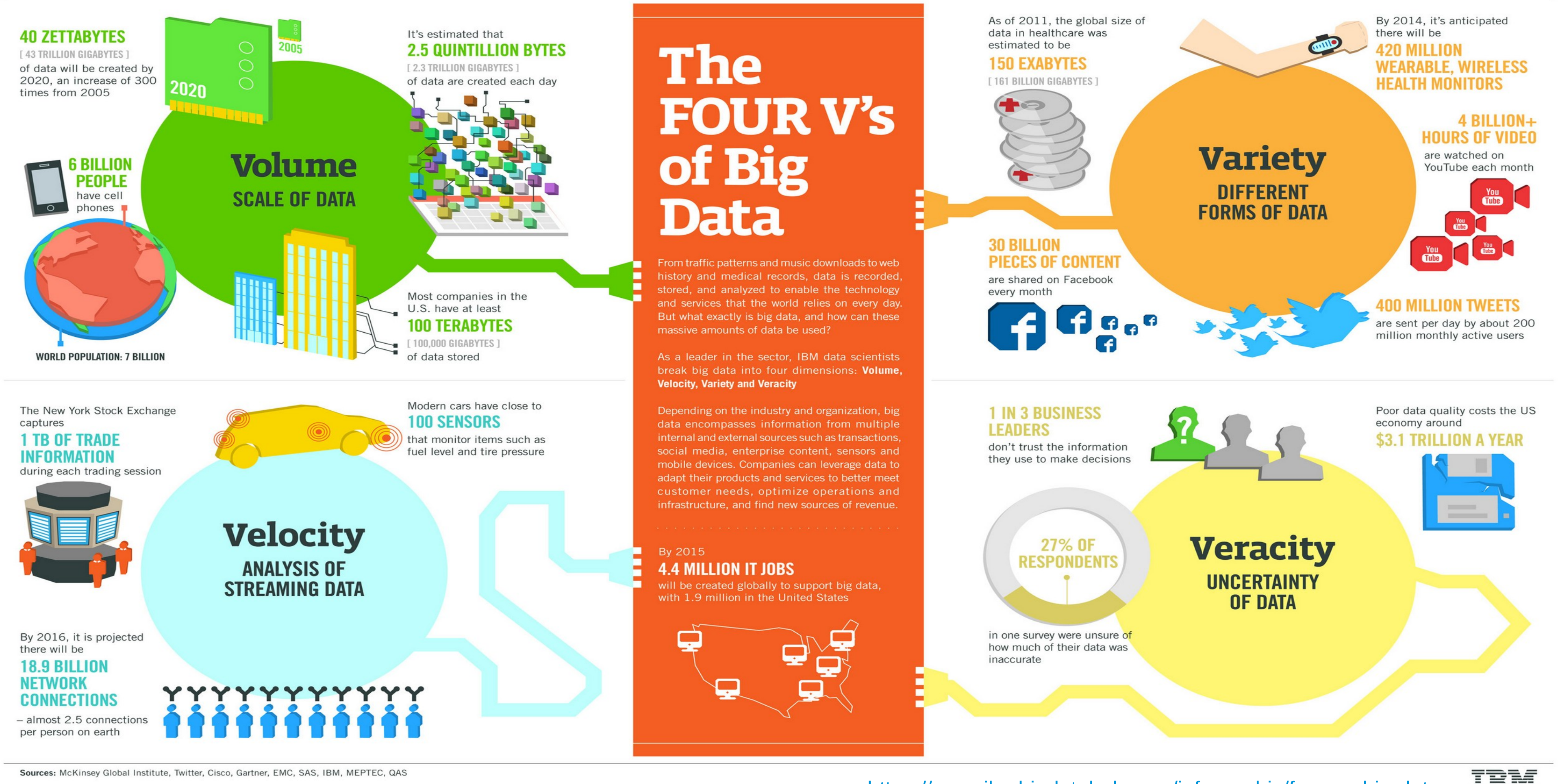
Large Scale Data Management

José Orlando Pereira

Departamento de Informática
Universidade do Minho

Big Data



40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005

6 BILLION PEOPLE have cell phones

WORLD POPULATION: 7 BILLION

It's estimated that 2.5 QUINTILLION BYTES [2.3 TRILLION GIGABYTES] of data are created each day

Most companies in the U.S. have at least 100 TERABYTES [100,000 GIGABYTES] of data stored

Volume
SCALE OF DATA

The New York Stock Exchange captures 1 TB OF TRADE INFORMATION during each trading session

Modern cars have close to 100 SENSORS that monitor items such as fuel level and tire pressure

