Aula 4

Big Data

Prof. Luis Henrique Alves Lourenço

1

Conversa Inicial

2

- Processamento de fluxos de dados
  - Arquitetura Lambda
  - Kafka, Flume, Storm, Flink

3

Processamento de fluxos de dados

4

Processamento de fluxos de dados distribuídos

- Ferramentas analíticas de baixa latência

5

Arquitetura Lambda

- Marz e Warren (2015)
- Processamento em lote
- Processamento de fluxos de dados
- *Batch layer*
- *Serving layer*
- *Speed layer*

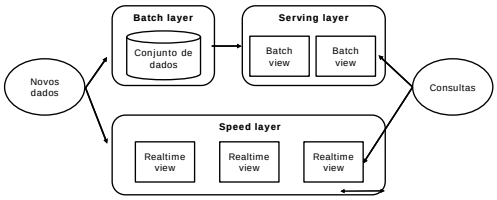
6

Propriedades de sistemas big data

- **Robustez e tolerância a falhas**
- **Escalabilidade**
- **Generalização**
- **Consultas ad hoc**
- **Manutenção mínima**
- ***Debuggability***

7

Arquitetura Lambda



8

Kafka

9

- **Plataforma de processamento de fluxos de dados**
- **Fundação Apache**
- **Alta capacidade e baixa latência**
- **Fila de mensagens**

10

Características centrais

- **Publicação e inscrição**
- **Armazenamento durável e tolerante a falhas**
- **Processamento em tempo real**

11

Cluster Kafka

- **Eventos**
- **Tópicos**

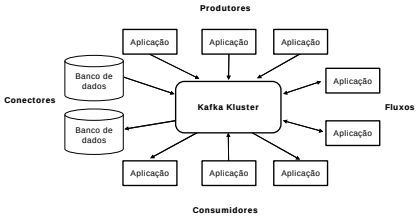
12

Kafka como um...

- **Sistema de mensagens**
- **Sistema de armazenamento**
- **Sistema de processamento de fluxos**

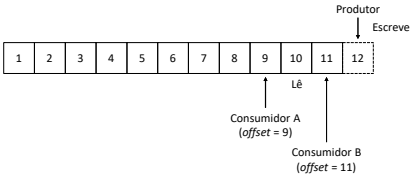
13

Arquitetura Kafka



14

Tópicos e logs



15

Flume

16

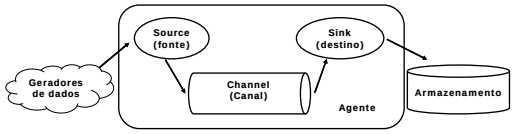
- **Projeto Apache**
- **Hadoop**
- **Log**

17

- **Distribuído, confiável e de alta disponibilidade**
- **Coletar, agregar e mover grandes quantidades de dados**

18

Arquitetura Flume



19

Componentes Flume

- Eventos
- *Source*
- *Channel*
- *Sink*
- *Runners*
- Agente
- Provedor de configuração
- Cliente

20

Exemplo de configuração

```
# exemplo.conf: Configuração de um nó Flume

# Nomeia os componentes deste agente
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Descreve e configura a source
a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 44444

# Descreve o sink
a1.sinks.k1.type = logger

# Configura Channel que bufferiza events em memoria
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# Conecta a source e o sink ao channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

21

Exemplo Flume

```
$ bin/flume-ng agent --conf conf --conf-file example.conf --name a1 -
Dflume.root.logger=INFO,console
```

```
$ telnet localhost 44444
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Hello world! <ENTER>
OK
```

22

```
12/06/19 15:32:19 INFO source.NetcatSource: Source starting
12/06/19 15:32:19 INFO source.NetcatSource: Created
serverSocket:sun.nio.ch.ServerSocketChannelImpl[/127.0.0.1:44444]
12/06/19 15:32:34 INFO sink.LoggerSink: Event: { headers:{} body: 48
65 6C 6C 6F 20 77 6F 72 6C 64 21 0D          Hello world!. }
```

23

Storm

24

- **Projeto Apache**
- **Sistema de computação distribuída em tempo real**
- **Processamento de fluxo de dados ilimitado**

25

- **Alto desempenho e baixa latência**
- **Escalabilidade horizontal**
- **Balanceamento de carga**
- **Tolerância a falhas**

26

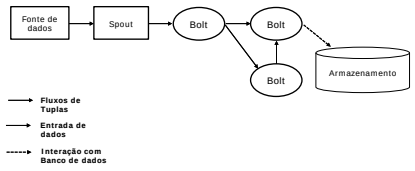
- **Mestre-trabalhador**
- **Nimbus**
- **Supervisores**

27

- ***Spout***
- ***Bolt***
- **Topologia**

28

Topologia Storm



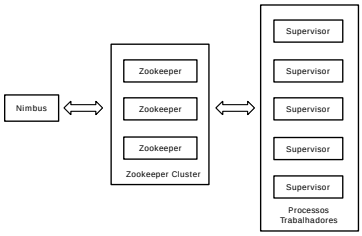
29

Agrupamento de fluxos

- ***Shuffle grouping***
- ***Fields grouping***
- ***Partial key grouping***
- ***All grouping***
- ***Global grouping***
- ***None grouping***
- ***Direct grouping***
- ***Local or shuffle grouping***

30

Arquitetura Storm



31

Interfaces para construção de topologias

- ***Trident***
- ***Stream* API**
- ***Storm* SQL**
- ***Flux***

32

Flink

33

- **Projeto Apache**
- ***Framework* e motor de processamento distribuído de fluxos**

34

- **Alto desempenho**
- **Baixa latência**
- **Tolerância a falhas**
- **Escalabilidade horizontal**

35

Principais conceitos

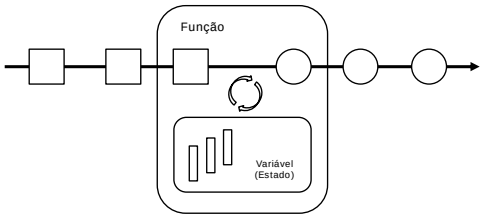
- **Fluxos ilimitados**
- **Fluxos limitados**

36

- **Paralelização em milhares de tarefas**
- **Integração com sistemas de gerenciamento de recursos: YARN, Mesos, Kubernetes**
- **REST**
- **Fluxos, estados e tempo**

37

Estados



38

APIs em camadas

- ***Process Functions***
- ***DataStream API***
- ***SQL e Table API***

39

40