What is Hadoop?

The webinar will begin at 3pm

- You now have a menu in the top right corner of your screen.
- The red button with a white arrow allows you to expand and contract the webinar menu, in which you can write questions/comments.
- We won’t have time to answer questions while we are presenting, but will answer them at the end
- You will be on mute throughout – we can’t hear you.

What is Hadoop?

Webinar

23 February 2016

Peter Smyth
UK Data Service

UK Data Service

Can you hear us?

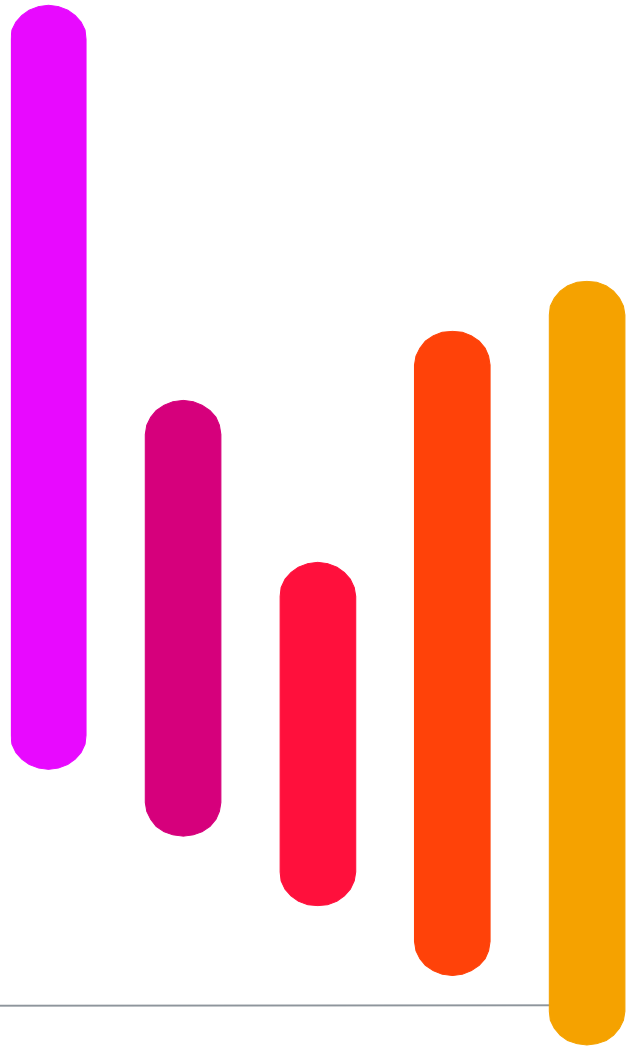
UK Data Service

Can you hear us?

- If Not:
  - Check your volume, and that your speaker/headset is plugged in.
  - Your invitation also included a phone number, you can call that to listen in.
    - UK +44 (0) 20 3713 5012
    - US +1 (646) 307-1716
  - We are recording this webinar, so you can always listen to it later.

Overview of this webinar

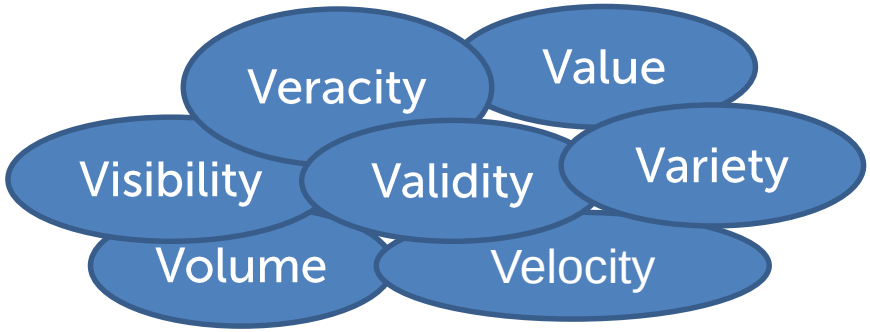
- welcome
- what is big data
- why big data
- processing big data with Hadoop
- examples using Hive

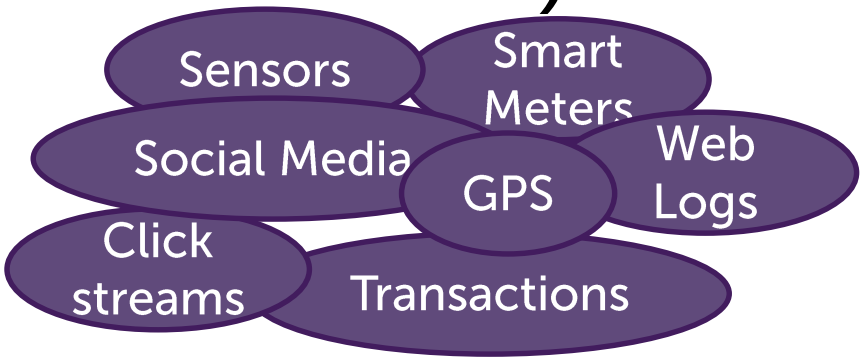


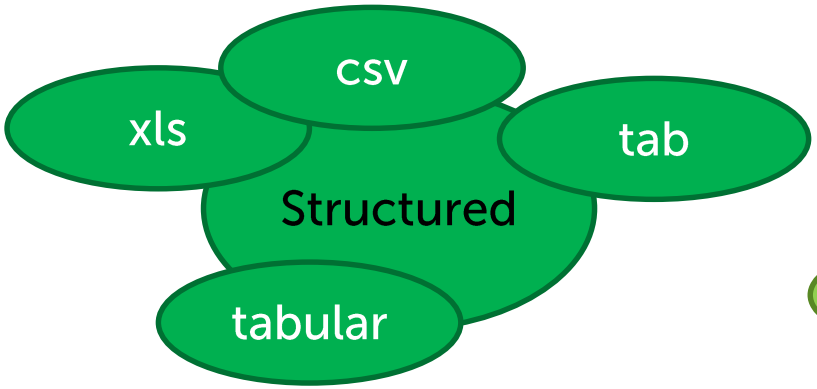
What is big data?

