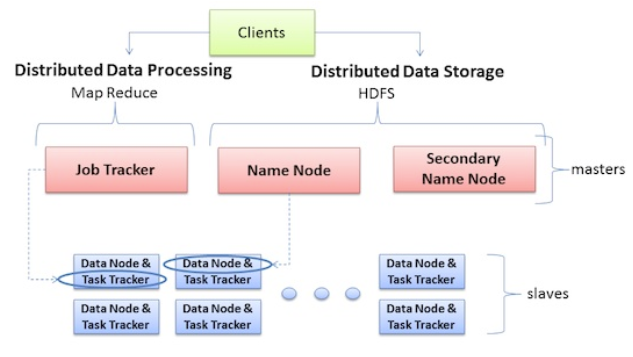
* **What is Hadoop?**
  + Open-source software platform designed to store and process quantities of data that are too large for just one particular device or server.
  + Strength
    - Its ability to scale across thousands of commodity servers that don’t share memory or disk space.
* **How it works**
  + Hadoop delegates tasks across servers (called worker/slave nodes), harnessing the power of each device and running them together simultaneously.
    - Allowing massive amounts of data to be analyzed.
* **Hadoop Ecosystem:**
  + Comprised of many different components that all work together to create a single platform.
  + **Two Key Components of the Ecosystem**
    - **Storage of Data (Hadoop Distributed File System (HDFS))**
    - **Framework for running parallel computations (MapReduce)**
* **Hadoop Distributed File System (HDFS):**
  + A scalable file system that distributes and stores data across all machines in a Hadoop cluster (a group of servers). Each HDFS cluster contains:
    - **NameNode:** Runs on a “master node” that tracks and directs the storage of the cluster.
    - **DataNode:** Runs on “slave nodes,” which make up the majority of the machines within a cluster. The NameNode instructs data files to be split into blocks, each of which are replicated three times and stored on machines across the cluster. These replicas ensure the entire system won’t go down if one server fails or is taken offline—known as “fault tolerance.”
    - **Client Machine:** neither a NameNode or a DataNode, Client machines have Hadoop installed on them. They’re responsible for loading data into the cluster, submitting MapReduce jobs and viewing the results of the job once complete.
* **MapReduce**
  + MapReduce is the system used to efficiently process the large amount of data Hadoop stores in HDFS.
  + Originally created by Google, its strength lies in the ability to divide a single large data processing job into smaller tasks. **All MapReduce jobs are written in Java, but other languages can be used via the Hadoop Streaming API**, which is a utility that comes with Hadoop.
  + **Delegation of Tasks**
    - Handled by two “daemons”
      * **JobTracker:** oversees how MapReduce jobs are split up into tasks and divided among nodes within the cluster.
      * **TaskTracker:** accepts tasks from the JobTracker, performs the work and alerts the JobTracker once it’s done.
        + The TaskTrackers and DataNodes are located on the same nodes to improve performance.
  + When a MapReduce job is submitted, part of what the **JobTracker** does is look to see **which machines the blocks required for the task are located** on.
    - This is why, when the **NameNode** splits data files into blocks, **each one is replicated three times**: the **first is stored on the same machine as the block, while the second and third are each stored on separate machines.**
    - Storing the data across three machines gives you a much higher chance of achieving **data locality**.
      * Since it’s likely that at least one of the machines will be freed up enough to process the data stored at that particular location.
* **Hadoop Ecosystem:**
* 
  + **Data Locality:**
    - Bringing the compute to the data
    - Whenever you use a MapReduce program on a particular part of HDFS data, you always want to run that program on the node, or machine, that actually stores this data in HDFS.
    - This allows processes to be run much faster and prevents you from having to move large amounts of data around.
* **Yet Another Resource Negotiator (YARN)**
  + An updated way of handling the delegation of resources for MapReduce jobs.
  + Takes the place of the **JobTracker**  and **TaskTracker**.
  + Gives added abilities, such as:
    - The ability to work with frameworks other than MapReduce and to translate jobs developed in languages other than Java.