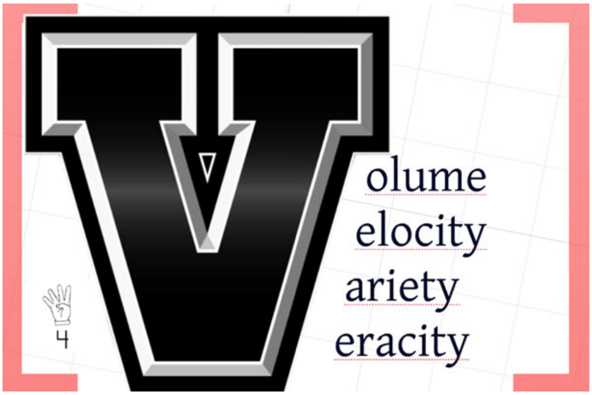


Fig. 6.1 The 4 "V"s

3. **V is for Variety**
   For those with a background in using relational databases the idea that data should be anything other than structured may be difficult to grasp. For most of the past three decades we would have models of all the data that a business needed to keep. Whole methodologies grew around storing known specific data items in the most space efficient way.
   But one of the interesting aspects of Big Data is that we can use data from a wide range of sources. Perhaps the most frequently quoted example is in analysing the mass of readily available social media data to provide companies with information about how their products are being perceived by the public, without the need for focus groups or questionnaires. They collect and then measure the volume of comments and their sentiment. But the details about their products will be needed too, and they may well be stored in a more traditional RDBMS.
   Many data sources are now readily available through open data government portals, or sites like Infochimps. The data itself can be in many different formats, and the ability to cope with CSV, Excel, XML, JSON and many other formats is one of the skills the Data Scientist now needs.
4. **V is for Veracity**
   Data is just a series of ons-or-offs, usually on a magnetic medium, often squirted around the globe in packets of ons-and-offs. Things go wrong! Organisations need to be able to verify the incoming data for accuracy and provenance.
   But as we begin to hear some astonishing helpful outputs from analytics we also need to be aware that data analysis can get things wrong too. The fact is that when you search for patterns in large data sets it is possible that the patterns discovered are entirely caused by chance.
   We also need to attempt to extract some meaning from the variety of inputs so that we can be accurate about their content. When you have no control over the provenance of data it can become very difficult to ensure its accuracy. Let's say you have mined some social media and discovered the phrase: "Life of Pi is the

best film showing in Washington this weekend." Does this mean Washington state, Washington DC or Washington in County Durham in the UK?

6.2.1.1 Non-V

The trouble with helpful terms like "The 4 V's" is that it prevents other equally helpful characteristics from being considered unless you can force them somehow into a V-word. So, for example, there may be an argument that there should be a "U" in there—Usefulness. This can be measured as Potentially Useful, Immediately Useful, and so on. Some analytics experts would counter that all data is potentially useful! Data storage costs have plummeted, but they aren't zero yet. There are occasions when an organisation will decide it's just not worth hanging on to some sorts of data.

6.2.2 The Cloud Effect

Cloud computing brought with it flexible approaches to data storage. Before cloud, if you needed to capture 10 Terabytes of data, then filter it to remove unwanted data, ending up with one Terabyte of data, you actually needed to buy 10 Terabytes of storage. Once the filtering was done that would mean you had over-provisioned by 9 Terabytes. Now, with cloud, you can simply rent ten Terabytes from a service provider for a short period, carry out your filtering and then store only the one Terabyte you need, releasing the unwanted disk space.

If you add this flexibility to the relatively cheap cost of disk storage we can see why the propensity to save data in case it might be useful has risen and the drive to only store what is vital is reducing. If you then add the cloud's worldwide reach and the ease with which datasets can be gathered and analysed we can begin to see how Big Data is becoming so widely talked about.

When organisations begin to use Facebook and Twitter data for data mining purposes, they are, in effect, using the Cloud as part of their data storage strategy. Just as with more traditional database centred approaches, the data professional needs to ensure appropriate levels of availability to the data sources for the business users.

Guaranteed broadband speed connections to the internet are not always available, with availability rates depending upon geography as much as anything else. The most obvious decision is that if the data you need is critical you should store it where you have control and replicate it to ensure Disaster Recovery can happen.

However, it may be the case that it does not really matter to a business that its market intelligence gathered from Twitter does not have to be absolutely current. In that sort of case, bothering with the effort and expense to store the twitter data locally may not make sense, even in the era of cheap data storage. The decision might be to gather the data directly from the internet on an as needed basis, and just live with any gaps caused by lost connections.

In addition to externally sourced data, the Cloud now allows organisations to rethink their backup and DR strategies. As Hill et al. (2013) say:

> "Traditional backup and recovery methods have largely been based around magnetic tape as a medium for storing data. The advantage of this medium is that it is relative cheap. The disadvantage is that it is slow to access, and can be quite labour intensive. Systems where the user simply clicks a button to send their backup information to one of several servers somewhere in the Cloud are fast pressuring the traditional approach."

Because of the Pay-as-you-Go approach, Data Storage-as-a-Service has provided organisations with the opportunity to store their data off-site and by storing multiple copies of the data on several servers, these sites have built-in redundancy.

Database-as-a-Service is also becoming an alternative to maintaining your own database server. Microsoft's Azure is one example of this, but there are many others. In terms of Big Data, what this provides is instant scalability and flexibility. As we said above, should you need to store and analyse 1 Petabyte of data as a one-off exercise, you merely buy the space and pay for it only whilst you need it. Traditionally you would have had to ensure you had a free petabyte's worth of disk doing nothing—not a frequent occurrence in even rich organisations!