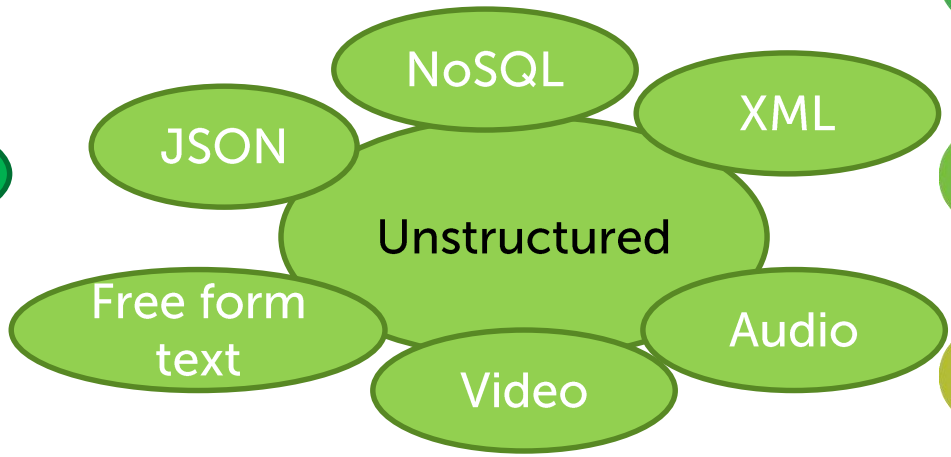
- There are plenty of definitions available on the internet involving words beginning with V.

Defining Big Data (Pick Your Own)









Big data solutions?

What is big data?

- Perhaps most relevant, is the most obvious;

  “Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate.” Unstructured Data

  (wikipedia)

- Or to put it another way, too big to fit into your favourite desktop application.

Why big data?

- Data comes from a variety of new sources
- The data has not been generated specifically for analysis
- Unstructured data is typically verbose
- Or, data simply contains more than we actually need for our analysis

Structured data

- Easiest to think of as tabular data
- That is, you can create the table structure in advance (the column names and the types of data each will contain) and then when you get the data, you populate the table with it.
- A disadvantage of this is that it is somewhat inflexible in dealing with changes to the structure
- An advantage, is that you can perform a lot of validation checks as you load the data.

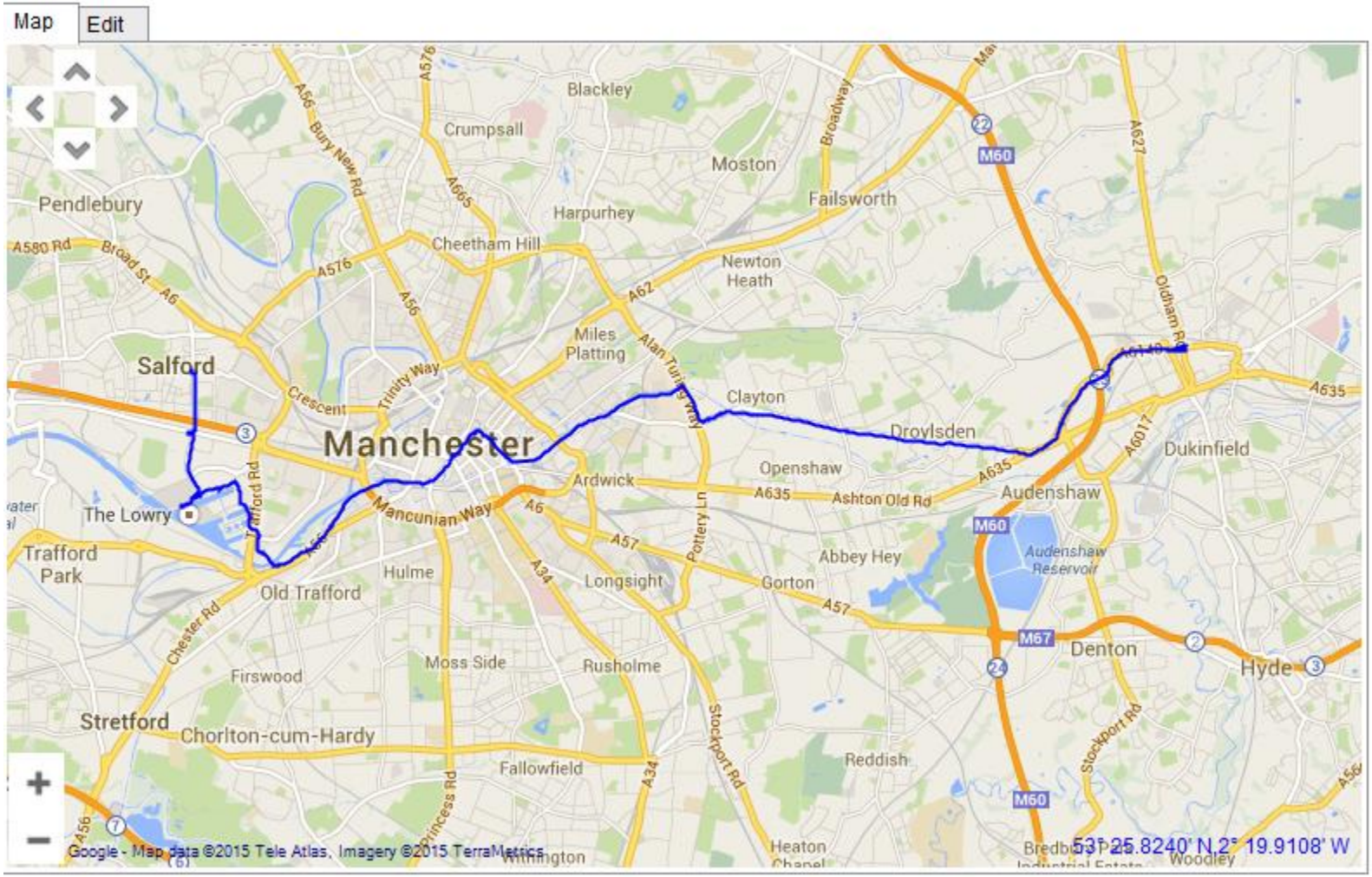
Unstructured Data

- This is ‘free-format’ data
- There is no guarantee of what data items will be included nor of the order in which they will appear
- Despite the name however, there is still some structure
- When data is originally received, it is stored ‘as-is’
- When the data is subsequently processed, the values from specific fields are requested

