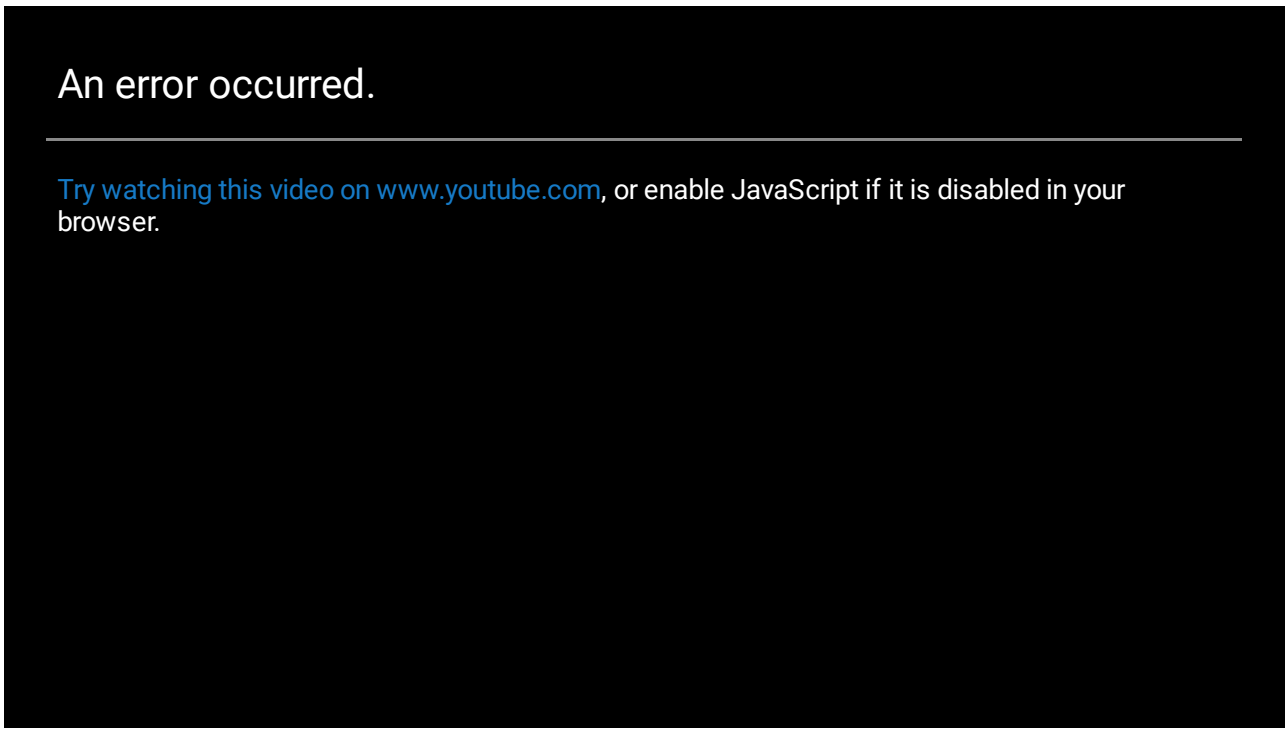
Hadoop

Origins of Hadoop (3:12)



Hadoop

This week we start considering how to store and process enormous volumes of data.

When you have a small volume of data you can store it on a single computer. If your volume of data increases and you run out of space on that computer then you can switch to a larger and more powerful one. This is called **vertical scaling**.

There is only so much vertical scaling you can do - there is a limit to how large and powerful a single computer can be, and even before you reach that limit the computer might cost more than you can afford.

The solution is **horizontal scaling** - distributing your data across a cluster of computers (each of these computers in the cluster is called a **node**). These computers can be ordinary, low cost, commodity computers, so this is a cheap way to store large volumes of data. It is also extremely scalable - as your data grows you can just add more computers to your cluster, as many as thousands if you need them.

You will need a way to manage the distribution of data across these machines. A popular way is to use **Hadoop**, a piece of software that runs on these machines and manages the process for you (including making replicas of your data, in case of problems).

About Hadoop

Hadoop manages the storing of files across a cluster of nodes using what's called the **Hadoop Distributed File System** (**HDFS**).

