Apache Airflow Scheduler and YARN

September 9, 2025

Agenda

- Introduction to Apache Airflow
- Airflow Scheduler Overview
- Introduction to YARN
- Integration of Airflow and YARN
- Use Cases of Airflow and YARN
- Benefits and Challenges
- Summary and Q&A

Airflow

Introduction to Apache Airflow

What is Apache Airflow?

Apache Airflow is an open-source platform that allows users to create, schedule, and monitor workflows as directed acyclic graphs (DAGs). It is widely used for automating complex data pipelines.

Core Functionality

Airflow enables programmatic workflow authoring using Python, supports scheduling, and provides a rich user interface for monitoring task progress and managing dependencies.

Role in Workflow Orchestration

Airflow coordinates and manages the execution order of tasks across distributed systems, ensuring tasks run in the correct sequence with retry and alerting capabilities.

Automation and Scalability

Airflow automates repetitive workflows and scales efficiently with distributed executors, making it suitable for large-scale data processing and complex pipeline management.

Scheduler

Understanding the Airflow Scheduler

Role of the Scheduler

The scheduler monitors DAGs (Directed Acyclic Graphs) and triggers task execution based on predefined schedules and dependencies, ensuring workflows run smoothly and on time.

How the Scheduler Triggers Tasks

It parses DAG files, evaluates task dependencies and schedules, then submits runnable tasks to the executor for execution, enabling automated and timely task runs.

Key Features of the Scheduler

Features include dynamic DAG parsing, dependency resolution, backfill support, failure handling, and integration with various executors to optimize workflow orchestration.

Scheduler Performance and Scalability

The scheduler is designed to handle large-scale workflows by supporting parallel task scheduling, scaling horizontally, and maintaining low latency in task dispatching.

YARN

Introduction to YARN

What is YARN?

YARN (Yet Another Resource Negotiator) is a cluster management technology in Hadoop that separates resource management from job scheduling, enabling better scalability and flexibility.

Architecture Overview

YARN architecture consists of a ResourceManager, NodeManagers on each node, and an ApplicationMaster managing individual applications, orchestrating resource allocation and task execution.

Role of ResourceManager and NodeManager

ResourceManager manages global resource allocation and scheduling, while NodeManagers handle resource monitoring and task execution on individual cluster nodes.

How YARN Manages Resources

YARN dynamically allocates CPU, memory, and other resources based on application needs, optimizing cluster utilization and supporting multi-tenant workloads efficiently.

Integration

Integration of Airflow with YARN

Why Integrate Airflow with YARN

Combining Airflow's workflow orchestration with YARN's resource management optimizes big data processing, enabling dynamic task scheduling and efficient cluster utilization.

How Airflow Schedules YARN Apps

Airflow triggers and monitors YARN applications by submitting jobs to YARN’s ResourceManager, controlling execution based on task dependencies and schedules.

Communication between Airflow and YARN

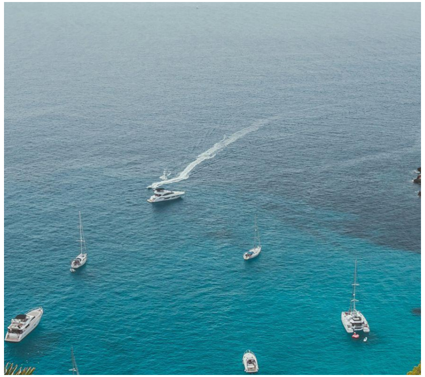
Airflow communicates with YARN via REST APIs or command-line interfaces to submit, monitor, and manage resource allocation for running applications.

Typical Workflows

Common workflows include big data ETL pipelines, machine learning training jobs, batch processing, and data analytics tasks orchestrated across Airflow and YARN.

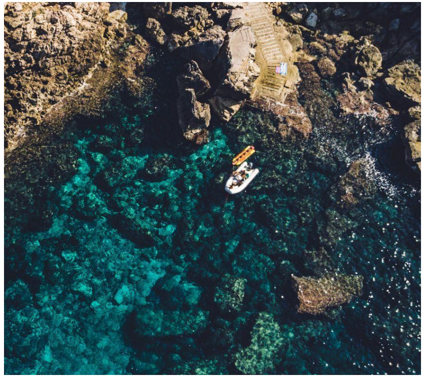
Use Cases

Use Cases of Airflow Scheduler with YARN



Big Data Processing Workflows

Airflow schedules complex big data tasks on YARN clusters, ensuring efficient resource allocation and job execution.



Machine Learning Pipelines

Automates training and deployment pipelines by coordinating YARN-managed resources for scalable machine learning workflows.

ETL Processes

Manages extract, transform, load tasks seamlessly by scheduling ETL jobs on YARN, optimizing data movement and transformation.

Batch Job Orchestration

Coordinates large-scale batch jobs efficiently across YARN clusters to maximize throughput and minimize runtime.

Benefits

Benefits of Using Airflow Scheduler with YARN

Efficient Resource Utilization

Airflow Scheduler leverages YARN's resource management to allocate computing power dynamically, reducing waste and improving cluster efficiency.

Improved Workflow Automation

Automates complex, multi-step data pipelines by coordinating task execution and resource allocation, minimizing manual oversight and errors.

Scalability and Flexibility

The integration supports scaling workflows seamlessly across large distributed systems, adapting to changing workloads without manual intervention.

Enhanced Monitoring and Management

Provides comprehensive tracking of task statuses and resource usage, allowing proactive troubleshooting and optimized performance.

Challenges

Challenges and Considerations

Complexity of Setup

Configuring Airflow with YARN involves multiple components and dependencies, requiring careful coordination and expertise to ensure seamless integration and reliable operation.

Debugging and Troubleshooting

Identifying and resolving issues across distributed components in Airflow and YARN can be difficult, requiring comprehensive logging, monitoring tools, and deep system knowledge.

Resource Contention and Management

Competing demands for cluster resources can cause delays; effective resource allocation policies and monitoring are needed to prevent bottlenecks and optimize utilization.

Performance Tuning

Optimizing scheduler intervals, task concurrency, and YARN resource parameters is critical to achieve high throughput and low latency in workflow execution.

Summary

- Apache Airflow Scheduler automates and orchestrates complex workflows efficiently.
- YARN manages cluster resources dynamically to optimize utilization.
- Integration enables Airflow to schedule and monitor YARN applications seamlessly.
- This combination supports scalable big data processing and machine learning pipelines.
- Benefits include improved automation, resource efficiency, and enhanced monitoring capabilities.

Thank you

September 9, 2025