Examples of structured and unstructured data formats

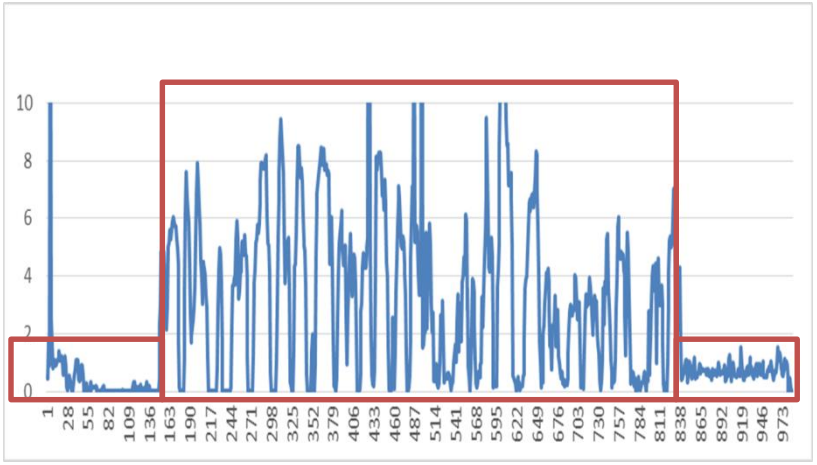
	Format	Example
Structured	XLSX	**qn** **pc** **hn** **qa** 1 M60 3SF 1 0
Structured	CSV	qn,pc,hn,qa 1,M60 3SF,1,0
Structured	Tab-delimited	qn pc hn qa 1 M60 3SF 1 0
Unstructured	XML	<Header> <Column>qn</Column> <Column>pc</Column> <Column>hn</Column> <Column>qa</Column> </Header> <Row> <Cell>1</Cell> <Cell>M60 3SF</Cell> <Cell>1</Cell> <Cell>0</Cell> </Row>
Unstructured	JSON	{ "qn":1, "pc":"M60 3SF", "hn":1, "qa":0 }

Map of Trip


Map
Edit
Blackley
Crumpsall
Moston
Failsworth
Pendlebury
Harpurhey
Cheetham Hill
Newton Heath
Miles Platting
Salford
Clayton
Manchester
Droylsden
Dukinfield
Openshaw
Ardwick
Audenshaw
The Lowry
Trafford Park
Old Trafford
Hulme
Abbey Hey
Longsight
Gorton
Audenshaw Reservoir
Denton
Hyde
Moss Side
Rusholme
Firswood
Stretford
Chorlton-cum-Hardy
Fallowfield
Reddish
Heaton Chapel
Woodley
M60
M67
A57
A635
A6
A34
A62
A576
A56
A665
Mancunian Way
Ashton Old Rd
Broadway
Oldham Rd
Stockport Rd
Pottery Ln
Trafford Rd
Chester Rd
Princess Rd
Trinity Way
Crescent
Alan Turing Way
Google - Map data ©2015 Tele Atlas, Imagery ©2015 TerraMetrics
53° 25.8240' N 2° 19.9108' W

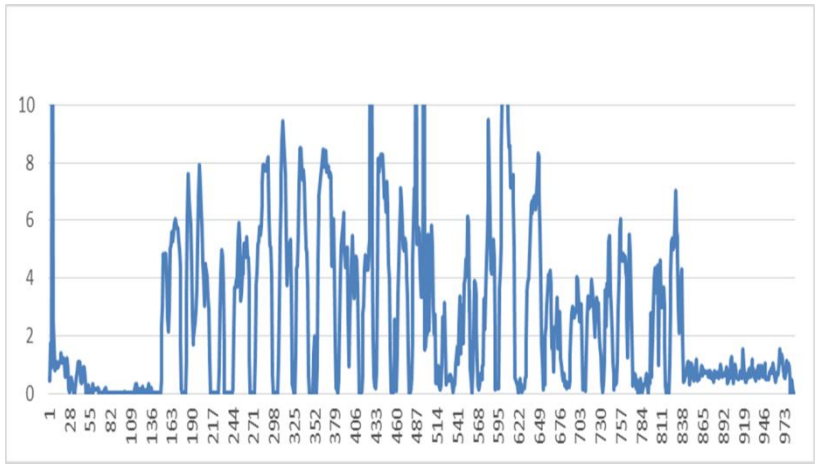
Graphs (all data points)

Speed vs time

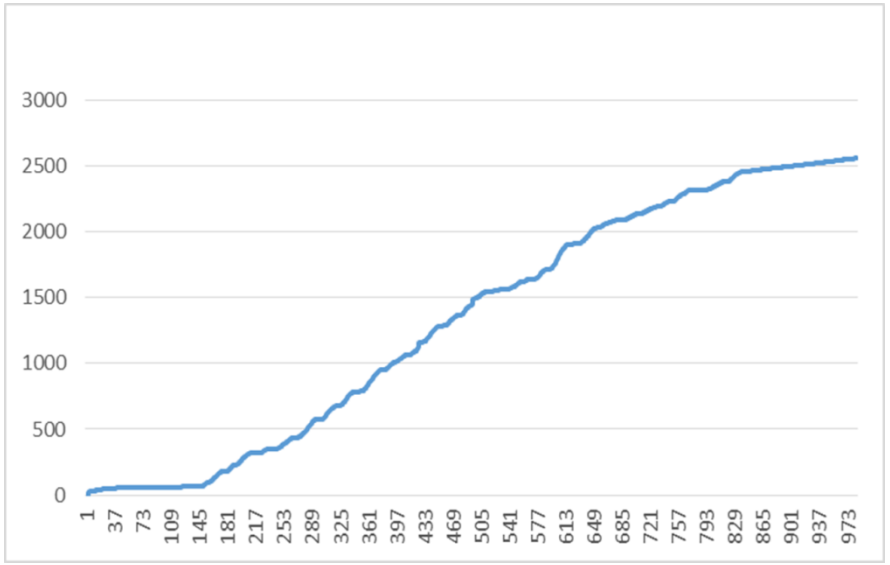


Graphs (all data points)