Velocity
ANALYSIS OF STREAMING DATA

By 2016, it is projected there will be 18.9 BILLION NETWORK CONNECTIONS – almost 2.5 connections per person on earth

The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States

As of 2011, the global size of data in healthcare was estimated to be 150 EXABYTES [161 BILLION GIGABYTES]

By 2014, it's anticipated there will be 420 MILLION WEARABLE, WIRELESS HEALTH MONITORS

4 BILLION+ HOURS OF VIDEO are watched on YouTube each month

30 BILLION PIECES OF CONTENT are shared on Facebook every month

400 MILLION TWEETS are sent per day by about 200 million monthly active users

Variety
DIFFERENT FORMS OF DATA

1 IN 3 BUSINESS LEADERS don't trust the information they use to make decisions

Poor data quality costs the US economy around $3.1 TRILLION A YEAR

27% OF RESPONDENTS in one survey were unsure of how much of their data was inaccurate

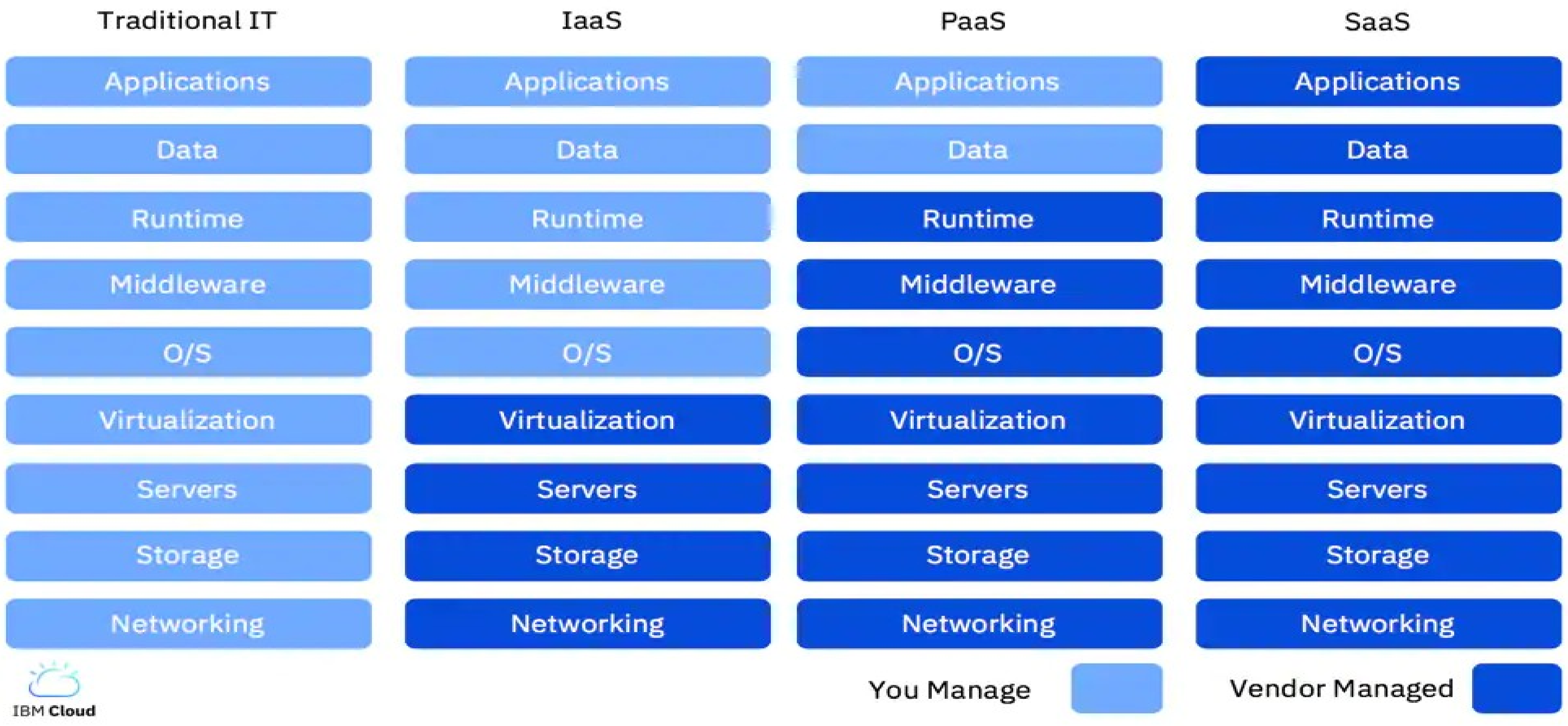
Veracity
UNCERTAINTY OF DATA

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

IBM

https://www.ibmbigdatahub.com/infographic/four-vs-big-data

Cloud computing



Traditional IT	IaaS	PaaS	SaaS
Applications 🟦	Applications 🟦	Applications 🟦	Applications ■
Data 🟦	Data 🟦	Data 🟦	Data ■
Runtime 🟦	Runtime 🟦	Runtime ■	Runtime ■
Middleware 🟦	Middleware 🟦	Middleware ■	Middleware ■
O/S 🟦	O/S 🟦	O/S ■	O/S ■
Virtualization 🟦	Virtualization ■	Virtualization ■	Virtualization ■
Servers 🟦	Servers ■	Servers ■	Servers ■
Storage 🟦	Storage ■	Storage ■	Storage ■
Networking 🟦	Networking ■	Networking ■	Networking ■

IBM Cloud

You Manage 🟦 Vendor Managed ■

https://www.ibm.com/cloud/learn/iaas

Contents

- Distributed processing (Map-Reduce)
- Scalable storage (HDFS and HBase)
- Data flow and streaming (Spark)

Main references

- M. Tamer Özsu, Patrick Valduriez. **Principles of Distributed Database Systems** (3rd Edition). Springer.
  - Chapter 18
- Peter Bailis, Joseph M. Hellerstein, Michael Stonebraker. **Readings in Database Systems** (5th Edition)
  - Chapter 5: http://www.redbook.io/ch5-dataflow.html
    - https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf
    - https://storageconference.us/2010/Papers/MSST/Shvachko.pdf
    - https://storage.googleapis.com/pub-tools-public-publication-data/pdf/68a74a85e1662fe02ff3967497f31fda7f32225c.pdf
    - http://people.csail.mit.edu/matei/papers/2012/nsdi_spark.pdf

Additional references

- Tom White. **Hadoop: The Definitive Guide** (4th Edition). O’Reilly.
- Lars George. **HBase: The Definitive Guide**. O’Reilly.
- Holden Karau, Andy Konwinski, Patrick Wendell & Matei Zaharia. **Spark: Lightning-fast data analysis**. O’Reilly.
- Software manuals

Grading

- Group projects
