Jeferson, Jonatas, Marcos

Spark

- O Apache Spark é um middleware de propósito geral projetado para ser rápido e acessível.
    - **Rápido** porque o Spark tem a habilidade utilizar muito mais a memória principal em suas computações, utilizando menos o disco que middlewares como o Hadoop MapReduce;
    - Acessível pois fornece APIs simples para Python, Java, Scala e SQL.
        - O Spark é compatível com, por exemplo, Hadoop e Cassandra.

Spark

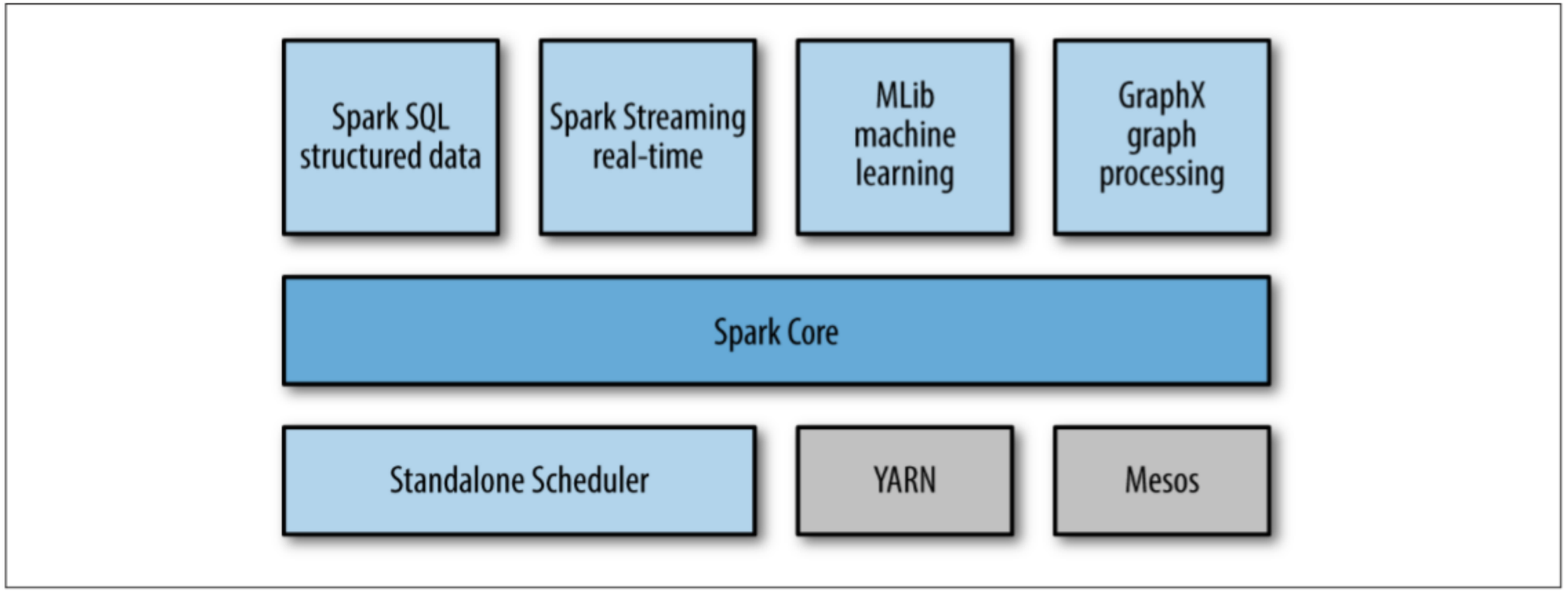


Figure 1-1. The Spark stack

Spark

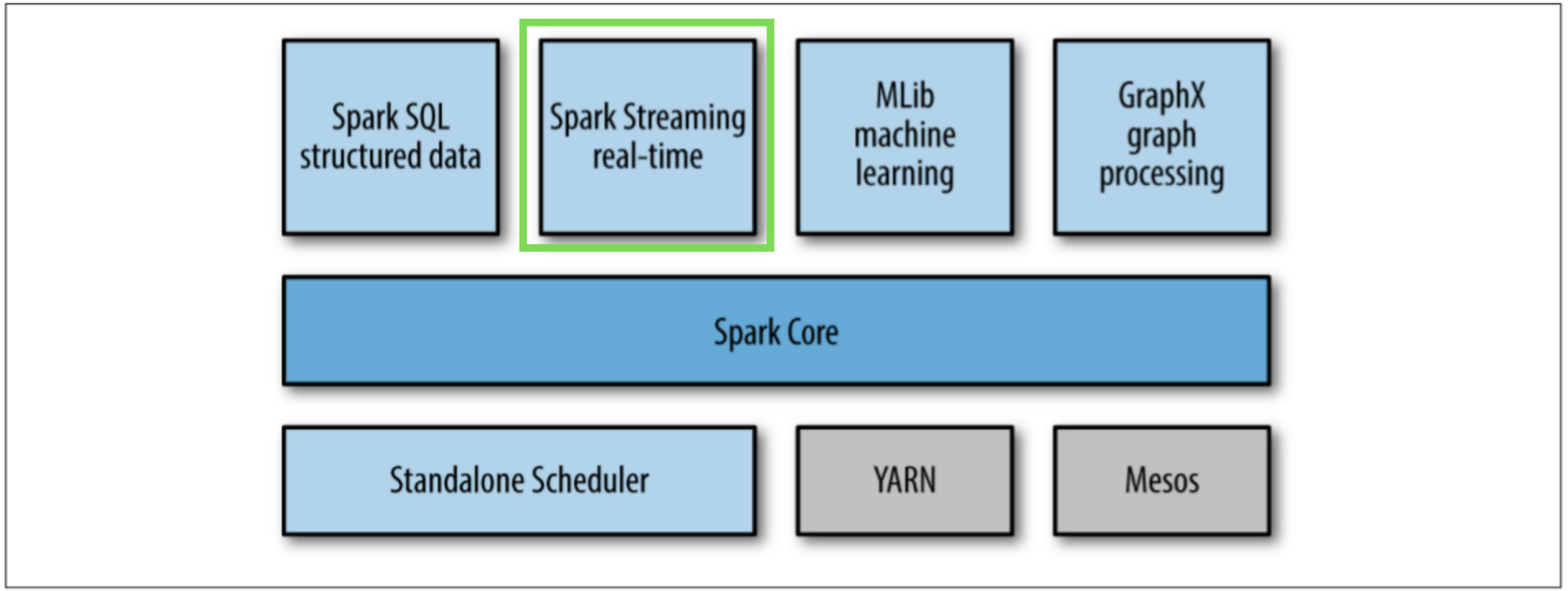
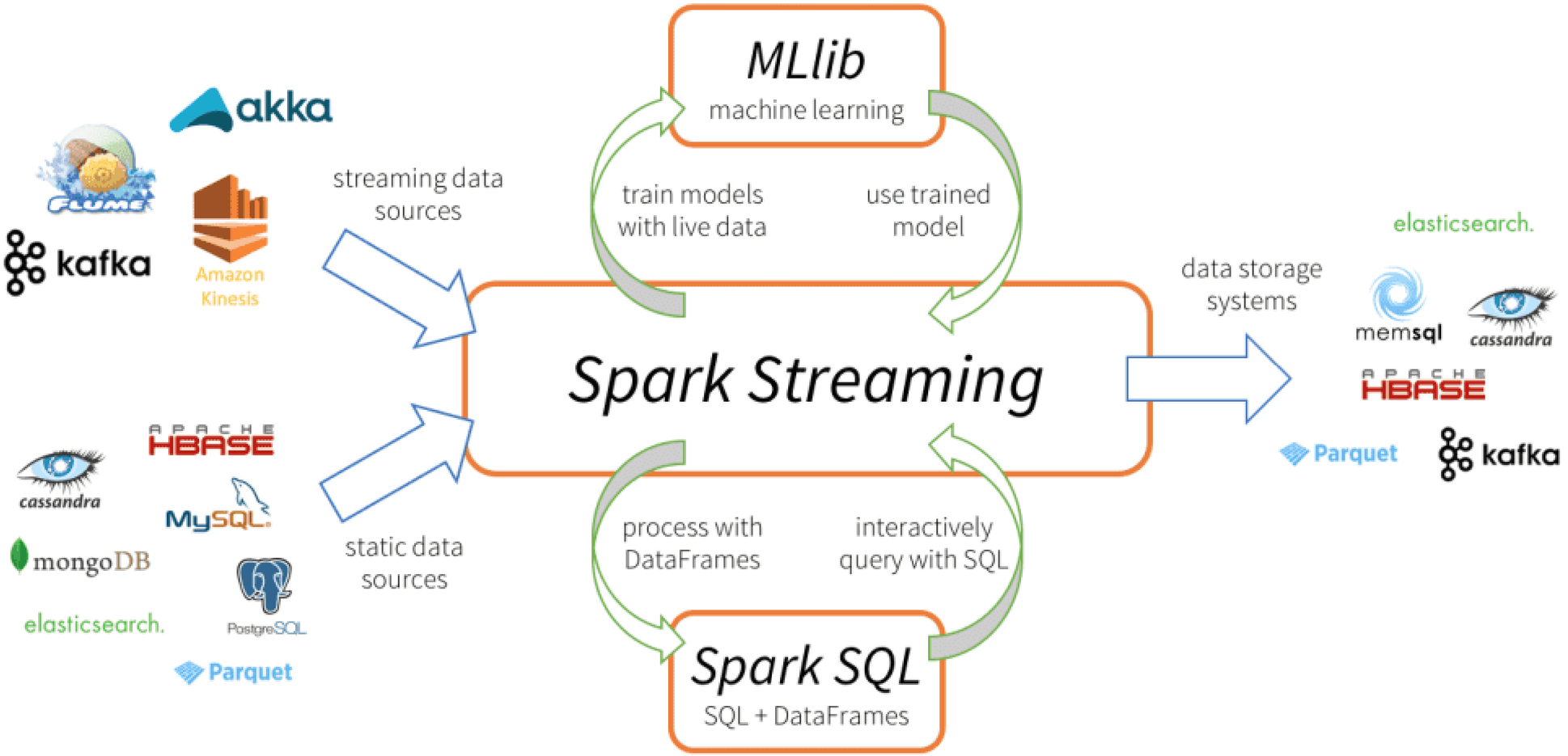