Speed vs time

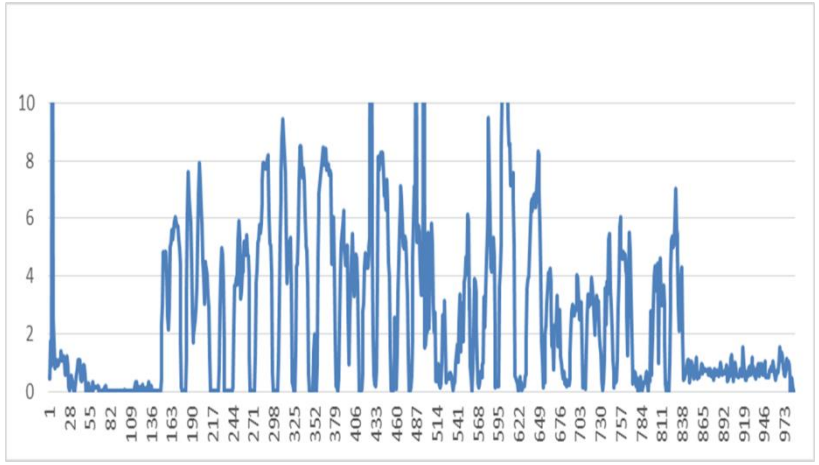


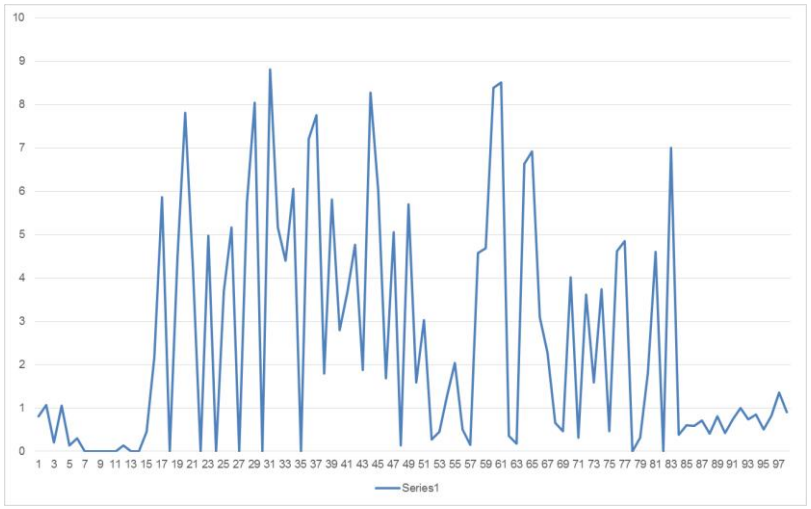
Distance vs time



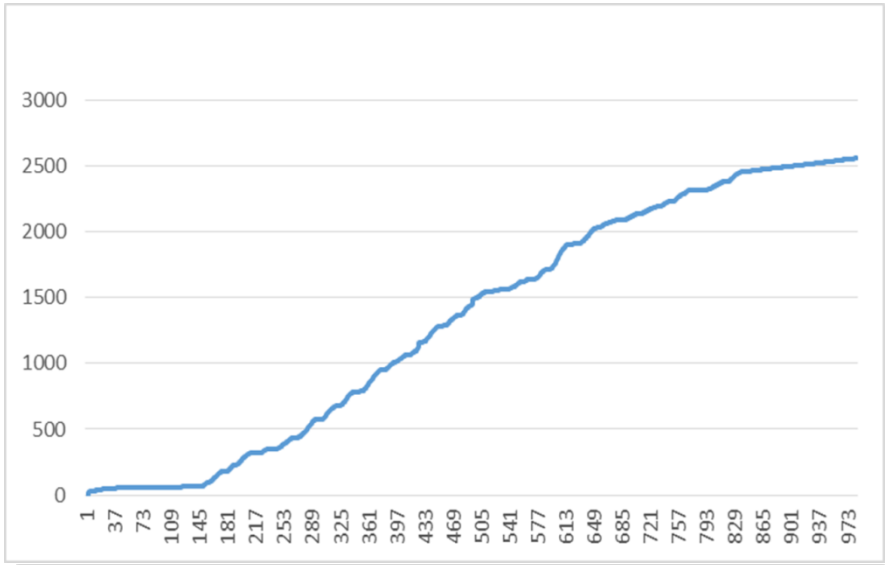
Graphs (all data points v 1/10 data points)