Cloud can be seen to be, in many ways, an enabler for Big Data. Arguably, we could even say that Big Data would not exist without it.

6.3 The Analytics View of Big Data

Something that comes out clearly from the material written about Big Data is that we are talking about making sense of the data we are storing, not just worrying about the physical aspects of storage.

Moreover, we are also examining data in new, innovative ways to discover new information. Some of this innovation in analysis is to do with the flexibility the new physical storage models allow. For the most part Business Analysts used to concentrate on asking questions of the data within their organisation's control; usually well structured in format and usually stored in a relational database.

Data warehousing as a discipline is not new. The core idea of capturing a company's OLTP data at certain points of time and storing them in a way that makes querying more efficient has been with us since the late 1980s. The heavy CPU usage caused by the analysis of the data by Business Analysts could dramatically slow down the database server and the core OLTP system would suffer as a result. Indexes can speed up the return of query results, but they always slow the Insert, Delete or Update process. The idea, then, was to extract the data, clean it and filter it, and then store it in a query friendly way, with appropriate schemas and indexes, and store it away from the "live" data.

This process is known by the acronym ETL; Extract, Transform and Load. As we look at the techniques used in Big Data we will see that the process is also a large part of the modern Analytics discipline and the Data Scientist needs to master it as part of their standard toolset.

Data warehousing is a large topic in its own right and outside of the scope of this chapter, but the technologies and techniques we cover here are already beginning to impact upon the area, and there will doubtless be a blurring of the edges between traditional warehousing and Big Data.

6.3.1 So Why Isn't Big Data Just Called Data Warehousing 2?

Well, in some ways it could be, but there are differences. Probably the most significant is to do with the analyst's ability to structure the data. As with most relational tasks, a warehouse needs a schema to describe it. To create that schema the designer has to know what data, and data-types, will be stored. They have to have some form of foreknowledge to be able to plan a warehouse effectively.

As we saw earlier, there are two "V" attributes of Big Data that make foreknowledge less likely: Variety and Velocity. Data is now coming into the organisation's purview from many different sources and presented in a variety of formats, such as CSV, Excel, XML, JSON. And that data is coming at the organisation quickly. When a new potentially useful data feed is found there is a need to instantly use it; not to have to define and model it, and design a schema to manage it.

The velocity and variety elements are going to be a big challenge. As global research and advisory firm Forrester employee Mike Gualtieri predicts in his blog (http://blogs.forrester.com/mike_gualtieri/13-01-02-big_data_predictions_for_2013):

> Real-time architectures will swing to prominence. Firms that find predictive models in big data must put them to use. Firms will seek out streaming, event processing, and in-memory data technologies to provide real-time analytics and run predictive models. Mobile is a key driver, because hyperconnected consumers and employees will require architectures that can quickly process incoming data from all digital channels to make business decisions and deliver engaging customer experiences in real time. The result: In 2013, enterprise architects will step out of their ivory towers to once again focus on technology—real-time technology that is highly available, scalable, and performant.

But the Volume aspect will also impact upon the analyst. Many data warehouses would archive out, or even delete, data that was beyond the standard reporting time periods—often this year and last. This would be to prevent data storage running out, and also to ensure good performance when querying the data. With these constraints, analysts tended to investigate trends within a relatively short time period—how many tins of beans did we sell last month as compared to the previous month or this time last year—but we are now able to store much more data for longer term analysis.

The ability to store and analyse huge volumes of historic data is likely to make new insights available to the savvy analyst. As always the hope when looking for new information in a commercial environment is that you can provide some, albeit temporary, competitive advantage. In less commercially focused organisations, such as those in the healthcare area, the hope is to be able to find trends that will help in the identification of causal relationships between lifestyle factors and disease.

As Samuel Arbesman put it in his wired article (http://www.wired.com/opinion/2013/01/forget-bigdata-think-long-data/):

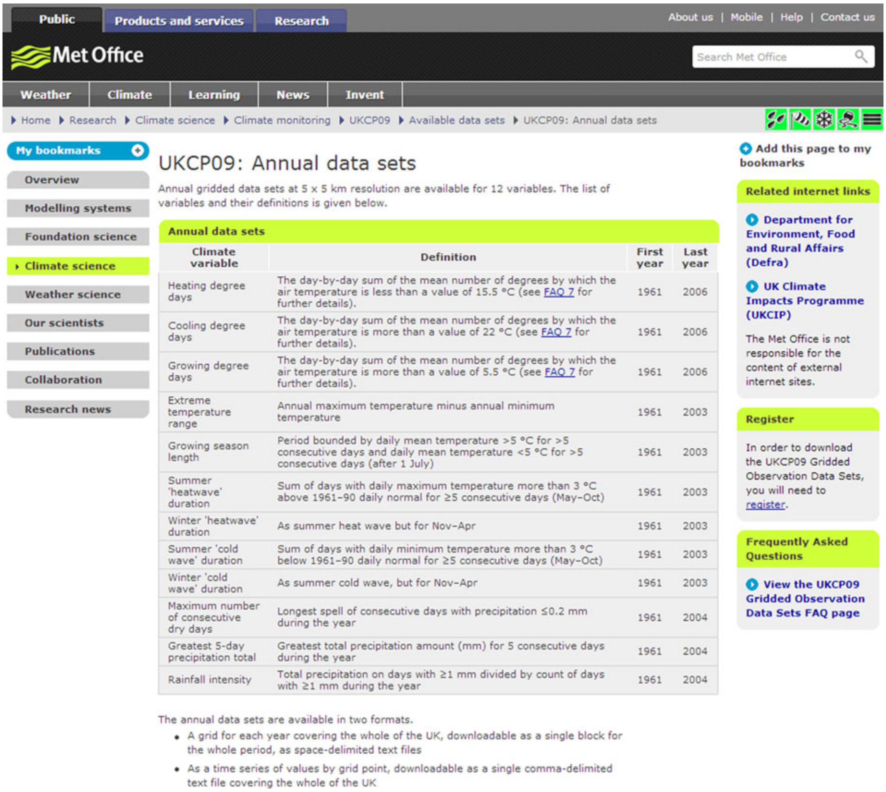
> Datasets of long timescales not only help us understand how the world is changing, but how we, as humans, are changing it—without this awareness, we fall victim to shifting baseline syndrome. This is the tendency to shift our "baseline," or what is considered "normal"—blinding us to shifts that occur across generations (since the generation we are born into is taken to be the norm).