Figure 1-1. The Spark stack

Spark Streaming

- O Spark Streaming é uma extensão do Spark que permite o processamento de **streaming** de dados em tempo real.
    - Estes dados pode vir de diversas fontes como Kafka ou soquetes TCP, e podem ser processados usando algoritmos expressos com funções da API do Spark, como *map*, *reduce*, *join* e etc.
    - Os dados processados podem ser enviados para sistemas de arquivos, bancos de dados e etc.

Spark Streaming



Para que serve o Spark Streaming?

- Trabalha com dados que precisam ser analisados em **tempo real**. (*"Low latency processing applications"*)
    - Trending-topics em redes sociais
    - Modelagem dos visitantes de um site
    - Monitoramento de logs
    - *Near real time recomendations* (case da Netflix - uma das maiores aplicações)

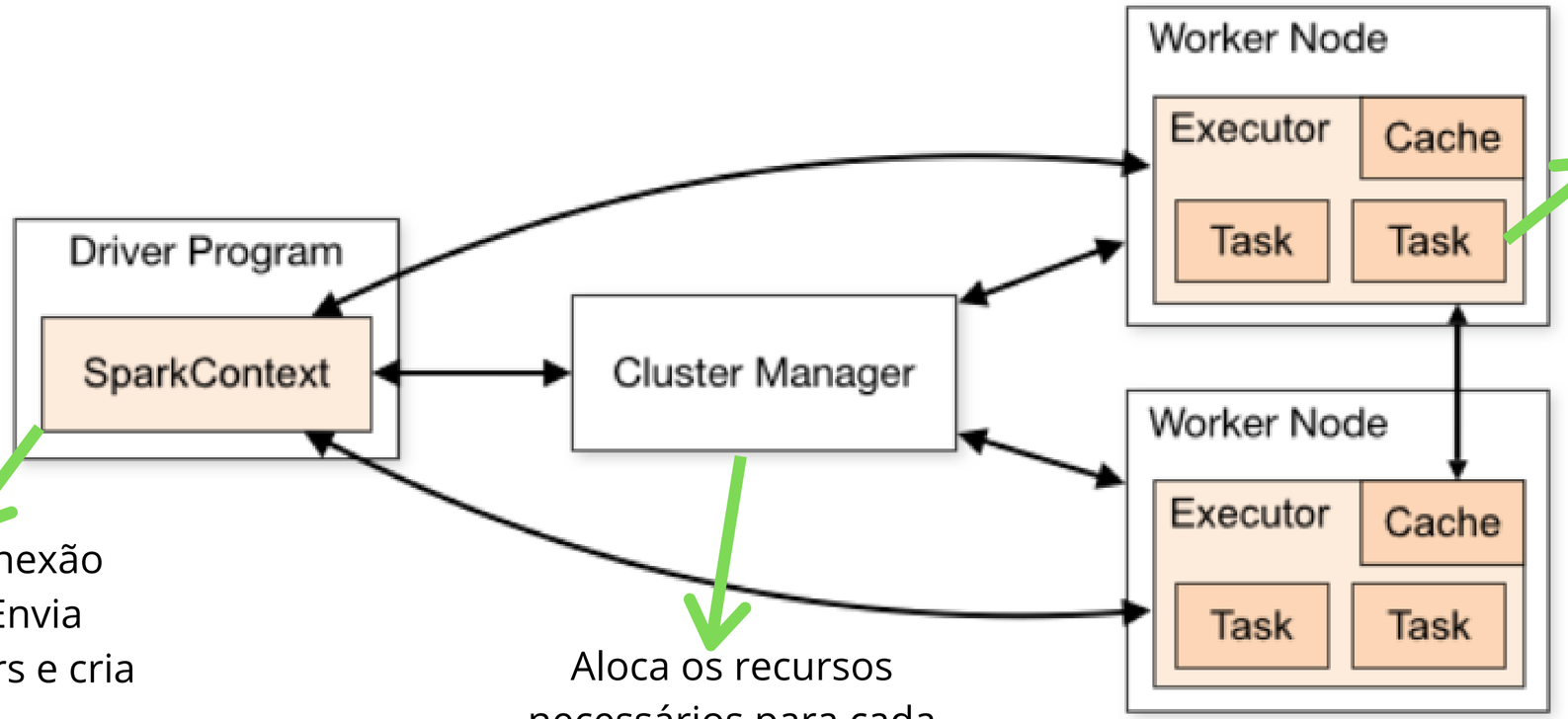
Para que não serve?

