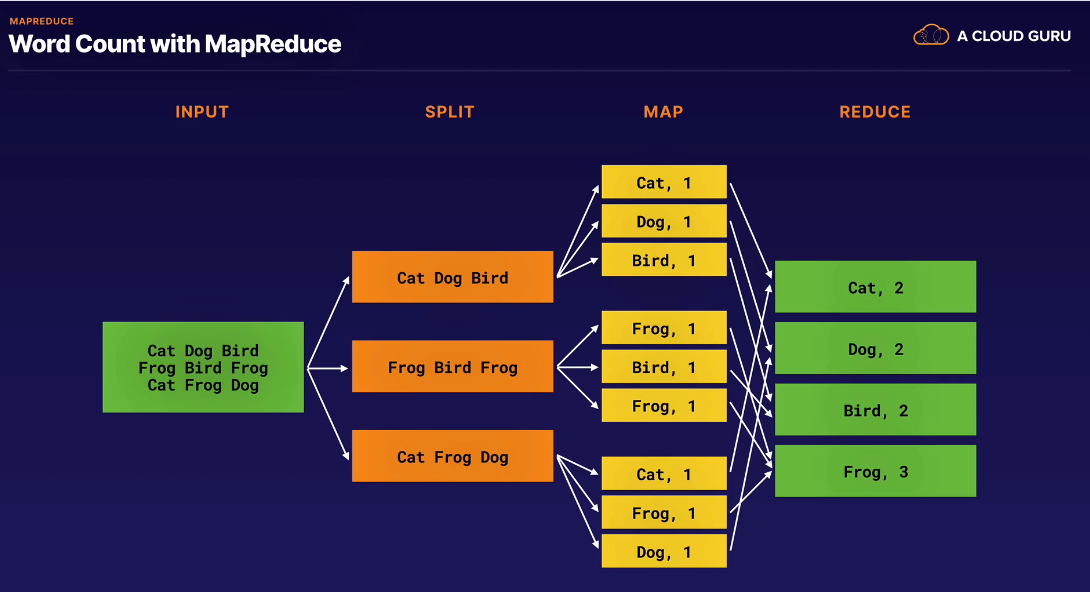
**Big Data Ecosystem:**

* Fundamentals of Big Data.

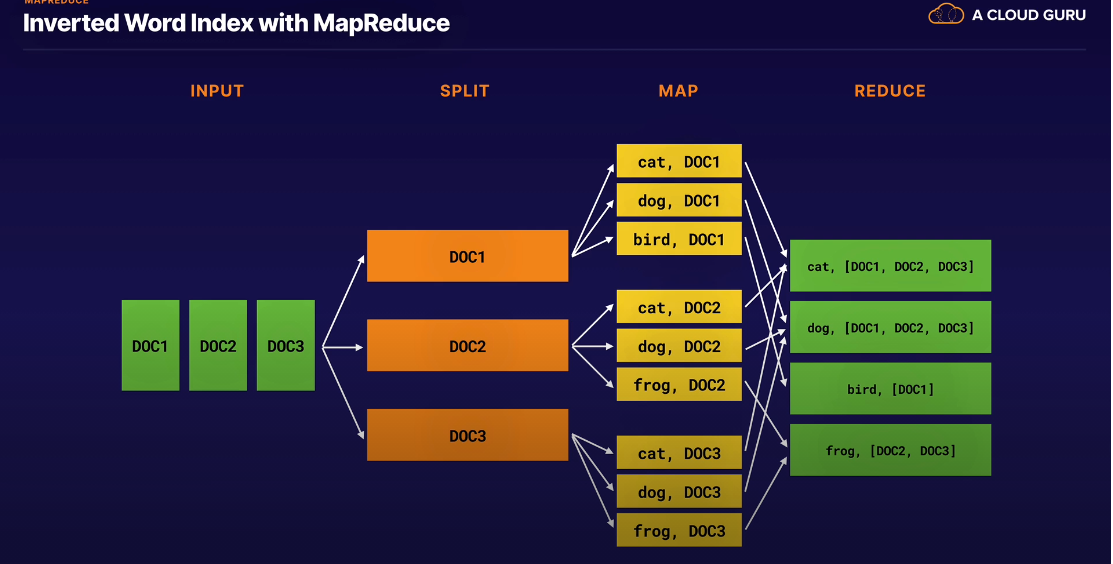
MapReduce:

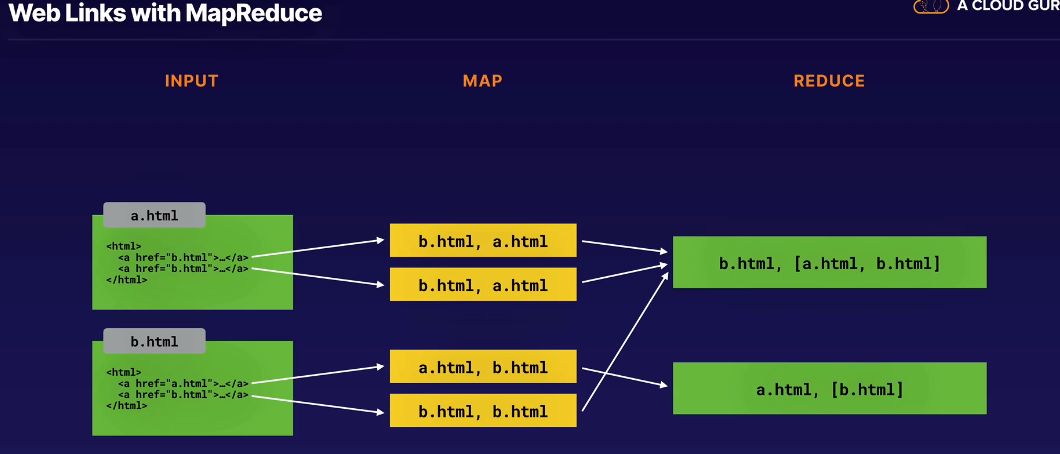
What is MapReduce:

* It’s actually 2 things:
  + It’s a programming model. – it is a design of how you can write a computer programs using two specific functions. – Map and Reduce – and how they’re combined to perform a task.
  + It’s also an implementation of that model – Distributed Implementation. - way of running those programs across multiple computers.
  + Created at Google.
  + Provided a common framework.
* Map function
  + Takes an input from the user.
  + And produces a set of intermediate key/value pairs. – keys are elements of the input we’ve given our function; value is just one. – intermediate means we are not done yet, we’re going to pass key/value pairs to the reduce function.
* Reduce function
  + Merges intermediate values we pass to it with others /merge intermediate values associated with the same intermediate key.
  + In most cases, this means we are going to end up with smaller set of values – hence the term reduce.









MapReduce Implementation:

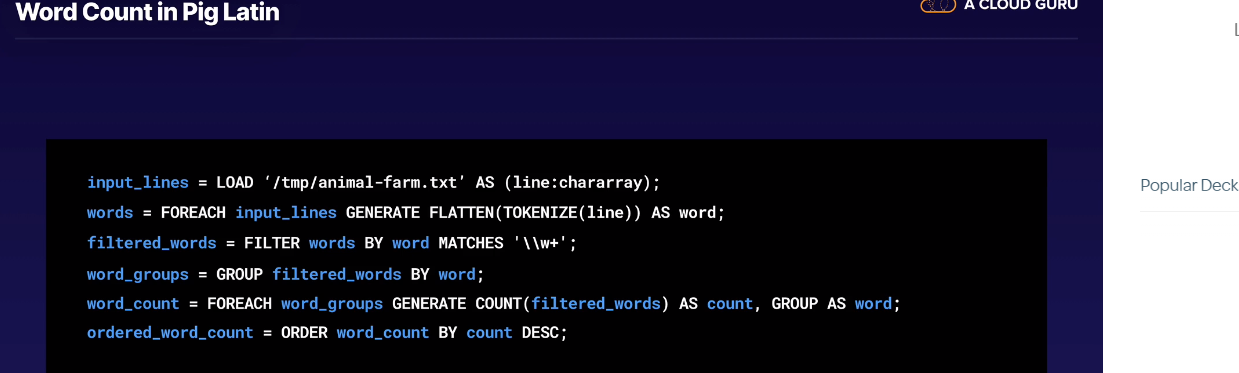
* Standard framework.
* Implementation abstracts away the distributed computing framework. Parallelizing and executing.
* MapReduce jobs split jobs into small chunks upto about 64MBs of data - which are schedule into a cluster using Master and worker cluster model.
* Master periodically undertakes health checks on worker nodes.
* Failed worker jobs are reassigned
* Worker files buffered to local disk
* Partitioned output files.