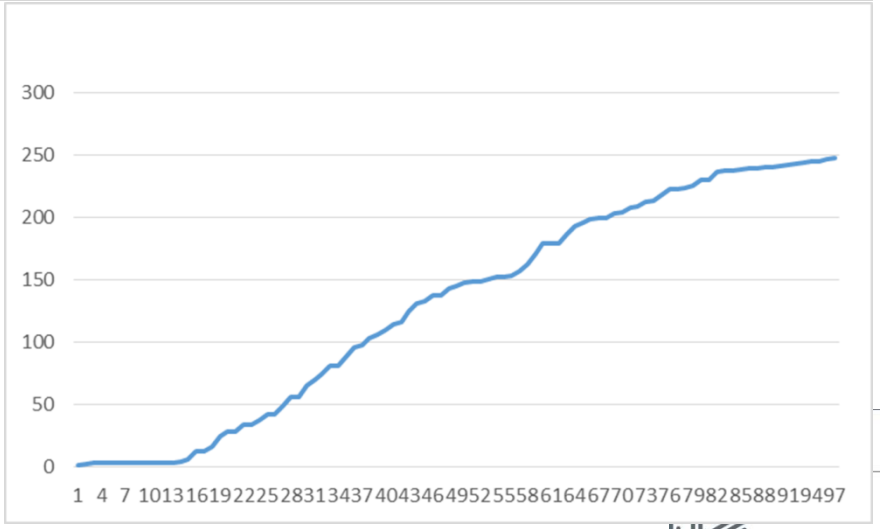
Speed vs time



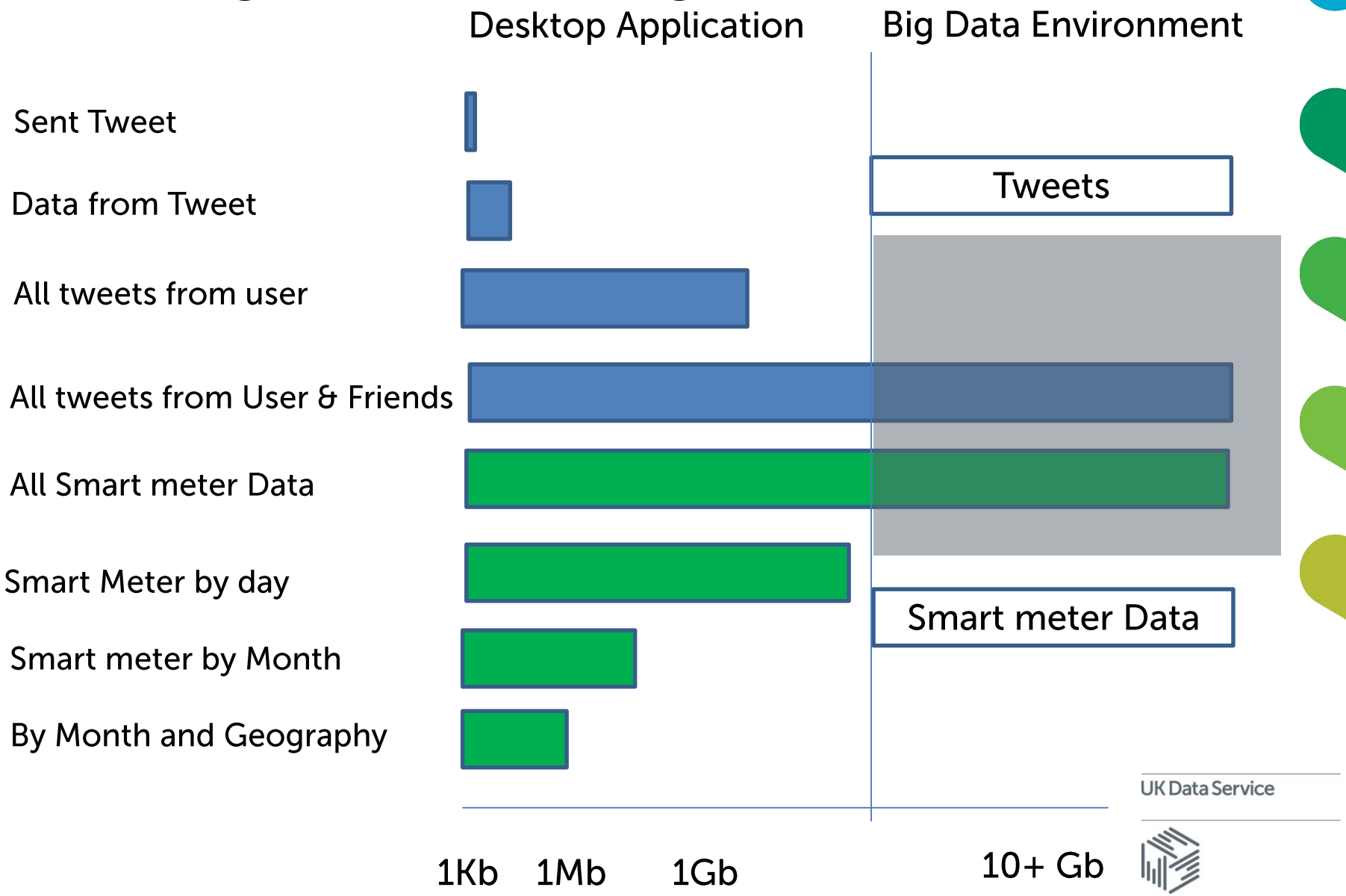


Distance vs time



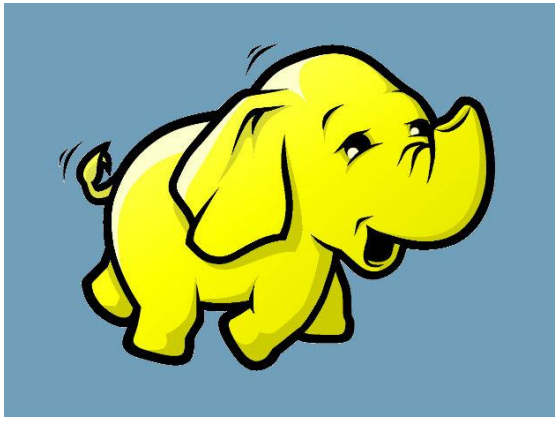


Growing and shrinking data



Hadoop

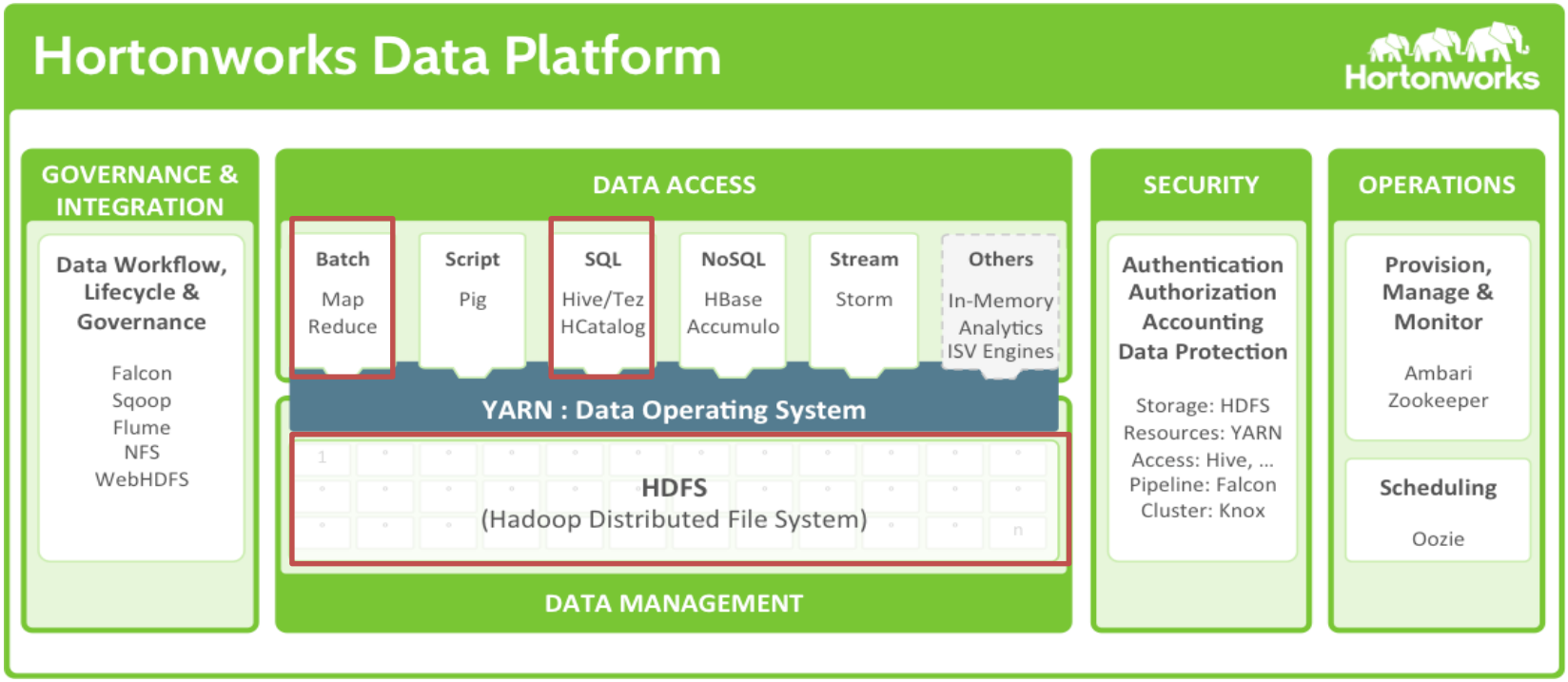
- Created by Doug Cutting and Mike Cafarella
- Based on the 2003-4 Google papers on MapReduce and GFS (Google File System)
- MapReduce had been around for 40 years, first appearing in the programming language Lisp in 1961
- The name Hadoop comes from the name of a cuddly toy elephant owned by Doug Cutting's son



