APACHE AIRFLOW: CADRU DE GESTIONARE A FLUXURILOR DE LUCRU

Andreea BAGRIN

Departamentul Ingineria Software și Automatică, grupa TI-216, Facultatea Calculatoare Informatică și Microelectronică, Universitatea Tehnică a Moldovei, Chișinău, Republica Moldova

Autorul corespondent: Andreea Bagrin, andreea.bagrin@isa.utm.md

Coordonator științific: **Dorian SARANCIUC**, lector universitar, Departamentul ISA

Rezumat. *Apache Airflow reprezintă un framework open-source de orchestrare a fluxurilor de lucru în mediul de date, dezvoltat inițial de Airbnb și ulterior donat Apache Software Foundation. Acest articol explorează conceptele cheie ale Apache Airflow, cum ar fi Grafurile Aciclice Direcționate (DAGs), Operatorii și Schedulerul, evidențiind arhitectura sa modulară și componente esențiale precum Metadata Database și Web Server. Se discută extensibilitatea și integrarea Airflow cu ecosistemul Big Data, evidențiind utilizarea Hooks și Operatorilor personalizate. Comunitatea activă și documentația detaliată sunt prezentate ca resurse valoroase pentru adoptarea și dezvoltarea cu succes a soluțiilor bazate pe Airflow. Studii de caz ale adoptării Airflow de către companii precum Airbnb, Lyft și PayPal subliniază utilitatea și eficacitatea acestui framework în gestionarea complexă a fluxurilor de lucru.*

Cuvinte cheie: *Airflow, Python, DAGs (Directed Acyclic Graphs), Operatori, Scheduler, Task*

Introducere

Airflow a fost creat de comunitate pentru a autoriza, programa și monitoriza fluxurile de lucru în mod programatic. Acesta are o arhitectură modulară și utilizează o coadă de mesaje pentru a orhestra un număr arbitrar de lucrători. Airflow a devenit o unealtă esențială în toolkit-ul datelor și în mediile de dezvoltare și administrare a sistemelor distribuite. Sistemele distribuite reprezintă o arhitectură de calcul în care componentele software sau hardware se găsesc pe mai multe noduri interconectate în rețea și colaborează pentru a realiza o funcționalitate comună. Într-un sistem distribuit, resursele și sarcinile sunt distribuite între mai multe mașini, ceea ce permite realizarea unor operațiuni paralele și coordonarea activităților între ele. Airflow poate fi implementat în mai multe moduri, variind de la un singur proces pe laptop la o configurare distribuită pentru a sprijini chiar și cele mai mari fluxuri de lucru și este gata să se extindă la infinit. Fluxurile de lucru sunt reprezentări structurate ale unui set de activități sau operațiuni care trebuie efectuate într-o anumită ordine pentru a atinge un anumit obiectiv. Cadrul Airflow conține operatori care se pot conecta cu multe tehnologii și este extensibil pentru a se conecta cu o nouă tehnologie. Dacă fluxurile dvs. de lucru au un început și un sfârșit clar și rulează la intervale regulate, ele pot fi programate ca un DAG de Airflow. Acest articol explorează rolul esențial al Apache Airflow în orchestrarea fluxurilor de lucru, evidențiind caracteristicile sale cheie și impactul său în mediul de date.

Concepte cheie ale Apache Airflow

DAGs (Directed Acyclic Graphs) Airflow utilizează conceptul de Grafuri Aciclice Direcționate (DAGs) pentru a reprezenta fluxurile de lucru. O DAG în Airflow este un set de task-uri interconectate, unde fiecare task reprezintă o unitate de lucru independentă.

Operatorii în Airflow reprezintă execuția concretă a unui task. Există o varietate de operatori încorporați care acoperă operațiuni comune, cum ar fi execuția de SQL, transferul de fișiere, trimiterea de e-mailuri și multe altele.

Schedulerul în Airflow este responsabil pentru planificarea execuției task-urilor în funcție de dependințele definite în DAG. Acesta asigură o execuție coerentă și eficientă a fluxului de lucru [1].

Un DAG este o colecție a task-lor pe care dorim să le executăm, organizate într-un mod ce reflectă relația și dependența dintre ele. Fiecare task este o implementare a unui operator, cum ar fi un PythonOperator va executa un cod în Python și definește valori specifice pentru acel operator. În timp ce operatorii pot fi PythonOperator, BashOperator sau un chiar un operator personalizat, ei determină ce de fapt va fi executat de acel task. Reprezentarea schematică a acestor concepte ale Apache Airflow este ilustrată în Fig. 1.

