Data Lake - Design, Projeto e Integração Versão 2.0

Vantagens do Apache Kafka

O Apache Kafka possui muitos pontos fortes quando o assunto é processamento de mensagens. A seguir estão algumas das vantagens que o Kafka possui, tornando-o ideal para nossa implementação do Data Lake:

- Alto rendimento: o Kafka é capaz de lidar com dados de alta velocidade e alto volume usando hardware modesto. É capaz de suportar o processamento de milhares de mensagens por segundo.

- Baixa latência: o Kafka é capaz de lidar com mensagens de latência muito baixa em intervalo de milissegundos, exigida pela maioria dos casos de uso para aplicações real-time.

- Tolerante a falhas: a capacidade inerente do Kafka de ser resistente a falhas de nó/máquina dentro de um cluster.

- Durabilidade: os dados/mensagens são persistentes no disco, tornando-os duráveis e as mensagens também são replicadas para que nunca sejam perdidas.

- Escalabilidade: o Kafka pode ser expandido on-the-fly, sem incorrer em tempo de inatividade, adicionando novos nós. A manipulação de mensagens dentro do cluster Kafka é totalmente transparente e elas são perfeitas.

- Distribuído: suporta inerentemente arquitetura distribuída, tornando-o escalável usando recursos como replicação e particionamento. Recursos do Message Broker.

- Alta simultaneidade: capaz de lidar com milhares de mensagens por segundo e também em condições de baixa latência com alta taxa de transferência. Kafka permite a leitura e escrita de mensagens em alta simultaneidade.

- Por padrão persistente: por padrão, as mensagens são persistentes, tornando-as duráveis e confiáveis.

- Consumidor: Kafka pode ser integrado com uma variedade de consumidores. Cada cliente tem uma capacidade diferente de lidar com essas mensagens

provenientes do Kafka e, por causa de sua capacidade inerente de persistência, pode se comportar ou agir de maneira diferente, de acordo com o consumidor com o qual se integra. Ele também se integra bem com uma variedade de consumidores escritos em várias linguagens de programação.

- Capacidade de manipulação de lotes (funcionalidade semelhante a ETL): como o Kafka persiste mensagens, ele também pode ser empregado para casos de uso semelhantes ao processamento em lotes e também pode fazer o trabalho de um ETL tradicional.

- Capaz de lidar com uma variedade de casos de uso comumente necessários para um Data Lake, ou seja, agregação de registros, rastreamento de atividades da Web e assim por diante.

- Projetado para funcionar em hardware de commodity. Para as POCs (Provas de Conceito), tudo bem usar um hardware simples, mas para uma empresa com um ambiente de produção, não é recomendável selecionar um hardware comum, embora ele funcione muito bem.

- Ajuda a gerenciar o pipeline de dados em tempo real. Este é um dos principais motivos de nossa escolha, pois precisamos encontrar uma peça de tecnologia para lidar com mensagens em tempo real.

- Apache Kafka é open source e tem uma grande comunidade de seguidores. Além disso, existem muitas empresas que estão prontas para fornecer suporte comercial, o que pode ser um aspecto importante para as grandes empresas devido à sua importância.

- Kafka funciona bem com o Apache Spark, nossa solução para processamento de dados em tempo real.

- Para transmitir mensagens aos consumidores, ele usa recursos do sistema operacional e, por isso, pode funcionar muito bem. Ele também usa muitos recursos do sistema operacional para fazer muitas coisas com eficiência, e esse é definitivamente um ponto positivo.

Mas o Apache Kafka também possui problemas e limitações, que veremos na próxima aula.

Referência:

http://kafka.apache.org/documentation/