One of nodes in the cluster is special, called the **name node**. It keeps track of where the data is located throughout the cluster, and manages the names of directories and files (thus its name). You can think of the name node as the **master** node.

The rest of the nodes are called **data nodes**, because they store the data. You can think of these as **slave** nodes.

When you use Hadoop and work with files in your HDFS you do so by using a client computer, which often sits outside the cluster and merely communicates with it. From the client computer, you can work with files in HDFS, just like you would work with files in a regular file system (you will learn how to do this in a later slide).

When you save a file to HDFS, Hadoop automatically makes replicas of it, in case of machine failure. This **fault tolerance** is one of the key features of HDFS. How many replicas it makes is determined by the **replication factor** you have set. The default is three. The name node monitors for faults, by

listening for 'heartbeat' signals from the data nodes. If it fails to get a scheduled heartbeat signal from a certain data node then it assumes that the data node is broken. The name node then draws upon one of the replicas of the data that was on that node, makes a new replica, and distributes the new replica across the remaining nodes, thereby making sure that everything remains replicated to the set factor.

Hadoop is not designed for quick reading and writing of files. Rather, it is designed to work more like a data warehouse. You might hear it said that Hadoop is **Write Once, Read Many** (WORM).

You can also have Hadoop running on just a single computer, which is often done for educational purposes. In this case, Hadoop is said to be in **standalone** mode. In the examples that you will be using in this course, Hadoop is running in standalone mode.

History of Hadoop

Hadoop was inspired by a similar system that Google was working on. In 2003, Google engineers published a paper describing their so-called **Google File System** (GFS), and then in 2004 they published a second paper describing a method for querying data on GFS, called **MapReduce**.

Doug Cutting recognised the importance of these two papers, and starting working on implementing the ideas with Mike Cafarella. In 2006, while working for Yahoo, Cutting developed a reasonably stable system and named it "Hadoop", after his son's toy elephant. Yahoo later decided to make it open-source, part of the Apache Software Foundation.

There have been three versions of Hadoop. All three have contained the HDFS and MapReduce components; a third component called **YARN** was introduced in the second version.

- 2011: Hadoop 1.0.0: HDFS + MapReduce
- 2012: Hadoop 2.0.0: HDFS + MapReduce + YARN
- 2017: Hadoop 3.0.0: HDFS + MapReduce + YARN

There are also commercial versions of Hadoop available, from companies such as Cloudera, Hortonworks, Amazon, and Microsoft.

More about Hadoop

In the series of slides that follow this slide are some videos that explain the important features of Hadoop.

They overlap, so you will hear some things explained more than once, but they are explained in slightly different ways, which you might find helpful.

After that, you'll get some hands-on experience using Hadoop HDFS.