Probably the most high profile area using long time-scale data at the moment is that of Climate Change, with historic data being used by proponents and doubters alike. And the recent trend in making public data more openly available is also a driver here. The UK Meteorological Office, for example has a selection of weather details back to 1961 available for use by anyone (see Fig. 6.2).

Data is also available from organisations like NOAA (National Oceanic and Atmospheric Administration), the US federal agency whose mission *is to understand and predict changes in climate, weather, oceans, and coasts*. They make public, for example, data from ice cores and tree rings which provide palaeoclimatology experts with the ability to deduce how the climate has changed through the earth's history.

So, lots of interesting, detailed new sources of data have recently become available and analysable. Exciting though this may seem for data scientists, it does raise another problem: just how do you select data which might produce useful results after analysis? For data scientists employed by a commercial organisation this question is likely to be framed as "can we contribute to the bottom line as a result of this research?" That constraint will tend to restrict the datasets investigated by any but the least risk-averse analyst as they try to make sure they keep their jobs. Nonetheless, the data scientists' job most certainly includes exploring for new datasets. It is just that it also must include a filtering process that ensures that the outcome is suitably focused.

This issue is well summarised by Vincent Granville on a blog (http://www.analyticbridge.com/profiles/blogs/the-curse-of-big-data):


Public
Products and services
Research
About us | Mobile | Help | Contact us
Met Office
Search Met Office
Weather
Climate
Learning
News
Invent
Home ▸ Research ▸ Climate science ▸ Climate monitoring ▸ UKCP09 ▸ Available data sets ▸ UKCP09: Annual data sets
My bookmarks
Overview
Modelling systems
Foundation science
Climate science
Weather science
Our scientists
Publications
Collaboration
Research news
Add this page to my bookmarks
Related internet links
Department for Environment, Food and Rural Affairs (Defra)
UK Climate Impacts Programme (UKCIP)
Register
Frequently Asked Questions
View the UKCP09 Gridded Observation Data Sets FAQ page

UKCP09: Annual data sets

Annual gridded data sets at 5 x 5 km resolution are available for 12 variables. The list of variables and their definitions is given below.

Annual data sets

Climate variable	Definition	First year	Last year
Heating degree days	The day-by-day sum of the mean number of degrees by which the air temperature is less than a value of 15.5 °C (see FAQ 7 for further details).	1961	2006
Cooling degree days	The day-by-day sum of the mean number of degrees by which the air temperature is more than a value of 22 °C (see FAQ 7 for further details).	1961	2006
Growing degree days	The day-by-day sum of the mean number of degrees by which the air temperature is more than a value of 5.5 °C (see FAQ 7 for further details).	1961	2006
Extreme temperature range	Annual maximum temperature minus annual minimum temperature	1961	2003
Growing season length	Period bounded by daily mean temperature >5 °C for >5 consecutive days and daily mean temperature <5 °C for >5 consecutive days (after 1 July)	1961	2003
Summer 'heatwave' duration	Sum of days with daily maximum temperature more than 3 °C above 1961–90 daily normal for ≥5 consecutive days (May–Oct)	1961	2003
Winter 'heatwave' duration	As summer heat wave but for Nov–Apr	1961	2003
Summer 'cold wave' duration	Sum of days with daily minimum temperature more than 3 °C below 1961–90 daily normal for ≥5 consecutive days (May–Oct)	1961	2003
Winter 'cold wave' duration	As summer cold wave, but for Nov–Apr	1961	2003
Maximum number of consecutive dry days	Longest spell of consecutive days with precipitation ≤0.2 mm during the year	1961	2004
Greatest 5-day precipitation total	Greatest total precipitation amount (mm) for 5 consecutive days during the year	1961	2004
Rainfall intensity	Total precipitation on days with ≥1 mm divided by count of days with ≥1 mm during the year	1961	2004

The annual data sets are available in two formats.

- A grid for each year covering the whole of the UK, downloadable as a single block for the whole period, as space-delimited text files
- As a time series of values by grid point, downloadable as a single comma-delimited text file covering the whole of the UK

The Met Office is not responsible for the content of external internet sites.

In order to download the UKCP09 Gridded Observation Data Sets, you will need to register.

Fig. 6.2 UK Met Office data

In short, the curse of big data is the fact that when you search for patterns in very, very large data sets with billions or trillions of data points and thousands of metrics, you are bound to identify coincidences that have no predictive power—even worse, the strongest patterns might be:

- entirely caused by chance (just like someone who wins at the lottery wins purely by chance) and
- not replicable,
- having no predictive power but obscuring weaker patterns that are ignored yet have a strong predictive power.

The question is: how do you discriminate between a real and an accidental signal in vast amounts of data?

Two elements of his own answer to the question posed are:

> Being a statistician helps, but you don't need to have advanced knowledge of stats. Being a computer scientist also helps to scale your algorithms and make them simple and efficient.