Hadoop Infrastructure

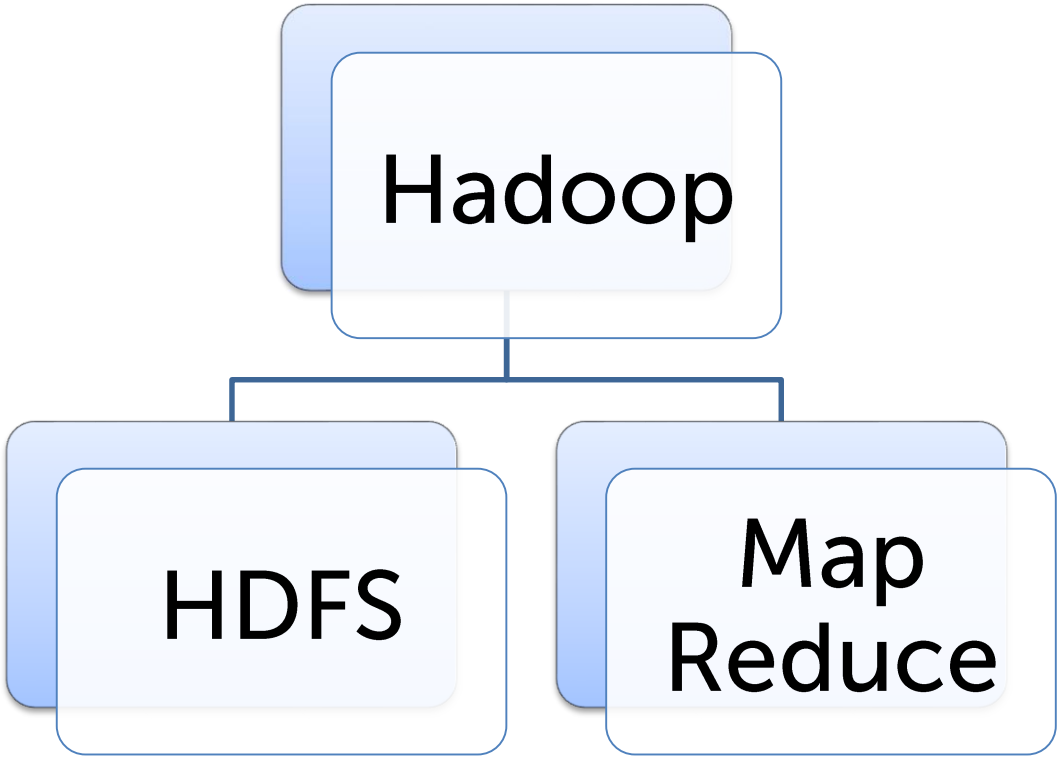
- A Hadoop cluster can be formed by thousands of nodes – 4 would be a minimum
- A node is an individual computer many times more powerful than the average desktop.
- The strength of the Hadoop system is not its raw power, but its ability to break a processing task down and run all of the parts in parallel
- It is built to be resilient, i.e. cope with server failures

A picture of the Hadoop Eco-system



(http://hortonworks.com/)

A (very) minimal picture of Hadoop



Hadoop components

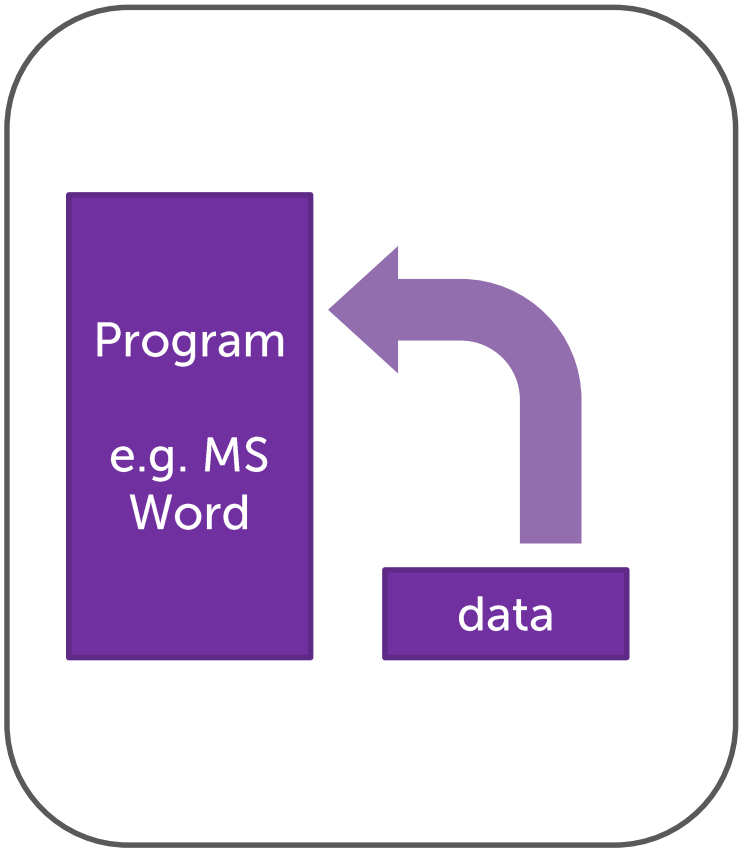
- HDFS (Hadoop Distributed File System)
  - To the end user, just a file system
  - Internally, files are segmented into blocks of 128MB and randomly distributed across the available datanodes
  - A datanode is a server in the Hadoop cluster where actual processing takes place – i.e. where your programs are run.
  - A namenode (another server in the cluster) keeps track of the random distribution of the blocks

Ordinary Client Server

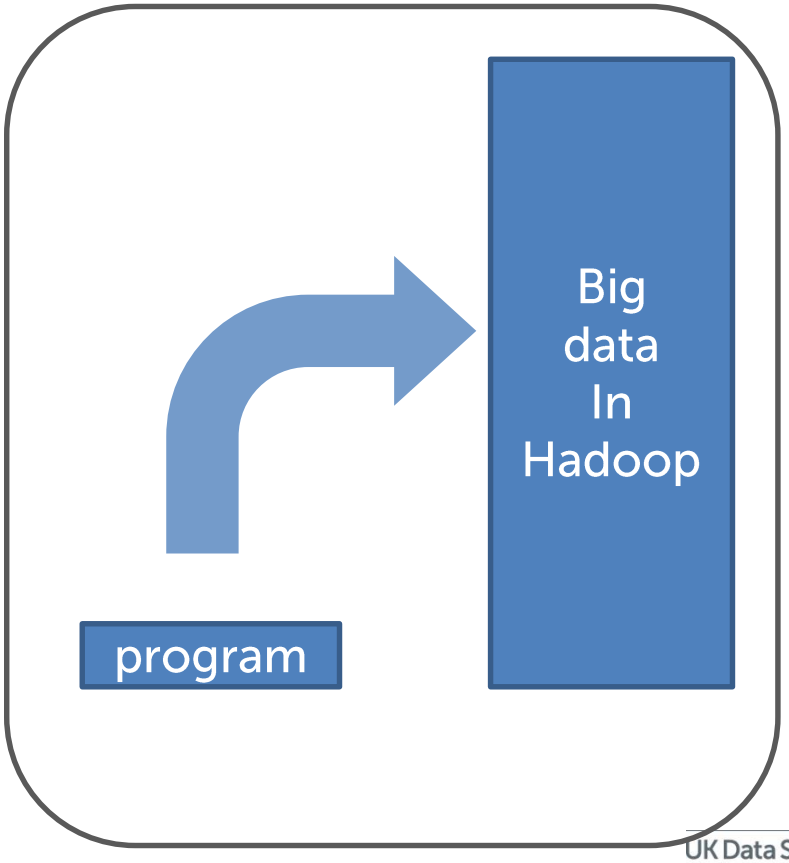
- In traditional Client-Server environment (which Hadoop is NOT) It is normal for the data to be stored on the server (a fileserver)
- Office Applications such as MS Word are typically installed on your desktop
- When you need to edit a Word document, a copy is sent from the server to the Desktop, and you edit there. When you finish, the new version is copied back to server.
- This makes sense because typically the MS Word program is many time larger than the document it is editing

