Your request covers a broad range of topics in the realm of Big Data and Cloudera's ecosystem. Here's a comprehensive overview of each topic:

YARN Overview

YARN (Yet Another Resource Negotiator) is a core component of Hadoop 2.x, allowing for resource management and job scheduling. It decouples MapReduce's resource management and scheduling capabilities, enabling other data processing frameworks to run on Hadoop. YARN consists of the ResourceManager, NodeManager, and ApplicationMaster.

Running Applications on YARN

To run applications on YARN, developers typically submit their application to the ResourceManager, which then handles the allocation of resources and scheduling. The ResourceManager works with NodeManagers on each node to execute tasks. The ApplicationMaster, specific to each application, negotiates resources with the ResourceManager and works with the NodeManagers to execute and monitor tasks.

Viewing YARN Applications

Applications running on YARN can be viewed and managed through the YARN ResourceManager UI or using command-line tools. The ResourceManager UI provides information about current and completed applications, resource usage, and cluster health.

YARN Application Logs

Logs for applications running on YARN are usually stored on the NodeManagers. They can be accessed via the ResourceManager UI or through tools like `yarn logs`. These logs are crucial for debugging and monitoring application performance.

MapReduce Applications

MapReduce is a programming model for processing large datasets with a parallel, distributed algorithm on a cluster. It simplifies data processing on large clusters and is an integral part of Hadoop.

YARN Memory and CPU Settings

YARN allows for fine-tuning of memory and CPU resources for applications. Configuration settings like `yarn.nodemanager.resource.memory-mb` and `yarn.nodemanager.resource.cpu-vcores` are used to set these resources. Proper configuration ensures efficient resource utilization and application performance.

Hands-On Exercise: Running YARN Applications

A practical exercise involves submitting a job to YARN, monitoring its progress through the ResourceManager UI, and reviewing its performance and logs. This helps in understanding the end-to-end process and nuances of running applications on YARN.

Tez Overview

Apache Tez is a framework for complex data-processing tasks, built on Hadoop YARN. It enhances the MapReduce paradigm by introducing a more flexible execution engine that allows for complex Directed Acyclic Graphs (DAGs) of tasks for processing data.

Hive on Tez

Hive can use Tez as its execution engine, replacing MapReduce. Hive on Tez offers significant performance improvements, especially for complex queries that benefit from Tez's more efficient task execution and resource management.

ACID for Hive

Hive supports ACID (Atomicity, Consistency, Isolation, Durability) properties for transactions. This is crucial for ensuring data integrity, especially in scenarios with multiple concurrent writes and reads.

Spark Overview

Apache Spark is a unified analytics engine for large-scale data processing. It offers high-level APIs in Java, Scala, Python, and R, and an optimized engine that supports general computation graphs. Spark is known for its speed, ease of use, and extensive library ecosystem.

How Spark Applications Run on YARN

Spark applications can run on YARN, utilizing its resource management capabilities. When running Spark on YARN, the Spark driver can run either in the cluster on a NodeManager or outside the cluster, while executors run on NodeManagers.

Monitoring Spark Applications

Spark applications can be monitored through the Spark UI, YARN ResourceManager UI, or third-party tools. These interfaces provide insights into job execution, resource usage, and other performance metrics.

Phoenix Overview

Apache Phoenix is a SQL layer on top of Apache HBase, providing a relational database layer over HBase's NoSQL foundation. It allows for SQL-like querying over HBase data, making it more accessible for developers familiar with SQL.

Hands-On Exercise: Running Spark Applications

A hands-on exercise with Spark on YARN could involve writing a Spark application, submitting it to YARN, and then monitoring its execution using the Spark UI and ResourceManager UI. It's a valuable exercise to understand the integration of Spark with the YARN resource management system.