- Trabalhar com processamento em *batch* e que não haja necessidade de dividir este processamento em *mini-batches*

- Como o próprio paper já alerta:
  - *Intencionalmente, não direcionamos aplicativos com necessidades de latência* ***abaixo de algumas centenas de milissegundos****, como trading de alta frequência.*

  - O Spark Streaming trabalha na escala de 0.5-2 segundos de latência

Arquitetura - Nível core

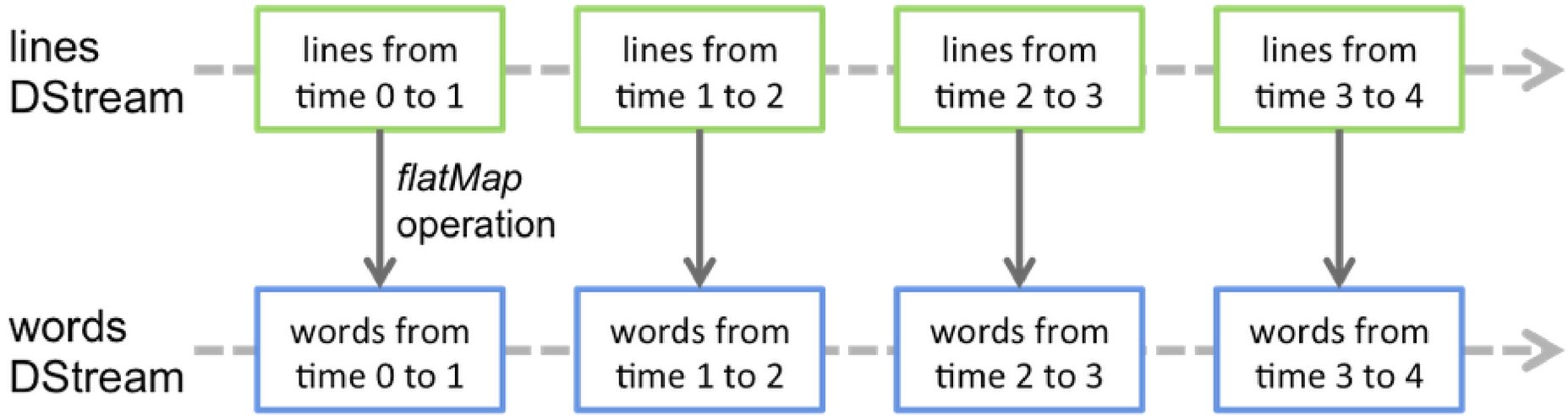


Pega o dado e executa transformações em cima dele.