Data + Program = Processing

Traditional client server



Hadoop



Hadoop components

- Map Reduce
  - There are two parts; A Map part and a Reduce part
  - Typically written in Java but can be written in other languages such as Python

Map Reduce – health Warning

- The illustration of the Map-Reduce process that follows is provided for completeness.
- There is very little need to actually code Map-Reduce jobs any more as there are so many alternatives Hadoop components which will either do it for you or provide alternative mechanisms to achieve the same thing

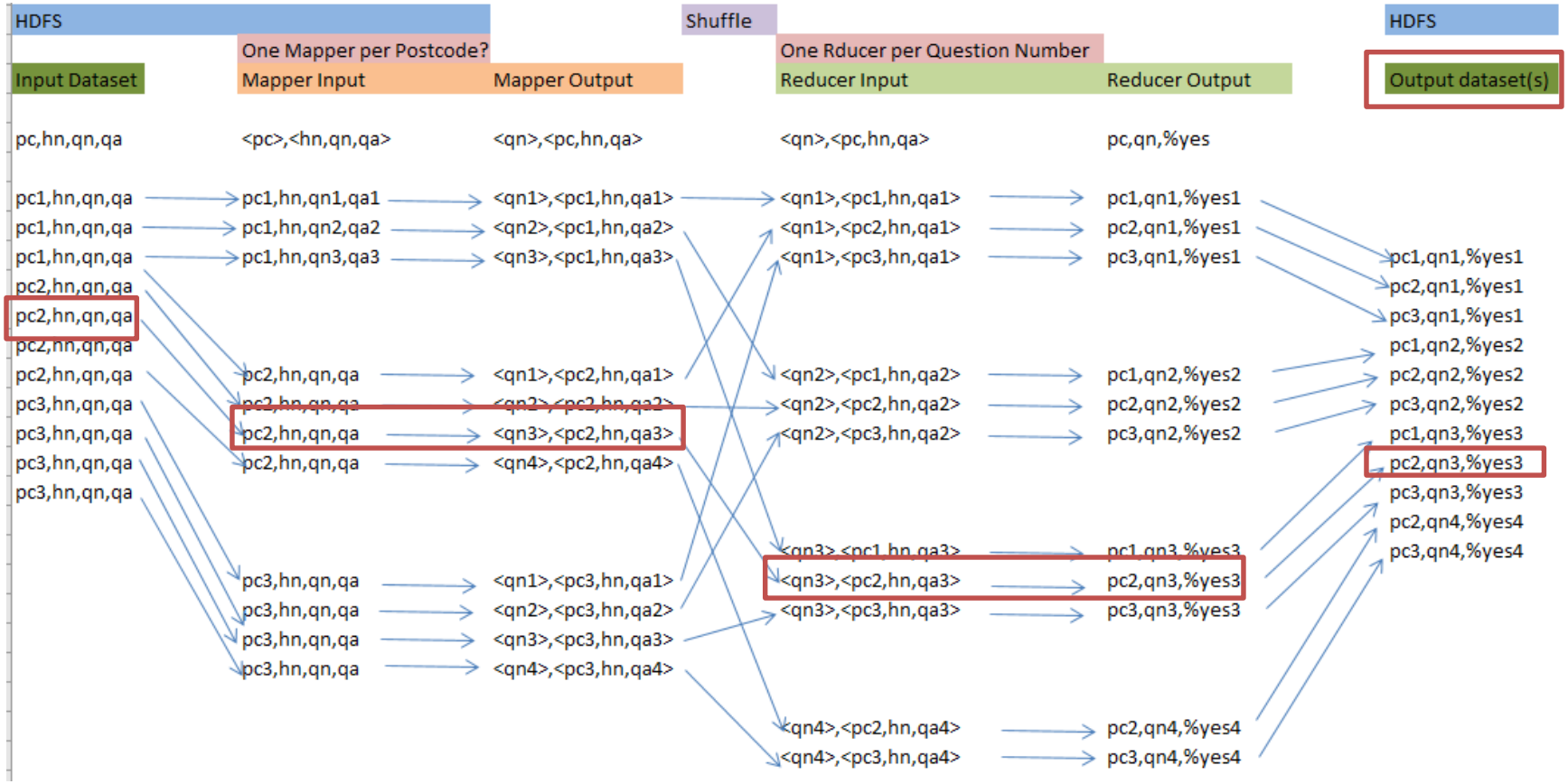
Example Scenario

- You have a Questionnaire of 10 question (yes/no answers)
- The questionnaire is completed by a random number of households in the each Postcode areas starting with ‘M’
- You want to know, by Postcode the % of yes answers for each of the 10 questions

The Map Reduce process

- Storing the dataset in HDFS means that it is automatically split into blocks and spread across multiple datanode.
- The mapper process is sent to each of these nodes to process the block of data on that node
- The records output from the mapper are sent to a datanode on which the reducer process will be run.
- All records with the same key value will be sent to the same datanode, which will be one which will run the reducer process.
- The output records from the reducer process are sent to a file on HDFS, each reducer produces its own output file

The Map Reduce Process



Alternatives to coding MapReduce

- Need to understand what it is you are trying to do with your data
- You only need to use Hadoop and big data tools because your data is too big for your favourite desktop application
- You have two choices
  - Either, use the big data tools to perform any necessary statistical analysis or data mining directly in the Hadoop environment
  - Or use the big data tools to transform and reduce the size of your data – if this is a viable option, and then download the results to your favourite desktop application

Analysis inside Hadoop

- Hadoop add-in products
  - Mahout

  (http://mahout.apache.org/users/basics/algorithms.html)
- Products that work with Hadoop
  - Spark
  - Mllib (Spark Machine Learning Library

  (http://spark.apache.org/mllib/)
- Add-in to Hadoop add-in Hive
  - Hivemall

  (http://docs.treasuredata.com/articles/hive-hivemall)
- Map Reduce Streaming
  - Using Python or R code as part of MapReduce

The Demonstrations – Using Hive

- For the demonstrations, we are assuming you want to transform (reduce) your dataset into something your desktop application can handle
- For this we will use Hive, which is a component of the Hadoop eco-system
- The Hive processing language – HiveQL is based on standard SQL. This makes Hive easily accessible to anyone with SQL experience (probably a lot more common than those with Java experience) and easier to learn than Java if you haven't!