  - First project (MapReduce)
  - Second project (Spark)
  - 50% weight
- Written exam
  - 50% weight
  - Minimum: 8 / 20

Maven™

Maven Build

- Automatic dependency downloading and packaging
- Set Java version:

```
<properties>
    <maven.compiler.source>8</maven.compiler.source>
    <maven.compiler.target>8</maven.compiler.target>
</properties>
```

- Packaging:

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-jar-plugin</artifactId>
    <configuration>
        <archive>
            <manifest>
                <addClasspath>true</addClasspath>
               <classpathPrefix>libs/</classpathPrefix>
                <mainClass>
                    packagename.MainClass
                </mainClass>
            </manifest>
        </archive>
    </configuration>
</plugin>
```

Maven Build

- Sample dependency:

```
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-compress</artifactId>
    <version>1.20</version>
</dependency>
```

- Dependency packaging:

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-shade-plugin</artifactId>
    <version>3.2.2</version>
    <executions>
        <execution>
            <phase>package</phase>
            <goals>
                <goal>shade</goal>
            </goals>
            <configuration>
                <transformers>
                    <transformer implementation=
"org.apache.maven.plugins.shade.resource.Services
ResourceTransformer"/>
                </transformers>
            </configuration>
        </execution>
    </executions>
</plugin>
```

Docker Setup

- Installation guide:

  https://docs.docker.com/get-docker/

  (do not skip "Post-installation steps for Linux"!!!)

- Testing:

```
$ docker ps
$ docker pull hello-world
$ docker run hello-world
$ docker ps -a
$ docker rm containername
```

Docker Basics

- Running a container:

```
$ docker run -it ubuntu
$ docker run -it --name contname ubuntu
```

- Entering into an existing container

```
$ docker ps
$ docker exec -it contname bash
```

- Stopping and cleaning up:

```
$ docker kill contname
$ docker ps -a
$ docker rm contname
```

Packaging a Java application

- Sample Dockerfile:

```
FROM openjdk:8
COPY target/jarname.jar /
ENTRYPOINT ["java", "-jar", "/jarname.jar"]
```

- Build with:

```
$ docker build -t imagename .
$ docker images
```

- Run with:

```
$ docker run -it --name contname imagename arg1 arg2 …
$ docker rm contname
$ docker rmi imagename
```

Accessing host files

- The container and the host have separate file-systems
  - Map a folder from the host into the container
  - Reference files in that folder

```
$ docker run -it \
    -v /hostfolder:/containerfolder
    imagename /containerfolder/filename
```