APACHE FLINK

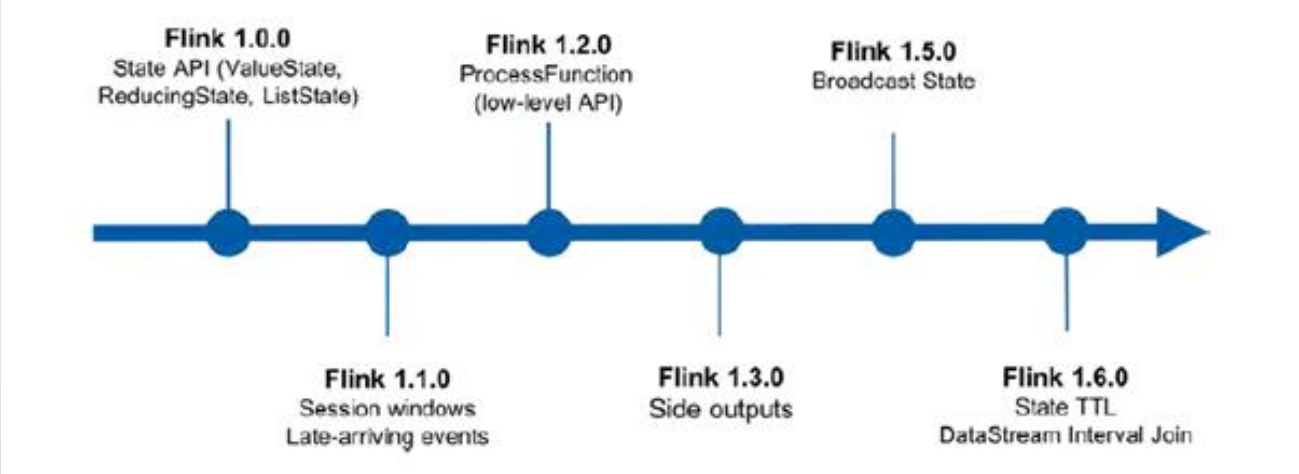
ALEKSANDAR IGNJATIJEVIĆ E2-12/2021

TEME

- Istorija Apache Flink-a
- Arhitekture Apache Flink-a
- Pojmovi bitni za Apache Flink aplikacije
- Slučajevi korišćenja Apache Flink alata na primerima *event-driven*, *data-analytics* i *data pipeline* aplikacija

ISTORIJA APACHE FLINK-A

- Apache Flink predstavlja *open-source* platformu napravljenu od strane Apache Flink zajednice nastao 2011. godine i od tad, zahvaljujući neprestanom radu *open-source* zajednice, izašlo je nekoliko verzija, a trenutna stabilana verzija je 1.14.2.
- Na slici je prikazan razvoj, zaključno sa 2019. godinom.



OSNOVE ARHITEKTURE APACHE FLINK-A

- Apache Flink je radni okvir (eng. *framework*) i alat za distribuiranu obradu podataka sa ograničenim i neograničcenim tokovima podataka (eng. *data streams*). Napravljen je sa idejom da radi u svim poznatim klaster okruženjima i da obavlja izračunavanja *in-memory* brzinom, nad bilo kom nivou. Takođe, Flink koristi istu arhitekturu koja podržava i *batch* i *stream* obradu podataka.

PODIZANJE I SKALIRANJE

- Apache Flink-a dobro funkcioniše sa popularnim cluster resource managers, poput Kubernetes-a, Apache Mesos-a i YARN-a, ali je isto tako moguće podesiti ga da radi na pojedinačnom klasteru.
- Pri samom podizanju arhitekture, Flink će sam prepoznati koliko resursa je potrebno za uspešno izvršavanje aplikacije, na osnovu toga kako je konfigurisana aplikacija, sa aspekta paralelizacije
- Skaliranje sa Apache Flink-om je veoma zahvalno, i otprilike neograničeno. Razlog za to je sama sposobnost Flink-a da aplikacije paralelizuje u veliki broj zadataka koji su distribuirani i koji se konkurentno izvršavaju u klasterima
- Flink ima sposobnost da održava veoma veliki application state. Uz pomoć asinhronog i inkrementalnog checkpoint algoritma (eng. asynchronous and incremental checkpointing algorithm), Apache Flink osigurava minimalni uticaj kašnjenja obrade

APACHE FLINK APLIKACIJE

- Bitni pojmovi za Apache Flink aplikacije:
  - State
  - Time
  - Stream
  - Layered APIs: ProcessFunctions, DataStream API i SQL & Table API

SLUČAJEVI KORIŠĆENJA

- Nekoliko karakterističnih primera upotrebe Apache Flinka, poput event-driven aplikacija, data analytics aplikacija i data pipeline aplikacija

EVENT DRIVEN APLIKACIJE

- Ovakve aplikacije koriste određene događaje iz nekih tokova koji aktiviraju izračunavanja ili promene stanja
- Event-driven aplikacije se baziraju na stateful stream processing aplikacijama
- Podaci i izračunavanja skladište na istom mestu, što olakšava pristup.
- Otpornost na greške se postiže upisivanjem checkpoint-a u odvojeno perzistentno skladište
- Ograničenja kod ovakvih aplikacija se svode na to koliko dobro tokovi mogu da rukuju vremenom i stanjima, a kako Flink uspešno rukje ovim stvarima, predstavlja veoma dobar izbor.
- Primeri: sistemi za detekciju anomalija, prevara, sistemi za monitoring poslovnih procesa ili društvene mreže.

DATA ANALYTICS APLIKACIJE

- Ovakve aplikacije izvlače informacije iz sirovih podataka. Obično se ovo radi batch obradom nad ograničenim skupom podataka.
- Ali kako uključiti i nove podatke koji stižu, potrebno je dodati te podatke u već obrađeni set ili ponovo pokretati aplikaciju.
- Flink ovaj problem rešava tako što omogućava korisnicima da koriste streaming obradu prilikom analize podataka.
- Tipične aplikacije ovog tipa su: monitoring kvaliteta, analiza update-ova i eksperimentalna evaluacija u mobilnim aplikacijama, kao i ad-hoc analiza podataka u konzumerskim tehnologijama

DATA PIPELINE APLIKACIJE

- Extract-transform-load (ETL) je čest pristup transformacijama i premeštanju podatak između različitih sistema za skladištenje.
- Često su ovakve akcije pokrenute za periodično premeštanje podatak iz transakcione u analitičku bazu podataka. Data pipeline-ovi rade poslove slične ETL-u.
- Razlika je u tome što su njihove akcije kontinualne, a ne periodične, pa se samim time koriste za izvore podatak koji konstantno izbacuju nove podatke.
- Primer ovkve aplikacije može biti aplikacija koja će konstantno pisati podatke koji stižu sa event stream-a u bazu. Slika 8 prikazuje razliku između ETL-a i data pipeline-a

HVALA NA PAŽNJI