Apache Kafka

Introdução ao ecossistema

Ermogenes Palacio

Apache Kafka é...

...uma plataforma de *streaming* de eventos distribuídos de código aberto usada por milhares de empresas para:

- *pipelines* de dados de alto desempenho
- análise de *streaming*
- integração de dados
- aplicativos de missão crítica

Entidades vs. eventos

- **Entidade**: algo que existe e é identificável
  - *existência*
- **Evento**: algo que aconteceu
  - *ocorrência*

Bancos de dados relacionais são organizados em torno do conceito de entidades, mas não escalam bem no tratamento de eventos.

Uma estrutura de dados que permite escalabilidade no tratamento de eventos é o *log* (um registro persistente de eventos).

Streaming de eventos

Permite um fluxo contínuo e interpretável dos dados para que as informações estejam no lugar certo, na hora certa.

- Captura de dados em tempo real de **diferentes origens**
- **Armazenamento durável** dos fluxos de eventos
- **Processar** os fluxos de eventos em tempo real e retrospectivamente
- Direcionar os fluxos de eventos para **diferentes destinos**

Casos de uso comuns

- Processar pagamentos e transações financeiras em tempo real
- Rastrear e monitorar carros, caminhões, frotas e remessas em tempo real
- Capturar e analisar continuamente os dados de sensores de dispositivos IoT
- Coletar e reagir imediatamente às interações (varejo, viagens, redes sociais, etc.)
- Monitorar pacientes em cuidados hospitalares e prever mudanças nas condições para garantir o tratamento oportuno em emergências
- Conectar, armazenar e disponibilizar dados produzidos por diferentes setores de uma organização
- Servir como base para integração de sistemas e plataformas de dados

Plataforma de *streaming* de eventos

- Funcionalidades:
  - Publicar (gravar) eventos no fluxo de eventos
  - Persistir o fluxo de eventos com a durabilidade desejada
  - Processar em tempo real e retrospectivamente (eventos passados)
  - Inscrever-se para receber (ler) eventos do fluxo de eventos

Escalável, tolerante a falhas, distribuído, seguro, *free/open source*, implantável em diversas plataformas (*on premises* ou em nuvem, *bare-metal*, *VMs*, *containers*, SaaS).

Componentes embutidos

- *Apache ZooKeeper*: coordenação
- *Apache Kafka*: mensageria e persistência, nos *brokers*
  - *Admin API*: configuração e inspeção do serviço
  - *Producer API*: entrada de dados
  - *Consumer API*: saída de dados
  - *Kafka Streams API*: transformação de dados
  - *Kafka Connect API*: criação e utilização de componentes reusáveis para casos de uso comuns de produção e consumo

Apache ZooKeeper

- Componente requerido
- Realiza a coordenação entre os *brokers*
- *Cluster* dedicado com um número ímpar de servidores (1 para desenvolvimento, 3 ou 5 para produção)

APACHE ZooKeeper™

Site oficial

Brokers (instâncias do Apache Kafka)

- Principal componente
- Entrada, persistência e saída dos registros
- Acesso binário, via TCP
  - APIs: Java, .NET, Python, Go, C++, ...
- *Cluster* de 1 até centenas de servidores

Estrutura de dados

- Evento, mensagem ou registro
  - Chave
  - Valor

São tratados como arranjos de *bytes* (ou seja, binários) e não são desempacotados ou interpretados pelo Kafka.

As APIs possuem suporte para tipos complexos usando JSON e Apache Avro.

Tópicos

Armazenas coleções de eventos.

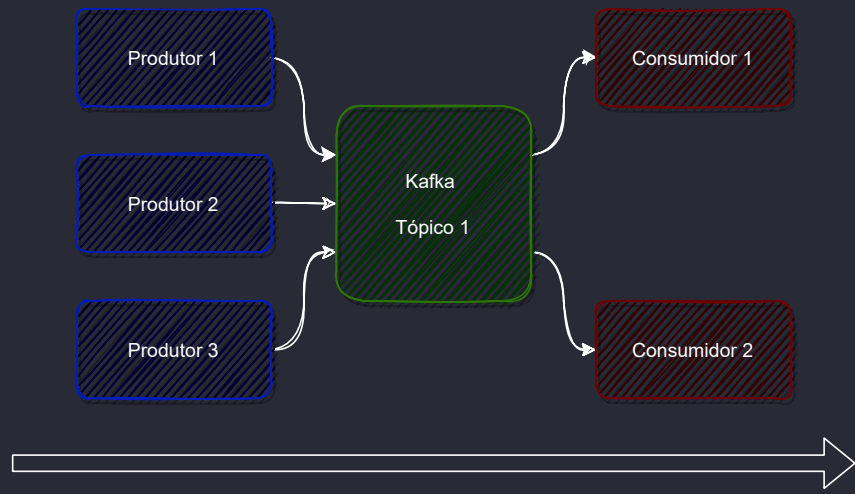
- multi-produtor e multi-consumidor
- fila persistente, com desempenho constante O(1) em todas as operações
- retenção configurável por tempo ou tamanho
- materializados em arquivos físicos, chamados de partições

Partições, *offset* e replicação

- Tópicos são em partições, distribuídas entre os *brokers*
  - dados (.log) e índices (.index)
  - podem ser replicados em mais de um *broker*
- Registros com a mesma chave são armazenados sempre na mesma partição, garantindo o sequenciamento
- Registros sem chave são balanceados *round-robin*
- Cada registro recebe um número sequencial na partição chamado *offset*

Produtores e consumidores

- **Produtores** são aplicações cliente que publicam eventos em tópicos, enviando dados a um *broker* do *cluster* através da *Producer API*.
- **Consumidores** são aplicações que se inscrevem em tópicos e recebem os seus eventos dos *brokers* do *cluster*, utilizando a *Consumer API*.
    - Podem ser agrupados em *clusters* de consumidores, garantindo a unicidade na leitura



Produtores e consumidores não são acoplados.

Obrigado!

Ermogenes Palacio
github.com/ermogenes

Referências

https://kafka.apache.org/

https://kafka.apache.org/powered-by

https://kafka.apache.org/documentation/

https://zookeeper.apache.org/

https://zookeeper.apache.org/doc/current/zookeeperOver.html

https://github.com/ermogenes/estudos-kafka