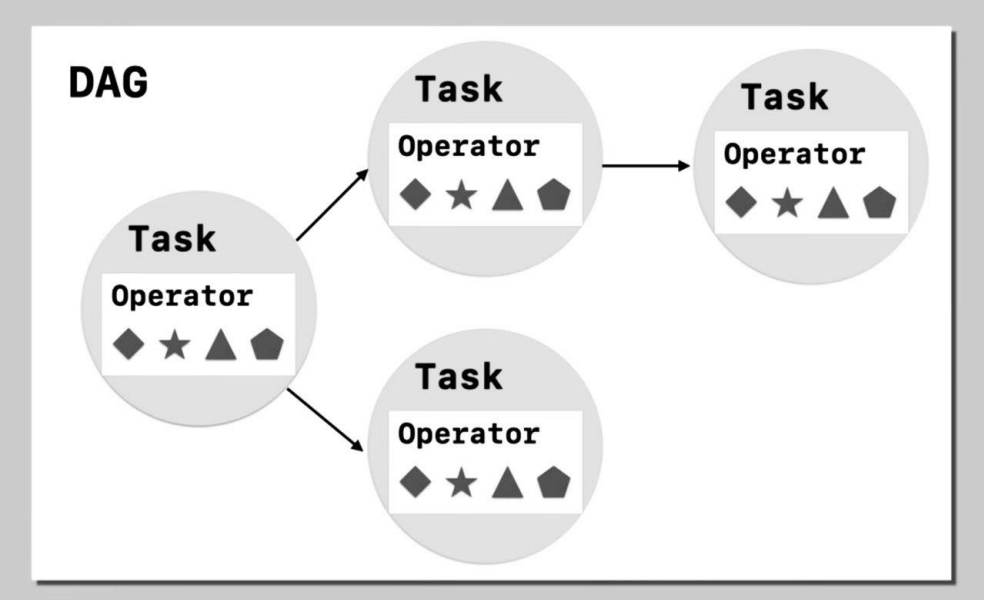

Figura 1. Reprezentarea conceptelor cheie

Caracteristici

Airflow oferă multe caracteristici utile, descrise mai jos.

Scalabil, Airflow are o arhitectură modulară și utilizează o coadă de mesaje pentru a orhestra un număr arbitrar de lucrători.

Dinamic, fluxurile de lucru Airflow sunt definite în Python, ceea ce permite generarea dinamică a fluxurilor de lucru.

Extensibil, se pot defini cu ușurință proprii operatori și extinde bibliotecile pentru a se potrivi nivelului de abstractizare care se potrivește cu mediul dvs.

Intuitiv, fluxurile de lucru Airflow sunt simple și explicite.

Integrări robuste, Airflow oferă mulți operatori plug-and-play care sunt gata să execute sarcinile dvs. pe Google Cloud Platform, Amazon Web Services, Microsoft Azure și multe alte servicii terțe [2].

Dacă preferați să codați decât să faceți clic, Airflow este instrumentul pentru dvs. Fluxurile de lucru sunt definite ca cod Python, ceea ce înseamnă:

- fluxurile de lucru pot fi stocate în controlul versiunilor, astfel încât să puteți reveni la versiunile anterioare;
- fluxurile de lucru pot fi dezvoltate de mai multe persoane simultan;
- testele pot fi scrise pentru a valida funcționalitatea;
- componentele sunt extensibile și puteți construi pe o colecție largă de componente existente.

Semantica bogată de planificare și execuție vă permite să definiți cu ușurință conducte complexe, care rulează la intervale regulate. Completarea vă permite să (re)rulați conducte pe datele istorice după ce faceți modificări logicii. Iar capacitatea de a reexecuta conducte parțiale după rezolvarea unei erori ajută la maximizarea eficienței.

Arhitectură și Componente

Metadata Database - Airflow folosește o bază de date de metadate pentru a stoca informații despre DAGs, execuții, task-uri și alte entități. Acest lucru facilitează monitorizarea și gestionarea istoricului execuțiilor.

Web Server - interfața web Airflow oferă o vedere detaliată a DAG-urilor, execuțiilor și a altor informații utile pentru dezvoltatori și administratori.

Scheduler - responsabil pentru programarea execuției sarcinilor în funcție de dependențele lor.

Executor - responsabil pentru efectuarea efectivă a task-urilor. Airflow suportă executori locali, executori distribuiți și integrări cu tehnologii precum Celery [3].

Reprezentarea componentelor și arhitecturii în Apache Airflow este demonstrată in Fig. 2, unde în partea stângă, un "Inginer de date" sau "Autor" este reprezentat, conectat la un folder "DAGs" care simbolizează crearea fluxurilor de lucru. Caseta "Interfață utilizator" indică locul în care utilizatorii pot interacționa cu sistemul, casetele "Server web" și "Programator" sunt componente centrale care gestionează și execută fluxurile de lucru. Un fișier "Airflow.cfg" este afișat, indicând unde sunt stocate setările de configurare ale sistemului. În partea dreaptă, există un "Executor" care poate fi "Local sau Secvențial", conectat la "Muncitor(i)" indicând unde sunt executate sarcinile. În partea de jos, un simbol de bază de date etichetat "Metadata DB (Postgres)" arată unde sunt stocate metadatele.