Representa a conexão com o cluster. Envia tarefas aos workers e cria RDDs

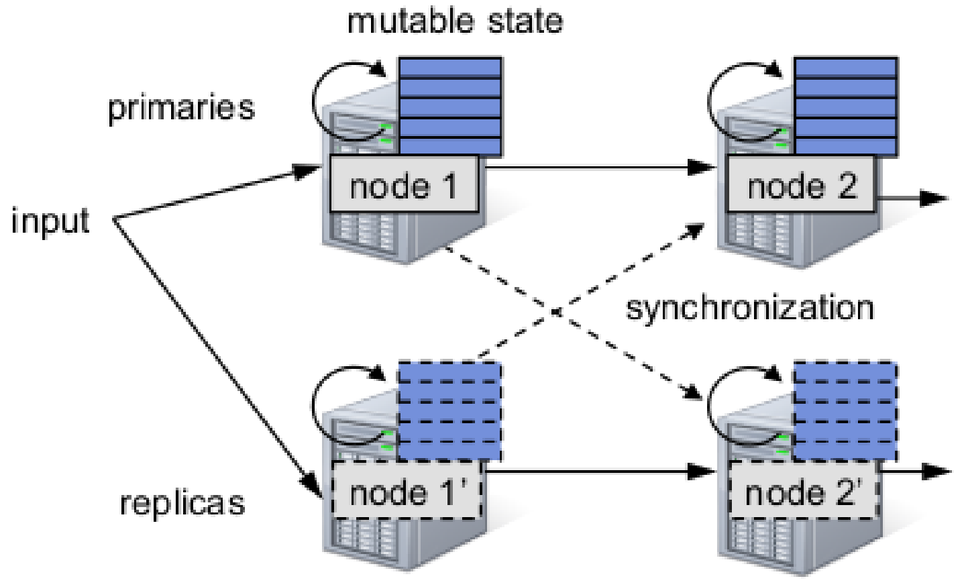
Aloca os recursos necessários para cada aplicação (Yarn, Mesos, Standalone)

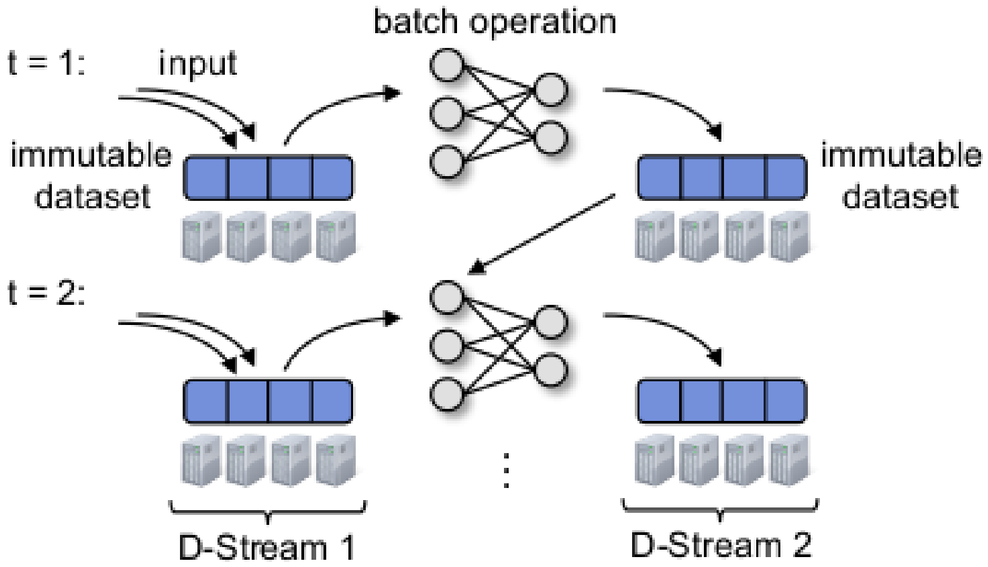
Arquitetura - Nível core



- *Discretized Stream* ou DStream é a abstração básica fornecida pelo Spark Streaming.
  - Ela representa um fluxo contínuo de dados. Internamente, um DStream é representado por uma série contínua de RDDs, que é a abstração do Spark Core de um conjunto de dados distribuído e imutável.