Which leads us nicely on to the next section:

6.3.2 What Is a Data Scientist?

There are beginning to be many jobs advertised for organisations looking for Data Scientists. The evidence is that this is a booming field and that the shortage of Data Scientists means that those that exist can earn very good salaries. Interestingly there are very few jobs advertised as looking for Big Data expertise, although the Data Scientist is clearly the most obvious user of Big Data techniques in most organisations.

The IBM Big Data website says that:

> Data scientists are inquisitive: exploring, asking questions, doing "what if" analysis, questioning existing assumptions and processes. Armed with data and analytical results, a top-tier data scientist will then communicate informed conclusions and recommendations across an organization's leadership structure.

Harvard Business Review published an article in October 2012 (http://hbr.org/2012/10/datascientist-the-sexiest-job-of-the-21st-century/) with the headline:

Data Scientist: The Sexiest Job of the 21st Century

However, as a word of warning before we all go running off to become a data scientists, they do go on to say that:

> There is also little consensus on where the role fits in an organization, how data scientists can add the most value, and how their performance should be measured.

So this is a new field and if you are early to the market you might well be able to earn significant rewards. In the UK, in one week (commencing 15th April 2013) there were newly listed 114 Data Science jobs on the single recruitment site www.jobsite.co.uk.

Many jobs for data scientists and data analysts are listed on the internet with salaries in the midrange professional band of around £40000 to £80000 ($60000–120000). The salary is usually dependent upon the expertise and experience of the individual, and the importance perceived by the employer of engaging high calibre employees. The story is the same in the USA. The CIA recently advertised for a data scientist with a salary range of $51,418–$136,771, for example.

Quoted in the Guardian's DataBlog (http://www.guardian.co.uk/news/datablog/2012/mar/02/datascientist) Monica Rogati, chief scientist at LinkedIn defines a data scientist in this way:

> In my opinion, they are half hacker, half analyst, they use data to build products and find insights. It's Columbus meets Columbo—starry eyed explorers and skeptical detectives.

This is an example of what is often quoted as the ideal mix—that of statistical and programming, or at least hacking, skills. However, the other part of the equation for a good data scientist is that they should understand the business they are working in. A hacking statistician with experience of working in the coal industry, for example, is more likely to be able to ask appropriate questions for a coal mining corporation, that than a hacking statistician who has experience in the clothes retailing industry. They will know what to look for amongst the patterns they find in the data.

Global consulting company McKinsey recently issued a Big Data report that predicted that:

> There will be a shortage of talent necessary for organizations to take advantage of big data. By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions.

http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation.

Similarly, in Great Britain, e-Skills UK published a report (e-Skills UK 2013) called *An assessment of demand for labour and skills, 2012–2017* which identified:

> … that there were approximately 3,790 advertised positions for big data staff in the UK in the third quarter of 2012, 75 % of which were for permanent posts.

The report goes on to say:

Despite the currently unfavourable economic climate, demand for big data staff has risen exponentially (912 %) over the past five years from less than 400 vacancies in the third quarter of 2007 to almost 4,000 in the third quarter of 2012.

6.3.3 What Is Data Analysis for Big Data?

Probably the most important place to start is with questions, not data. Being data driven can occasionally help find unexpected information, but given the need to prioritise a data scientist's tasks, as mentioned above, some sort of identification of what an organisation needs to know is essential. This is another reason why some experience in the industry concerned can be a big advantage.

Different industry sectors will tend to want to ask different types of questions. Food retailers, for example, are often looking for buying patterns, both at the individual, and at the market level. Put very simply, if you can use data to evidence the fact that every summer month we sell twice as much ice cream as we do in other months, we know that we should stock more to meet the expected demand.

Many retail store managers would tell you that they make this sort of decision intuitively anyway. And with years of experience in the job they may well be able to "sense" trends rather than precisely discover them. However, they can not hope to be able to identify trends for all products, especially when the trends are less obvious than "hot weather equals more ice cream".

Production orientated organisations, such as in the coal mining industry, may want to ask similar demand focused questions to help them regulate their supply to the market. However, they may also want to ask questions about the production itself. Historically, for example, can they say that the kilogrammes of coal dug per man-shift is lower on any particular day of the week? Some senior mining engineers might use their gut-feel and say that Friday is always the least productive day. But does the data support that conjecture?

In general, then, big data analytics is about exploring large volumes of data looking for trends, anomalies, previously unknown correlations and obscure patterns which can be used to an organisation's advantage by evidencing some sort of prediction upon which better business decisions are taken.

6.4 Big Data Tools

If you were to doubt that Big Data is really a part of the corporate lexicon, just have a look at some of the vendors who use the phrase prominently:

- Oracle
- SAS
- SAP

- IBM
- EMC

These are all leaders in the field of delivering enterprise scale solutions. All have significant amounts of Web collateral which explain, propose, and sell Big Data products. It is not likely that they have all called this wrong. And they are by no means the only players in the field.

The same e-Skills UK (2013) report referred to above identified the tools and skills that employers are looking for:

> The technical skills most commonly required for big data positions as a whole were: NoSQL, Oracle, Java and SQL, whilst the technical process/methodological requirements most often cited by recruiters were in relation to: Agile Software Development, Test Driven Development (TDD), Extract, Transform and Load (ETL) and Cascading Style Sheets (CSS).

Some of these tools, whilst highly relevant to Big Data, are not Big Data specific and have been around for a while, but it is worth exploring a couple of the newer tools. As with all new areas there are different understandings of what terms mean. Hadoop, for example, is most certainly a frequently asked for skill, but does not appear in the above list as it is presumably classified as NoSQL. In this book the preceding chapter was all about NoSQL databases, but we have decided that Hadoop is so Analytics focused that we would review it here, rather than in Chap. 5.