Introduction to Hadoop

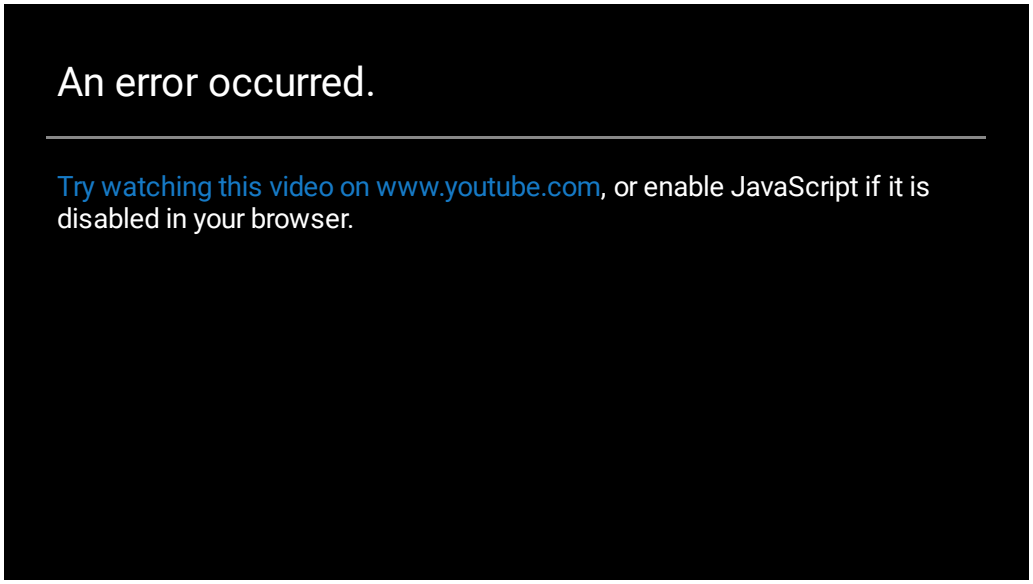
Introduction to Hadoop

ZZEN9313_Hadoop_Transcript.pdf

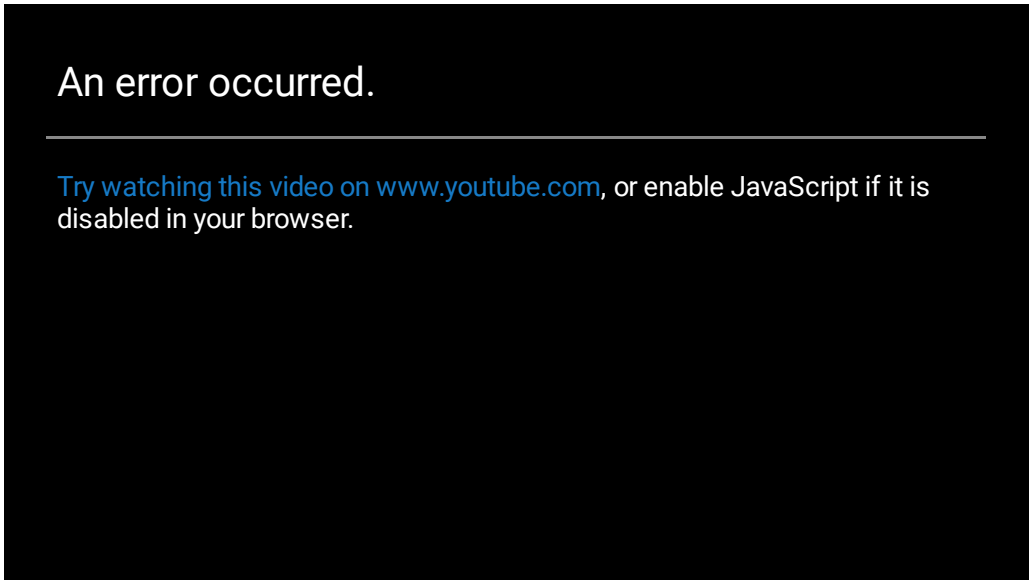
More videos that can help you understand Hadoop

Some videos that may help you better understand Hadoop:

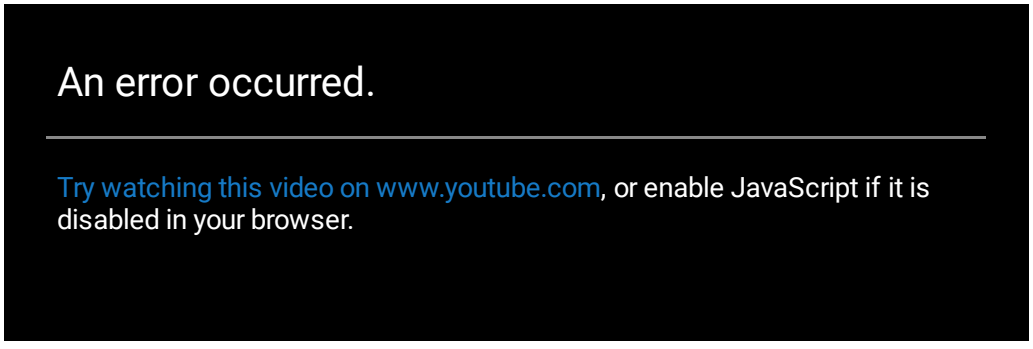
1. Hadoop Tutorial - Introduction



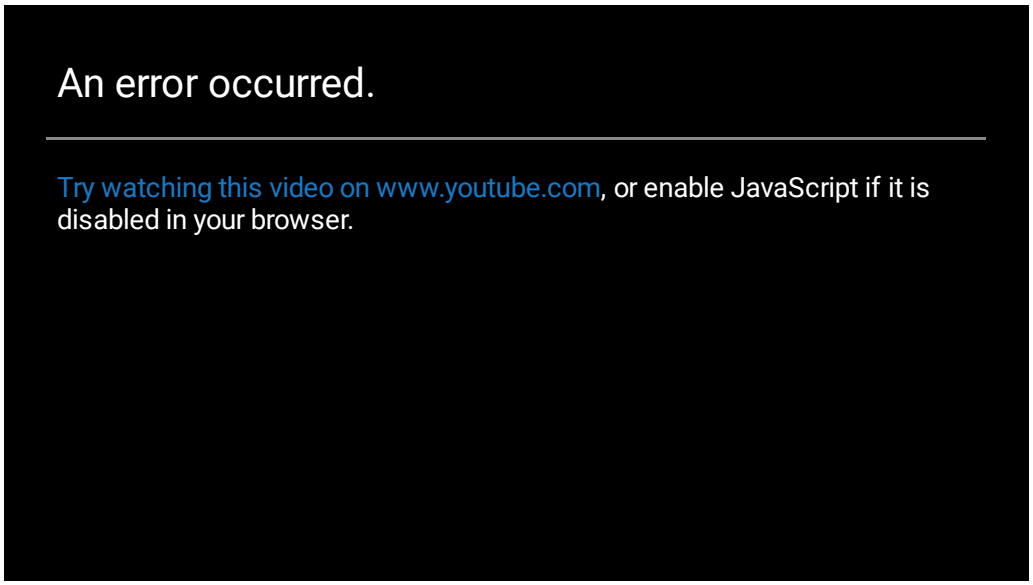
2. MicroNugget: What is Hadoop?



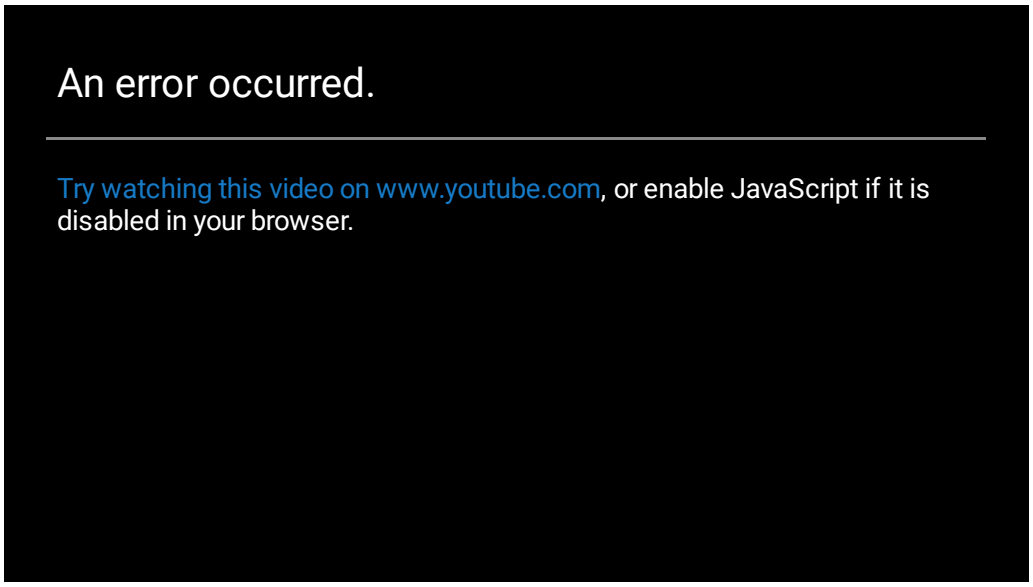
3. What is Big Data and Hadoop?



4. Comparing Hadoop and SQL



5. Hadoop Tutorial - The YARN



Futher Resources

If you're interested in the technicalities of Hadoop and MapReduce then you might find it worthwhile to look at the two Google papers at the origins:

The Google paper on Google File System

The Google paper on MapReduce

The Hadoop official website is a good source of further information about Hadoop:

The Hadoop official website

For example, if you would like to install Hadoop on your own computer, you can refer to this page "Hadoop: Setting up a Single Node Cluster" at the official website.