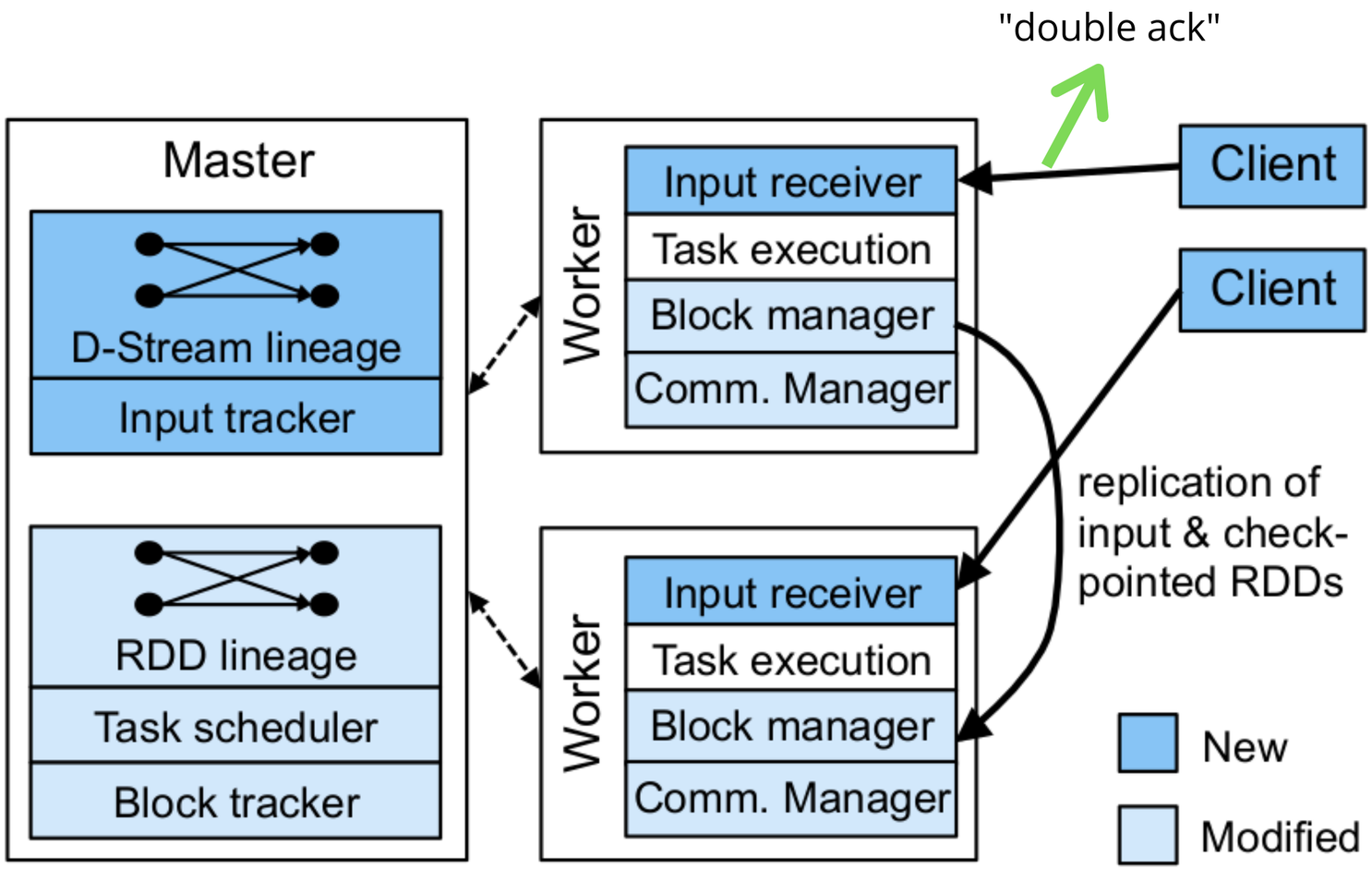
Arquitetura - nível Streaming





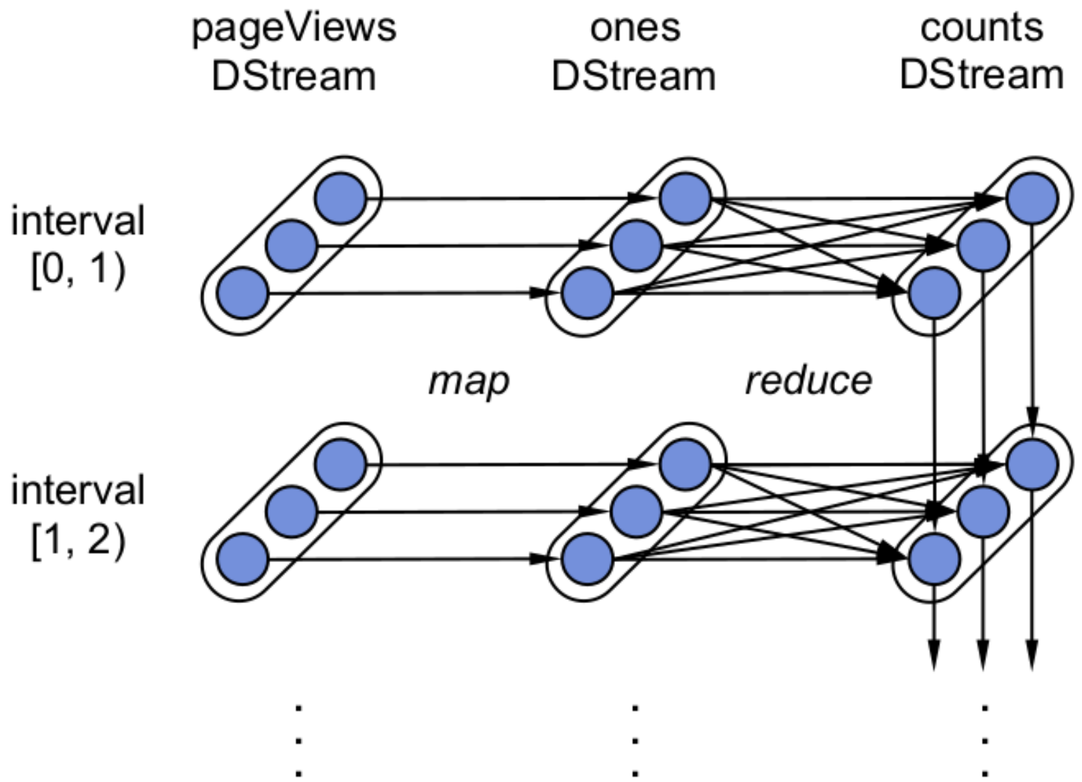
- Os sistemas de Streaming antigos eram baseados em um modelo de operação contínua, que tinha latência baixa pois extraia os dados da fonte continuamente *(long-lived tasks)*. No entanto, este modelo torna cara a recuperação de falhas (replicação e sincronização constante) e não lida bem com *stragglers (*nós lentos).