6.4.1 MapReduce

MapReduce has become a core technique for data scientists. Recognising that analysing high volumes of potentially unstructured data that may be spread across many data nodes is not something that the relational model would be good at handling, people began to turn to a two stage process. At its simplest this entails:

- looking through all the data and extracting a key identifier and a value, and then storing that in a list.
- Then using grouping, reducing the data thus listed such that you can get information like group totals, averages, minimums and the like.

This process can be shown graphically by referring to Fig. 6.3.

Although the concept is not new, it is as a result of the Google innovators that we are currently seeing such heavy reliance on the technique to cope with large volumes of data. Dean and Ghemawat (2008) tell us that in order to handle the complex and large data analysis tasks they were daily encountering at Google:

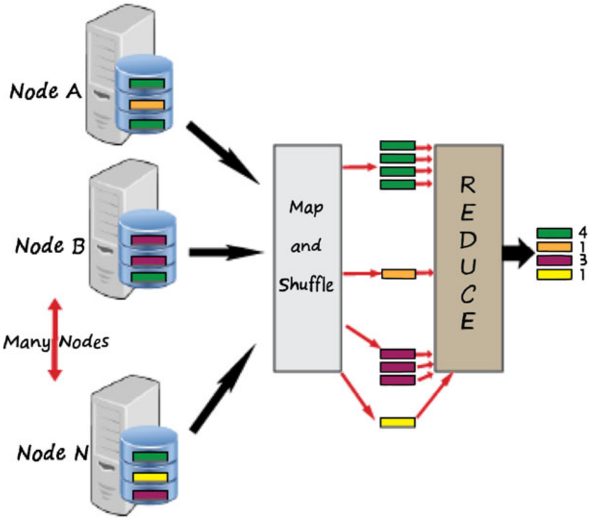


Fig. 6.3 The MapReduce process

> … we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. Our abstraction is inspired by the map and reduce primitives present in Lisp and many other functional languages.

NoSQL tools, such as MongoDB have this open source method built into their databases. For examples of using Map Reduce to analyse data, see the MongoDB example in the tutorial section later in this chapter.

6.4.2 Hadoop

Just as MapReduce has become a big feature in the Big Data arena, one of the most frequently used implementations of MapReduce is Hadoop. This is part of the Apache open source suite, and so is freely downloadable from http://hadoop.apache.org/. The website describes it as a:

> … a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.

The two key elements of Hadoop are the distributed file management system (HDFS) and Hadoop MapReduce. The MapReduce examples we provide later are similar except that we use MongoDB to manage the raw data, as opposed to HDFS.

HDFS is distributed and expects to run over several, or many, nodes, although it is possible to run it on a single PC and simulate multi-node operation. It uses replication of data across nodes for both availability and performance reasons. Data nodes are managed by a master node called the NameNode which keeps track of all the files and rebalance data by moving copies around. Client applications, such as MapReduce, talk to the NameNode to locate data. This makes the NameNode a single point of failure and is the potential weakness in terms of High Availability (see Chap. 10 for information on HA). MapReduce is a separate layer which uses HDFS to source its data requests.

6.4.2.1 If You Should Want to Explore Hadoop…

The examples later in this chapter, using MongoDB, show that MapReduce can be a powerful tool. It is also at the core of Hadoop. At the time of writing there are several example Hadoop installations available for downloading as Virtual Machines, meaning you do not have to go through the pain of installation. That said, if you have access to an Ubuntu (or other Linux) machine, then undertaking the installation journey for yourself will help you get a good understanding of the underpinning architecture.

Amazon's Elastic MapReduce A paid for service, but for the relatively low levels of processing we are talking about for these tutorials, this is not expensive and is a relatively hassle free way of getting started. As their website says:

> Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data. It utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3).

http://aws.amazon.com/elasticmapreduce/.

Cloudera Cloudera sample downloadable VM image (https://ccp.cloudera.com/display/SUPPORT/Cloudera's+Hadoop+Demo+VM+for+CDH4), also have a downloadable VirtualBox Image.

Hortonworks Their sandboxed version allows you to download for VM or VirtualBox. For these tutorials the author used the VirtualBox version 4.2.10 running on a Windows 7 host.

6.4.3 Hive, Pig and Other Tools

With computing tools there is often a trade-off between powerfulness and ease of use. This is certainly true of Hadoop which is a very powerful and flexible tool, but for which even its biggest fans do not claim ease of use as one of its attributes. In order to get the best from Hadoop the user should ideally have Java skills and be fully conversant with the Linux operating environment and distributed systems.

As Hadoop is open source there has been a sudden upsurge in Hadoop add-on software to attempt to make the data scientist's job easier. Some, like JasperSoft (http://www.jaspersoft.com/) provide an entire Business Intelligence stack that can plumb into Hadoop (and other data sources). Other vendors, like Oracle, provide their own tools but allow the use of "connectors" to use Hadoop when needed. Two popular open source Hadoop running mates are Hive and Pig.

Hive is, in effect, a data warehousing environment that uses Hadoop behind the scenes. It aims to make data summaries and *ad hoc* queries against big data sets simpler than it would be using raw Hadoop. One of the ways it does this is with a SQL-like query language called QL. http://hive.apache.org/.

Pig uses a programming language called Pig Latin which, when compiled, produces sequences of Map-Reduce programs using Hadoop. It provides a variety of functions and shell and utility commands. Originally developed by Yahoo to provide an ad-hoc way of creating and executing map-reduce jobs, it is now open source and freely usable. http://pig.apache.org/.

