YARN (Yet Another Resource Negotiator)

YARN is Hadoop's resource management system introduced in Hadoop 2.x to enhance scalability and flexibility, primarily for MapReduce but also for other distributed programming paradigms. It replaces the Job Tracker and Task Tracker roles from Hadoop 1.x with a more modular architecture.

YARN Components

1. **Resource Manager (RM)**:
    - Oversees resource allocation and job scheduling across the Hadoop cluster.
2. **Application Master (AM)**:
    - Manages the lifecycle of a specific application (e.g., a MapReduce job).
    - Communicates with the Resource Manager to request and allocate containers.
3. **Node Manager (NM)**:
    - Runs on each node, managing containers and monitoring their resource usage.
    - Reports the status of containers and data nodes to the Resource Manager.
4. **Container**:
    - A unit of resource allocation (CPU, memory) that executes application-specific processes.

Anatomy of a Job Run in YARN

1. A client submits an application to the **Resource Manager**.
2. The RM allocates a container to launch the **Application Master**.
3. The AM registers with the RM and requests additional containers for tasks.
4. The AM instructs **Node Managers** to launch containers for task execution.
5. Application code runs within the containers.
6. The client monitors job progress by contacting the RM or AM.
7. Upon completion, the AM unregisters from the RM, and resources are released.

Job Completion in YARN

A job completes when all tasks are executed successfully, with YARN dynamically managing resources and handling failures (e.g., restarting failed containers).

Hadoop Streaming

Hadoop Streaming is a utility that enables running MapReduce jobs using any executable or script (e.g., shell scripts, Python, Perl) instead of Java code. It facilitates custom processing by reading input from **STDIN** and writing output to **STDOUT**.

Hadoop Streaming Workflow

1. **Mapper Executable**:
    - Each mapper task launches the specified executable as a separate process.
    - The mapper reads input data, converts it into lines, and feeds them to the executable's **STDIN**.
    - The executable processes the input and writes key-value pairs to **STDOUT**.
    - By default, the line prefix (up to the first tab) is the key, and the rest is the value. If no tab exists, the entire line is the key, and the value is null. This behavior can be customized.
2. **Reducer Executable**:
    - Each reducer task launches the specified executable as a separate process.
    - The reducer converts input key-value pairs into lines and feeds them to the executable's **STDIN**.
    - The executable processes the input and writes key-value pairs to **STDOUT**.
    - Output parsing follows the same default key-value convention as the mapper, with customizable options.

Additional Features

- **Python Code for Mapper/Reducer**: Hadoop Streaming supports providing Python scripts as mapper and reducer files.
- **Customizing Splits**: Users can customize how map and reduce tasks split input data for processing.