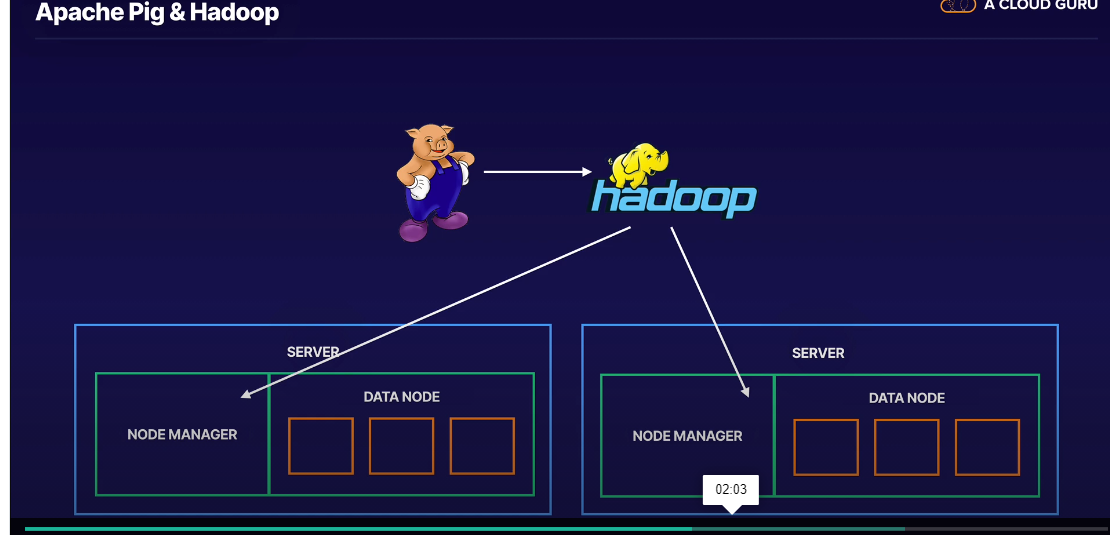
Hadoop & HDFS:

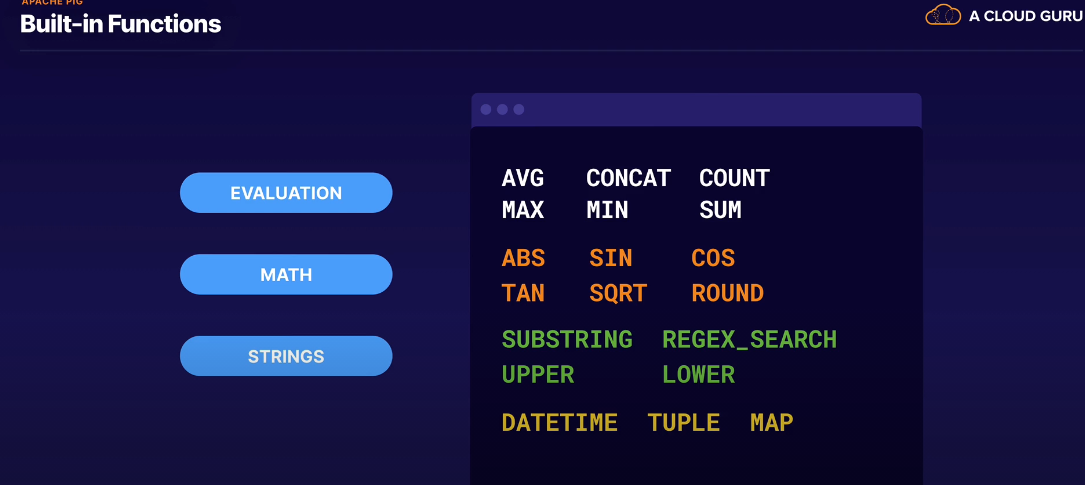
* Apache Open-Source Implementation of MapReduce and Cluster Model framework.
* What is Hadoop:
  + Named after a toy elephant.
  + Inspired by Google File System & MapReduce R&D papers.
  + Originated in Apache Nutch
  + Hadoop subproject started in 2006.
* Hadoop Now Comprises of 4 core Models:
  + *Hadoop Common* – base framework of Hadoop containing all the libraries, OS system abstractions & startup scrips
  + *Hadoop Distributed File System (HDFS)* – distributed fault tolerant filesystem. - runs on commodity HW.
  + *Hadoop YARN* – handles the task of resource mgt, job scheduling and monitoring for Hadoop jobs.
  + *Hadoop MapReduce* -MapReduce implementation – include libraries for Map and Reduce functions, partitioning, reduction & custom job config parameters.

Apache Pig:

* Platform for analyzing large datasets.
* It’s basically a language called Pig Latin which defines analytics jobs.; which may include Merging, Filtering, and Transformation.
* It’s a high-level language which can be used to define job
* An abstraction for MapReduce
* Used to write ETL.
* A procedural data flows.