6.5 Getting Hands-on with MapReduce

The tutorial material in this chapter demonstrates the use of MapReduce with MongoDB on Linux, and assumes you have carried out the data inserts in the MongoDB tutorial in Chap. 5.

MapReduce as a means of analysing large amounts of data really came to the fore as Google used it to analyse their BigTable data. In the open source world many organisations have turned to Hadoop to perform the same sort of functionality.

Before we move on to those examples, however, you do need to be aware that if you are looking to gain employment as a Data Scientist you will probably have to bite the bullet and become proficient with Hadoop, as it has become the *de facto* standard in the Big Data area. Even big and powerful vendors like Oracle, SAS and SAP have found no better solution than Hadoop and build it in to their own products. Some examples of how you might get to grips with Hadoop are given at the end of the chapter.

Hadoop would doubtless take a few chapters in a specialist Data Science textbook, and so we feel that its complications, and the need for reasonable Java programming skills, make it unsuitable for this text. Hadoop installation is complex and there are many potential pitfalls in the process, as the author learned to his cost! As we are more interested in using MapReduce itself, and we already have some MongoDB experience (Chap. 5) the easiest solution is for us to use the MapReduce functionality built in to MongoDB to demonstrate its use.

6.6 Using MongoDB's db.collection.mapReduce() Method

As we saw in Chap. 5, MongoDB provides a number of built-in methods in the shell. We used the db.collection.find() method, for example, to search for records.

Another shell method provided is the db.collection.mapReduce() method which is a wrapper around the mapReduce command. We need to write a little bit of code to make this work for us in our environment—in effect creating our own extension to the Map-Reduce process.

Start by re-opening the MongoDB database. Assuming you inserted the Airport data in Chap. 5 you can jump straight to the Airports Database and we will be working with the AllAirports collection. If you did not do this you need to go to the end of the MongoDB tutorial in Chap. 5. The screen dump below reminds us how to do this before starting to build the functions required.

You would probably also need to remind yourself of what the data looks like (again, see Chap. 5). Our first Map Reduce exercise will count the number of Airports each country has.

Remembering that MapReduce is actually a two stage process we need to create first the **map** function and then the **reduce** function. In this case the AirportCount_map function adds the number 1 to the output against each instance of every Country Code. The reduce function will use the 1 for its count. The "emit" line is where we define what to output. The reduce function will then add up all the 1s for each Country Code. Reduce takes the output from Map and creates an Array. The word "this" is used to refer to "the collection currently being operated on". When we call this function you will see that the collection is part of the call.

We have now created the functions (Fig. 6.4) required and the next step is to actually call the functions, using the mapReduce method that MongoDB provides. We have to tell the method which collection we are using, and this is what our map function will use as "this".

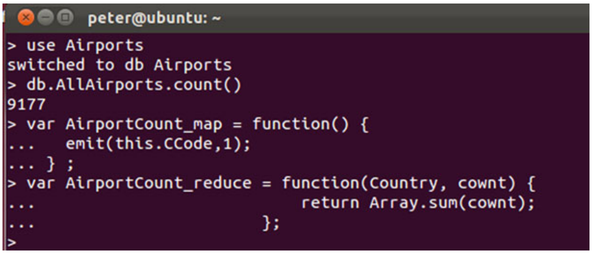
The first two parameters to pass are the names of the **map** and **reduce** functions that we have just created, and then a third parameter tells MongoDB which collection to write the results to (Fig. 6.5).

As you can see there is some reassurance returned when the method runs successfully. The final step is to look at the results, which we have asked MongoDB to store in the map_reduce_output collection. We need to be aware that if this collection already existed, it would get over-written by this. The opening section of the content of the collection is shown in Fig. 6.6.

So, given that we can discover that AU is the code for Australia we can see that they have 610 airports, whereas Andorra (AD) has just one.

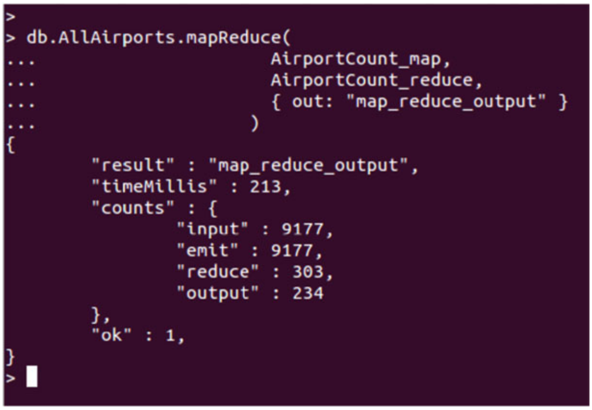
As can be seen from this example the limit of what you can do with map reduce is more to do with the user's programming ability than with the function itself! Just to push this a little further, now let's see if we can use map reduce to tell us how many airfields are on the same parallel.

Firstly, we need to recognise that lines of parallel refer to latitude. Moreover, the data we have is too exact and we will need to use only the units from the latitude stored. So now all we need to do is alter the **map** method so that it outputs a parallel and a 1. The reduce and output can remain the same.