Hadoop Installations

- VM (Virtual Machine) Sandboxes from:
  - Cloudera
  - Hortonworks – This is what we will be using for training
- Cloud VMs – Azure, AWS
- UKDS Cloud Hadoop – under development
- UKDS On-prem Hadoop (for secure data) – under development

Demo Environment

- All of the demonstrations have been run using
  - Hortonworks Hadoop Sandbox v2.3.2
  - Using VM Player installed on a PC with
    - 16Gb Ram
    - Intel I7 processor
- A guide on installing the Hortonworks VM will be on UKDS website in a few weeks
- We will access the system using both:
  - the Sandbox provided Web Interface and
  - A 3rd party Windows based Hive interface package
  - (you can access HDFS and Hive from the command line – if you want to)

Examples of Basic use of Hadoop and Hive

This is what we are going to do

1. Look at the dataset being used
2. Selection and filtering (slice & dice)
3. Generate a sample of records from the data set
4. Aggregation
5. Create a dataset containing a subset of the data

Next months webinar – ‘What is Hive’ will cover these in more detail as well as covering things, like loading data into tables, creating tables descriptions and table joins.

The dataset being used

- The dataset has been generated so as to simulate the scenario used in the Map-Reduce example.
- A random set of answer to the 10 questions were generated for a random number of Houses with postcodes starting with M.
- The actual file used has approx. 640K records.
- This particular example would fit into Excel, but of course there could have been more postcodes or more houses within the postcodes.

questionnaire.pc	questionnaire.hn	questionnaire.qn	questionnaire.qa
M60 3SF	1	1	0
M60 3SF	1	2	1
M60 3SF	1	3	0
M60 3SF	1	4	1
M60 3SF	1	5	1
M60 3SF	1	6	0
M60 3SF	1	7	1
M60 3SF	1	8	1
M60 3SF	1	9	0
M60 3SF	1	10	1
M60 3SF	2	1	0
M60 3SF	2	2	1

Selection and Filtering

```
SELECT pc, qa FROM questionnaire;
```

```
SELECT pc,
       hn,
       qn,
       qa
  FROM questionnaire
 WHERE pc = "M6 5AU";
```


Selection and Filtering

The Results

pc	qa
M60 3SF	0
M60 3SF	1
M60 3SF	0
M60 3SF	1

pc	hn	qn	qa
M6 5AU	1	1	0
M6 5AU	1	2	1
M6 5AU	1	3	0
M6 5AU	1	4	0

Sampling

- The dataset (table) is split into n (64) buckets and the records randomly distributed between them. The records from bucket 3 will be returned

```
SELECT *
  FROM questionnaire TABLESAMPLE (BUCKET 3 OUT OF 64 ON rand ());
```

```
SELECT *
  FROM questionnaire TABLESAMPLE (BUCKET 3 OUT OF 64 ON rand (12345));
```

- If the first select statement is run again, a different set of records will be returned.
- The second select statement will always return the same set of records.

Aggregation

The Code

```
-- Aggregation example
SELECT pc AS Postcode,
          sum (qa) AS Total_yes,
          count (pc) AS total_Questions_in_Postcode
     FROM questionnaire
GROUP BY pc;
```

Aggregation

- The Results

postcode	total_yes	total_questions_in_postcode
M6 5AA	392	780
M6 5AB	485	970
M6 5AD	15	30
M6 5AE	442	920
M6 5AN	91	200
M6 5AQ	356	740

Creating new Tables from the data

- It is possible to create new tables based on the original but with only a limited set of columns and a filtered set of rows.

-

```
CREATE TABLE IF NOT EXISTS questionnaire_postcode
AS
    SELECT pc,
           hn,
           qn,
           qa
      FROM questionnaire
     WHERE pc = "M6 5AU";
```

Creating new Tables from the data

- Select * from questionnaire_postcode;

questionnaire_postcode.pc	questionnaire_postcode.hn	questionnaire_postcode.qn	questionnaire_postcode.qa
M6 5AU	1	1	0
M6 5AU	1	2	1
M6 5AU	1	3	0
M6 5AU	1	4	0
M6 5AU	1	5	1
M6 5AU	1	6	1
M6 5AU	1	7	1
M6 5AU	1	8	0
M6 5AU	1	9	1
M6 5AU	1	10	0

Final Example

- The Quesionnaire_Answers dataset

```
CREATE TABLE questionnaire_answers
AS
     SELECT pc AS Postcode,
            qn AS QuestionNumber,
            round ( (sum (qa) / count (pc)) * 100) AS Yes_Answers
       FROM questionnaire
   GROUP BY pc, qn;
```


Questionnaire_Answers

postcode	questionnumber	yes_answers
M6 5AA	1	51.0
M6 5AA	2	58.0
M6 5AA	3	51.0
M6 5AA	4	50.0
M6 5AA	5	54.0
M6 5AA	6	53.0
M6 5AA	7	47.0
M6 5AA	8	45.0
M6 5AA	9	50.0
M6 5AA	10	44.0
M6 5AB	1	57.0

Summary

- There may be no choice in using Big Data especially if you do not control the source
- Hadoop and Big Data Tools are just that – Tools
- You can use them to do all of your analysis or to cut the data down to size for your preferred desktop application.
- Although Java and Map Reduce are available, there is an increasing number of other tools available which will make the processing simpler.

UK Data Service – Big data

Scaling up for big data

- setting up a Hadoop system
- to process and analyse large datasets
- to analyse safe, safeguarded and secure data

Upcoming webinars

- What is Hive? - 22 March

  Hive is a package that works with Hadoop that allows users to manipulate very large datasets, find out more about Hive and why you might want to use it

- What is Spark? - 19 April

  Spark might be considered as a one-stop tool for big data processing, providing data manipulation facilities to slice and dice datasets as well as statistical functionality and visualisation capabilities to present your results

ukdataservice.ac.uk/news-and-events/events

Questions

Peter Smyth

Peter.smyth@manchester.ac.uk

ukdataservice.ac.uk/help/

Subscribe to the UK Data Service news list at https://www.jiscmail.ac.uk/cgi-bin/webadmin?A0=UKDATASERVICE

Follow us on Twitter https://twitter.com/UKDataService or Facebook https://www.facebook.com/UKDataService