Spark

**CPU (Central Processing Unit)**  
The CPU is the "brain" of the computer. Every process on your computer is eventually handled by your CPU. This includes calculations and also instructions for the other components of the compute.

**Memory (RAM)**  
When your program runs, data gets temporarily stored in memory before getting sent to the CPU. Memory is ephemeral storage - when your computer shuts down, the data in the memory is lost.

**Storage (SSD or Magnetic Disk)**  
Storage is used for keeping data over long periods of time. When a program runs, the CPU will direct the memory to temporarily load data from long-term storage.

**Network (LAN or the Internet)**  
Network is the gateway for anything that you need that isn't stored on your computer. The network could connect to other computers in the same room (a Local Area Network) or to a computer on the other side of the world, connected over the internet.

**About CPU**

The CPU is the brains of a computer. The CPU has a few different functions including directing other components of a computer as well as running mathematical calculations. The CPU can also store small amounts of data inside itself in what are called **registers**. These registers hold data that the CPU is working with at the moment.

For example, say you write a program that reads in a 40 MB data file and then analyzes the file. When you execute the code, the instructions are loaded into the CPU. The CPU then instructs the computer to take the 40 MB from disk and store the data in memory (RAM). If you want to sum a column of data, then the CPU will essentially take two numbers at a time and sum them together. The accumulation of the sum needs to be stored somewhere while the CPU grabs the next number.

This cumulative sum will be stored in a register. The registers make computations more efficient: the registers avoid having to send data unnecessarily back and forth between memory (RAM) and the CPU.

A screenshot of a cell phone

Description automatically generated

**Hadoop Vocabulary**

Here is a list of some terms associated with Hadoop. You'll learn more about these terms and how they relate to Spark in the rest of the lesson.

* [**Hadoop**](https://hadoop.apache.org/) - an ecosystem of tools for big data storage and data analysis. Hadoop is an older system than Spark but is still used by many companies. The major difference between Spark and Hadoop is how they use memory. Hadoop writes intermediate results to disk whereas Spark tries to keep data in memory whenever possible. This makes Spark faster for many use cases.
* **Hadoop MapReduce** - a system for processing and analyzing large data sets in parallel.
* **Hadoop YARN** - a resource manager that schedules jobs across a cluster. The manager keeps track of what computer resources are available and then assigns those resources to specific tasks.
* **Hadoop Distributed File System (HDFS)** - a big data storage system that splits data into chunks and stores the chunks across a cluster of computers.

As Hadoop matured, other tools were developed to make Hadoop easier to work with. These tools included:

* **Apache Pig** - a SQL-like language that runs on top of Hadoop MapReduce
* **Apache Hive** - another SQL-like interface that runs on top of Hadoop MapReduce

Oftentimes when someone is talking about Hadoop in general terms, they are actually talking about Hadoop MapReduce. However, Hadoop is more than just MapReduce. In the next part of the lesson, you'll learn more about how MapReduce works.

**How is Spark related to Hadoop?**

Spark, which is the main focus of this course, is another big data framework. Spark contains libraries for data analysis, machine learning, graph analysis, and streaming live data. Spark is generally faster than Hadoop. This is because Hadoop writes intermediate results to disk whereas Spark tries to keep intermediate results in memory whenever possible.

The Hadoop ecosystem includes a distributed file storage system called HDFS (Hadoop Distributed File System). Spark, on the other hand, does not include a file storage system. You can use Spark on top of HDFS but you do not have to. Spark can read in data from other sources as well such as [**Amazon S3**](https://aws.amazon.com/s3/).

**Streaming Data**

Data streaming is a specialized topic in big data. The use case is when you want to store and analyze data in real-time such as Facebook posts or Twitter tweets.

Spark has a streaming library called [**Spark Streaming**](https://spark.apache.org/docs/latest/streaming-programming-guide.html) although it is not as popular and fast as some other streaming libraries. Other popular streaming libraries include [**Storm**](http://storm.apache.org/) and **[Flink](https://flink.apache.org/" \t "_blank)**. Streaming won't be covered in this course, but you can follow these links to learn more about these technologies.

MapReduce is a programming technique for manipulating large data sets. "Hadoop MapReduce" is a specific implementation of this programming technique.

The technique works by first dividing up a large dataset and distributing the data across a cluster. In the map step, each data is analyzed and converted into a (key, value) pair. Then these key-value pairs are shuffled across the cluster so that all keys are on the same machine. In the reduce step, the values with the same keys are combined together.

While Spark doesn't implement MapReduce, you can write Spark programs that behave in a similar way to the map-reduce paradigm. In the next section, you will run through a code example.