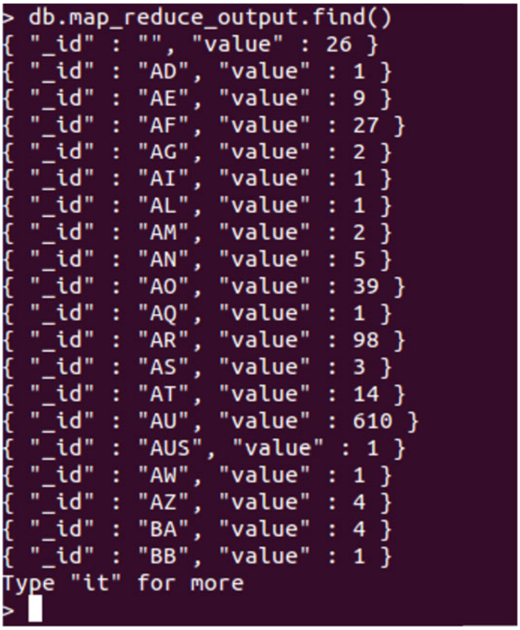
```
> use Airports
switched to db Airports
> db.AllAirports.count()
9177
> var AirportCount_map = function() {
...    emit(this.CCode,1);
... } ;
> var AirportCount_reduce = function(Country, cownt) {
...                           return Array.sum(cownt);
...                       };
>
```

Fig. 6.4 Creating the MapReduce functions



```
>
> db.AllAirports.mapReduce(
...                           AirportCount_map,
...                           AirportCount_reduce,
...                           { out: "map_reduce_output" }
...                         )
{
        "result" : "map_reduce_output",
        "timeMillis" : 213,
        "counts" : {
                "input" : 9177,
                "emit" : 9177,
                "reduce" : 303,
                "output" : 234
        },
        "ok" : 1,
}
> 
```

Fig. 6.5 MapReduce being called



```
> db.map_reduce_output.find()
{ "_id" : "", "value" : 26 }
{ "_id" : "AD", "value" : 1 }
{ "_id" : "AE", "value" : 9 }
{ "_id" : "AF", "value" : 27 }
{ "_id" : "AG", "value" : 2 }
{ "_id" : "AI", "value" : 1 }
{ "_id" : "AL", "value" : 1 }
{ "_id" : "AM", "value" : 2 }
{ "_id" : "AN", "value" : 5 }
{ "_id" : "AO", "value" : 39 }
{ "_id" : "AQ", "value" : 1 }
{ "_id" : "AR", "value" : 98 }
{ "_id" : "AS", "value" : 3 }
{ "_id" : "AT", "value" : 14 }
{ "_id" : "AU", "value" : 610 }
{ "_id" : "AUS", "value" : 1 }
{ "_id" : "AW", "value" : 1 }
{ "_id" : "AZ", "value" : 4 }
{ "_id" : "BA", "value" : 4 }
{ "_id" : "BB", "value" : 1 }
Type "it" for more
> 
```

Fig. 6.6 MapReduce output

Try changing the **map** code to read:

```
var LatCount_map = function() {
var key = Math.floor(this.Lat);
emit(key, 1);
};
```

But this isn't quite accurate enough since the **floor** function works the wrong way for negative numbers—−5.5 would become −6 if we used floor. So we need to use **ceil** with negatives. Here is the revised code, using the appropriate rounding tool depending upon the Latitude passed:

```
var LatCount_map = function() {
if (this.Lat > 0)
{
var key = Math.floor(this.Lat);
}
else
{
var key = Math.ceil(this.Lat);
}
emit(key, 1);
};
```

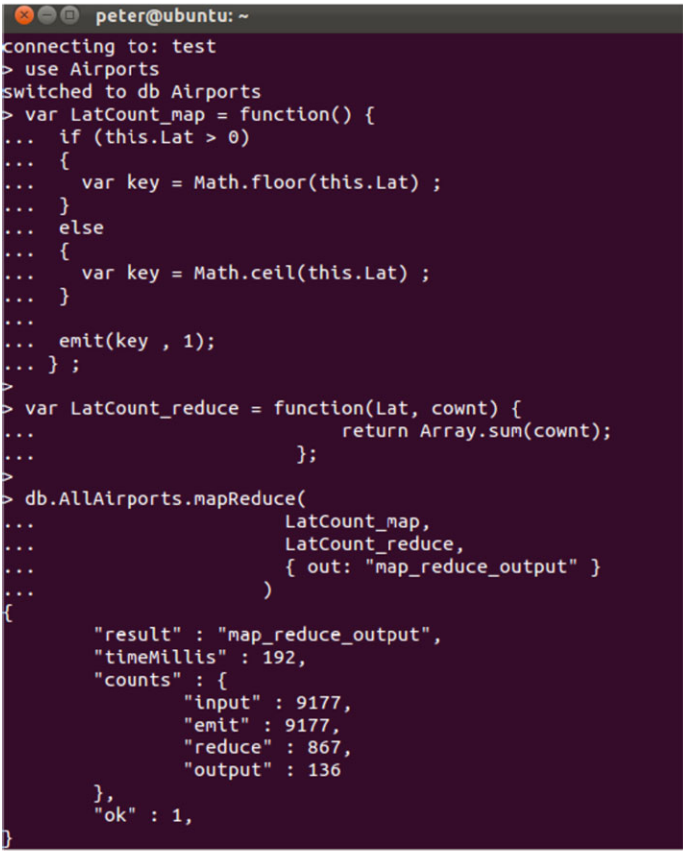
A screenshot of this being called is shown in Fig. 6.7.

Exercise 1

Now see if you can produce a map reduce output that tells us how many airports there are in each of the earth's hemispheres. A worked example is at the end of this chapter. **HINT:** the key in the last example was a number, but it could have been text.