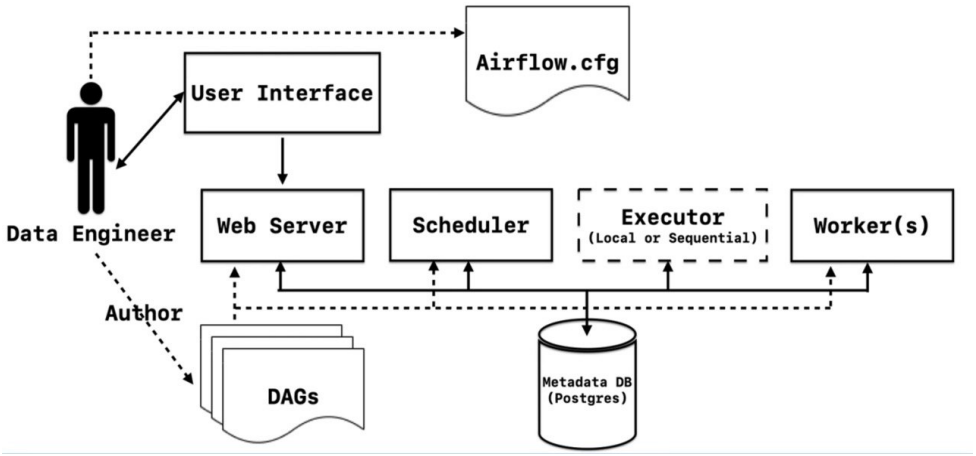

Figura 2. Reprezentarea arhitecturii și componentelor

Extensibilitate și Integrare

Hooks și Operators personalizate oferă posibilitatea dezvoltatorilor de a extinde funcționalitatea Airflow prin crearea de Hooks (conexiuni la sisteme externe) și Operatori personalizate, adaptate la necesitățile specifice ale organizației sau proiectului.

Integrare cu ecosistemul Big Data oferă conectivitate nativă cu tehnologii precum Apache Hadoop, Apache Spark și altele.

Utilizare

Airflow oferă mulți operatori plug-and-play care sunt gata să-ți execute sarcinile pe Google Cloud Platform, Amazon Web Services, Microsoft Azure și multe alte servicii terțe.

Acest lucru face ca Airflow să fie ușor de aplicat la infrastructura actuală și de extins la tehnologiile de nouă generație.

Airflow este utilizat într-o varietate de domenii, inclusiv analiza datelor, ingineria datelor, știința datelor, devops și multe altele. Acesta este utilizat de companii mari precum Airbnb, Yahoo, Intel și Lyft [5].

Concluzie

În ansamblu, Apache Airflow se remarcă ca un instrument esențial pentru orchestrarea eficientă a fluxurilor de lucru în domeniul datelor. Conceptele cheie precum Grafurile Aciclice Direcționate (DAGs), Operatorii și Schedulerul oferă o abordare modulară și flexibilă pentru definirea și gestionarea task-urilor. Arhitectura sa, cu componente precum Metadata Database și Web Server, asigură o monitorizare detaliată și gestionare eficientă a execuțiilor.

Cu extensibilitatea sa, dezvoltatorii pot personaliza funcționalitățile prin Hooks și Operatori personalizate, adaptând Airflow la nevoile specifice ale proiectelor. Integrarea sa cu ecosistemul Big Data îl face potrivit pentru mediile complexe și distribuite. Comunitatea activă și documentația detaliată consolidează atractivitatea Apache Airflow, furnizând suport solid și resurse pentru utilizatori. Studiile de caz ale adoptării de către companii precum Airbnb, Lyft și PayPal ilustrează succesul și versatilitatea acestui framework în contexte variate.

Apache Airflow se dovedește a fi o alegere puternică pentru automatizarea și gestionarea fluxurilor de lucru în mediul de date, oferind un cadru solid pentru dezvoltatori și administratori în căutarea unei soluții scalabile și flexibile

Referințe

[1] „Apache Airflow Documentation” [Online]. Available: https://airflow.apache.org/docs/
[2] „Apache Airflow – Tutorials” [Online]. Available: https://airflow.apache.org/docs/apache-airflow/stable/tutorial/fundamentals.html
[3] „Apache Airflow - Architecture Overview” [Online]. Available: https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/overview.html
[4] „GitHub Repository for Apache Airflow” [Online]. Available: https://github.com/apache/airflow
[5] „Apache Airflow: Use Cases, Architecture, and Best Practices” [Online]. Available: https://www.run.ai/guides/machine-learning-operations/apache-airflow