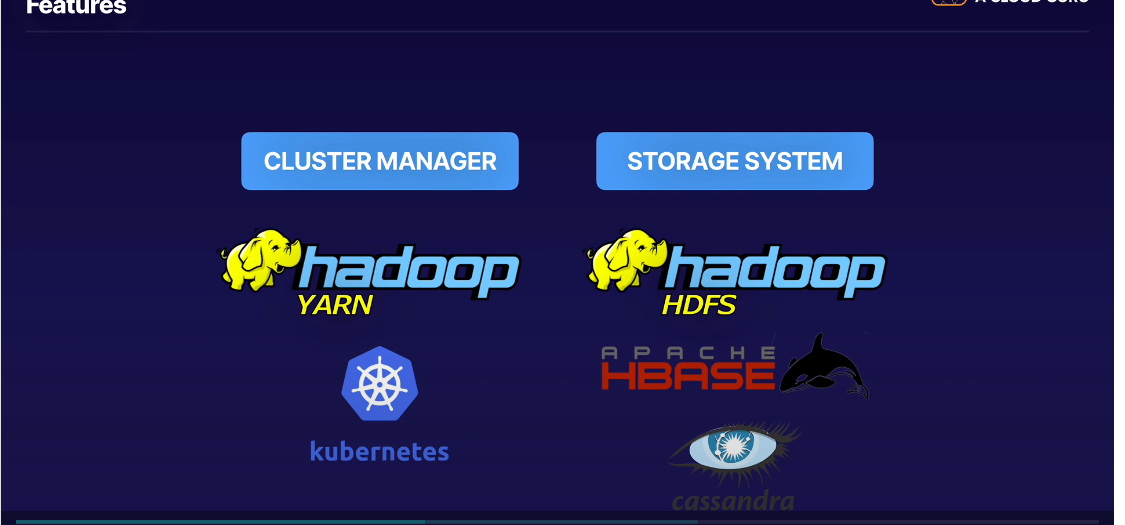


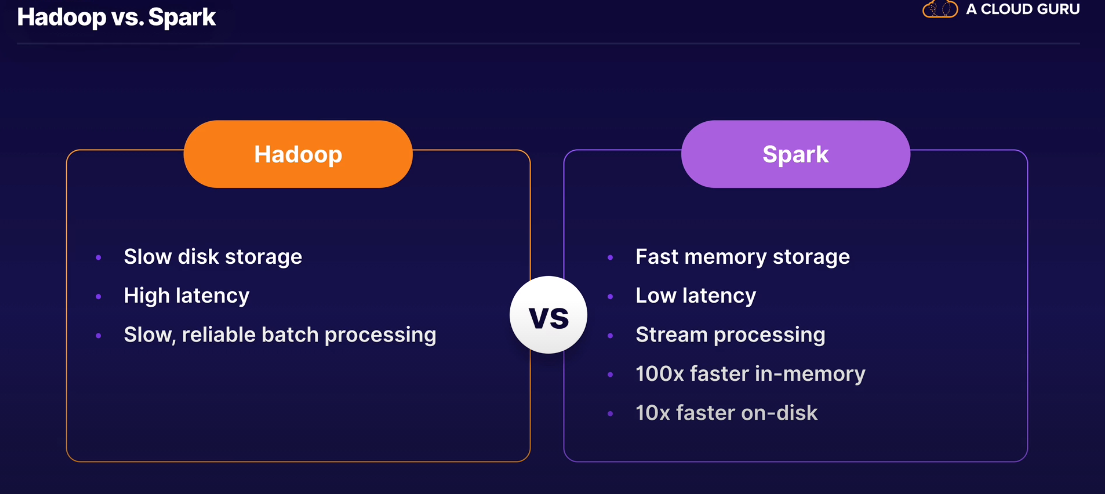
* + - Pig is a high-level framework for running MapReduce jobs on Hadoop clusters.

**Apache Spark:**

* Limits of MapReduce:

1. Linear Dataflow:
2. Read Data
3. Map Functions across data.
4. Reduce results.
5. Write to disk.
   * + Linear constraint made it harder to run complex calculations without waiting along time.
   * In response to the limitation, Apache Spark was created
   * Spark:
     + General purpose cluster-computing framework.
     + Uses the concept of resilient distributed data multisets.
     + Working sets as a form of distributed shared memory.
     + Spark supports 4 std modules – although there are 3rd party projects. That extend Spark’s functionalities.
     + **Spark SQL:**
       - Designed for working with structured data using a familiar SQL syntax.
       - Structured data in Spark is stored in an abstraction called a Data frame.
       - For programmatic querying, devs can access data frames API rather than embed SQL into their code.
     + **Spark Streaming:**
       - This module allows for streaming data ingestion configuration.
       - In reality, it uses very small batches to achieve this.
       - You can use the same code for batch processing, join streams against historical data or run ad-hoc queries on the state of current stream.
     + **MLLib:**
       - Sparks ML library; supports all ML libraries. - specification, regression, decision trees, recommendation etc.
     + **GraphX:**
       - A module specifically designed for iterative graph computation – to save you the hustle of stitching together numerous MapReduce jobs.
   * Spark Supports common languages: Java, Scala, Python, R, SQL
   * To run a Spark cluster, you need 2 things:
     + **Cluster Manager** and **Distributed Storage System.**





Apache Kafka:

* A distributed streaming platform.
* Allows you to publish/subscribe to streams of records.
* You can think of it like a message bus, but for data.
* Designed for High-throughput and low-latency.
* Originated at LinkedIn – Handling >800 billion messages per day.
* 4 main APIs in Kafka:
* Producer – allows for an app to publish a stream of records to a Kafka topic.
* Consumer – allows an app to subscribe to one or more topics and process stream of records contained.
* Streams – API designed to allow an app to act as a stream processor itself.- useful when you wish to transform some data and feed it straight back into Kafka as an outgoing stream.
* Connector -allows you to extend Kafka by connecting producers or consumers to external DBs e.g RD.

