De juiste streaming engine for the job
Onderzoeksvoorstel Bachelorproef

Piet Laureyns[1]

Samenvatting
Big Data Streaming is een proces waarbij data snel wordt verwerkt om er realtime inzichten uit te halen, dit levert een competitief voordeel tegenover de batch-geörienteerde data verwerking. Maar er zijn echter veel streaming tools waardoor het voor een bedrijf moeilijk is om de juiste tool voor hun situatie te kiezen.
We gaan dus een aantal experimenten uitvoeren op 3 streaming tools(Reactive Streaming, Kafka Streams, Apache Spark) om zo een overzicht te kunnen geven waarop te zien is welke streaming tool het beste is voor een bepaalde situatie. We verwachten dat dit overzicht meer inzicht geeft over de 3 streaming tools en hun sterktes & zwaktes.

Big Data Streaming wordt steeds belangrijker en populairder, het is dus belangrijk dat bedrijven een goed overzicht hebben van verschillende streaming tools.

Sleutelwoorden
Onderzoeksdomein. Databanken en big data — Data Streaming

Co-promotor
Sam Waegeman[2] (XTi NV)

Contact: [1] piet.laureyns.w1870@student.hogent.be; [2] sam.waegeman@xt-i.com;

Inhoudsopgave

1. Introductie

Voor vele bedrijven is de batch-geörienteerde architectuur om Big Data te verwerken gewoon te traag. Moderne ‘fast data’ architecturen zijn geëvolueerd naar stream-gebaseerde architecturen.

Bij streaming wordt de data meteen verwerkt terwijl die verzameld wordt. Dit leidt tot een competitief voordeel voor deze bedrijven. Er zijn echter vele stream processing tools, en de vraag is dan uiteraard welke tool het beste gebruikt wordt bij bepaalde situaties.

2. Literatuurstudie

Big data streaming is een proces waarbij big data snel wordt verwerkt om er realtime inzichten uit te halen. Er zijn momenteel heel veel streaming tools die elk hun eigen sterktes en zwaktes hebben.

De 3 tools die we gaan onderzoeken zijn:

- Reactive Streams
- Kafka Streams
- Apache Spark (Spark Streaming)

Reactive Streams is een initiatief om een standaard te bieden voor asynchrone streamverwerking. Reactive streams is sinds JDK 9 beschikbaar in Java.

Kafka Streams is een client library voor het bouwen van applicaties en microservices, waar de invoer- en uitvoergegevens worden opgeslagen in Kafka-clusters. Het combineert de eenvoud van standaard Java- en Scala-applicaties aan de cliëntzijde met de voordelen van Kafka’s server-side clustertechnologie.

Spark Streaming brengt de taalgeïntegreerde API van Apache Spark in data streaming, waardoor men streamingopdrachten op dezelfde manier als batch-jobs kan ontwikkelen. Het ondersteunt Java, Scala en Python.

3. Methodologie

In deze bachelorproef willen we een aantal criteria onderzoeken die belangrijk zijn voor het kiezen van specifieke streaming technologieën. De experimenten zullen uitgevoerd worden op 3 streaming tools: Reactive Streams, Kafka Streams

en Apache Spark, die allen op een ander niveau toelaten om te werken met een continue stream van data.

De belangrijkste criteria die we zullen bekijken om de correcte streaming tool te kiezen zijn:

- **Latency:** welke snelheid van verwerking is nodig?
- **Volume:** hoeveel data moet verwerkt worden?
- **Processing:** welke soort data moet verwerkt worden?
- **Integratie met andere tools:** welke en hoe?

Naast een overzicht van de 3 vermelde technologieën is het ook de bedoeling om concreet werkende implementaties te ontwikkelen. We gaan een tool trachten te ontwikkelen die een stream van inzichten geeft in cryptomunten(price changes, trade volume, markt inzichten & visualisatie, enz...). Deze tool zal ook detecteren wanneer een bepaalde cryptomunt een significante prijs en/of volume aanpassing meemaakt, om zo een potentieel interessant inkoop moment te signaliseren.

4. Verwachte resultaten

We verwachten een overzicht te kunnen geven die de 3 streaming tools(Reactive Streams, Kafka Streams en Apache Spark) met elkaar vergelijkt op meerdere voorbeeld-situaties.
Reactive streams is de recenste tool die beschikbaar is in Java sedert JDK 9, we verwachten dus dat deze tool het beste zal werken met andere java technologieën.
Er zijn al in andere werken vergelijkingen gedaan tussen Kafka Streams en Spark Streaming, hieruit werd geconcludeerd dat Kafka Streams beter is in een ”Kafka > Kafka” context terwijl Spark Streaming beter zou zijn voor een ”Kafka > Database” context. We verwachten dezelfde conclusies te kunnen trekken.
Met deze resultaten zullen we ook een tool trachten te ontwikkelen die inzichten geeft in cryptomunten. Het succes van deze tool zal afhangen van onze resultaten.

5. Verwachte conclusies

Uit deze resultaten verwachten we te kunnen concluderen dat iedere streaming tool elke zijn eigen sterktes & zwaktes heeft. Deze resultaten zullen dan door bedrijven kunnen geraadpleegd worden bij het kiezen van de juiste streaming tool voor hun specifieke situatie.