We can also alter the functionality of the reduce function. Let us say we want to discover the furthest north there was an airport. The **map** function could just send out the list of latitudes and then the **reduce** could discover the maximum value:

```
var MostNorth_map = function() {
    emit('Lat', this.Lat);
};
```



```
connecting to: test
> use Airports
switched to db Airports
> var LatCount_map = function() {
...   if (this.Lat > 0)
...   {
...     var key = Math.floor(this.Lat) ;
...   }
...   else
...   {
...     var key = Math.ceil(this.Lat) ;
...   }
...
...   emit(key , 1);
... } ;
>
> var LatCount_reduce = function(Lat, cownt) {
...                               return Array.sum(cownt);
...                           };
>
> db.AllAirports.mapReduce(
...                          LatCount_map,
...                          LatCount_reduce,
...                          { out: "map_reduce_output" }
...                        )
{
        "result" : "map_reduce_output",
        "timeMillis" : 192,
        "counts" : {
                "input" : 9177,
                "emit" : 9177,
                "reduce" : 867,
                "output" : 136
        },
        "ok" : 1,
}
```

Fig. 6.7 Airfields on the same Parallel

```
var MostNorth_reduce = function(key, values) {
   var max = values[0];
   values.forEach(function(val){
      if (val > max) max = val;
   })
   return max;
}

db.AllAirports.mapReduce(
     MostNorth_map,
     MostNorth_reduce,
     { out: "map_reduce_output" }
     )
db.map_reduce_output.find()
```

Exercise 2

Now see if you can produce a map reduce output that tells us what is the most westerly airport from the Prime Meridian (0°). A worked example is at the end of this chapter.

6.6.1 And If You Have Time to Test Your MongoDB and JS Skills

Only try this if you have some Javascript knowledge! To help, you need to be aware that what is returned when you use the collection find() method is an array, and the output can be used programmatically by declaring a variable array to capture the output values.

Exercise 3

Use the output from Exercise 2 to print the all airport details we hold for the most westerly airport. A worked example is at the end of this chapter.

6.6.2 Sample Solutions

Ex 1 Hemispheres:

```
var HemiCount_map = function() {
if (this.Lat > 0)
{
    var key = “North”;
}
else
{
    var key = “South”;
}
emit(key, 1);
};
var HemiCount_reduce = function(Hemi, cownt) {
         return Array.sum(cownt);
      };
```

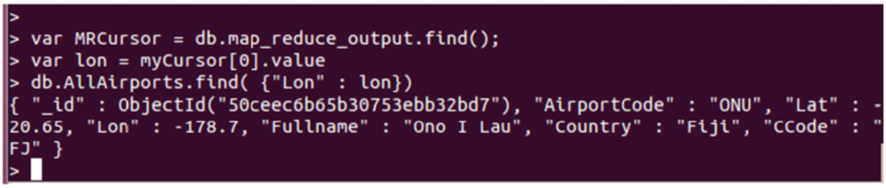
```
db.AllAirports.mapReduce(
        HemiCount_map,
        HemiCount_reduce,
        { out: "map_reduce_output" }
     )
db.map_reduce_output.find()
```

Ex 2 Most Westerly:

```
var MostWest_map = function() {
      emit('Lon', this.Lon);
};
var MostWest_reduce = function(key, values) {
   var min = values[0];
   values.forEach(function(val){
      if (val < min) min = val;
   })
   return min;
}
db.AllAirports.mapReduce(
        MostWest_map,
        MostWest_reduce,
        { out: "map_reduce_output" }
     )
db.map_reduce_output.find()
```

Ex 3 Most Westerly details:

```
var MRCursor = db.map_reduce_output.find();
var lon = myCursor[0].value
db.AllAirports.find( {"Lon": lon})
```



```
>
> var MRCursor = db.map_reduce_output.find();
> var lon = myCursor[0].value
> db.AllAirports.find( {"Lon" : lon})
{ "_id" : ObjectId("50ceec6b65b30753ebb32bd7"), "AirportCode" : "ONU", "Lat" : -
20.65, "Lon" : -178.7, "Fullname" : "Ono I Lau", "Country" : "Fiji", "CCode" : "
FJ" }
> 
```

6.7 Summary

In this chapter we have seen that Big Data is not just about the number of bytes of data we are storing, but about the complexity of the data, and the speed with which it arrives. Organisations need to be able to make sense of more and more data, and begin to look outside of their own data sources. Tools that have enabled the change in the way we store and analyse data are also increasing in number and maturity rapidly. At the time of writing Hadoop seems to have become the *de facto* standard such tool with many leading commercial vendors adopting it and integrating with it. MapReduce is a key part of Hadoop, but is a general programming approach and other tools allow MapReduce, as we saw with MongoDB in the tutorial.

6.8 Review Questions

The answers to these questions can be found in the text of this chapter.

- What has become the *de facto* standard approach for handling large datasets stored across many nodes? Your answer could be a framework or a programming model—or both!
- What do the letters SPA stand for, as used by Forrester talking about big data?
- What are the 4 “V”s?
- What makes Big Data different from traditional data warehousing?
- What is HDFS?

6.9 Group Work Research Activities

*These activities require you to research beyond the contents of the book and can be tackle*d *individually or as a discussion group.*

Discussion Topic 1 As the CIO of a company you need to come to terms with Big Data. Your board of directors are asking you what it might mean for them. Review the key elements of Big Data, and how they may impact upon any organisation’s information strategy, and attempt to report, in terms simple enough for a non-technical executive, what changes to information management might be worth exploring as a consequence of the new technologies becoming available.

Discussion Topic 2 “*Big Data is nothing new*”. Discuss this assertion. You should review the benefits and disadvantages of different approaches to storage and analysis of large volumes of data.

References

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113. 2008. doi:10.1145/1327452.1327492,

e-Skills UK (Jan 2013) Big data analytics: an assessment of demand for labour and skills, 2012–2017. http://www.e-skills.com/research/research-publications/big-data-analytics/

Hill R, Hirsch L, Lake P, Moshiri S (2013) Guide to Cloud Computing: principles and practice. Springer, London