Arquitetura - nível Streaming



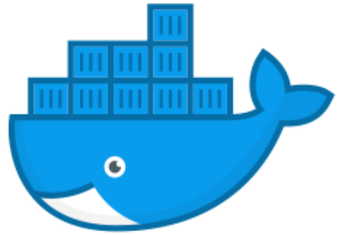
Exemplo Simples

```
pageViews = readStream("http://...", "1s")
ones = pageViews.map(event => (event.url, 1))
counts = ones.runningReduce((a, b) => a + b)
```



Instalação

Google Cloud

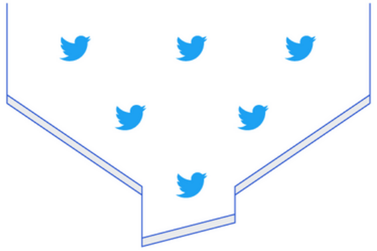


docker

kubernetes

amazon
EMR

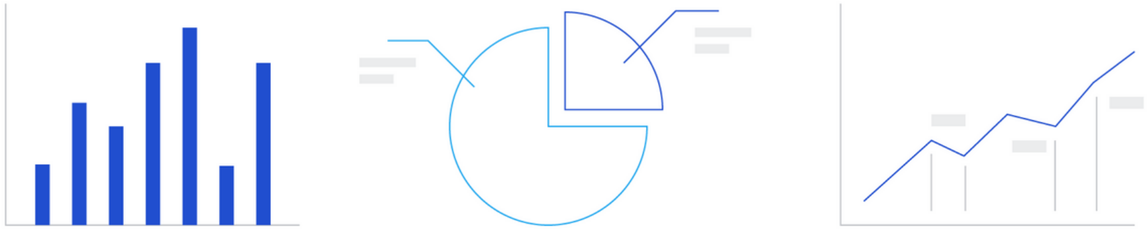
Aplicação



API do Twitter

Conexão com Spark Streaming: Envia uma amostragem de todos os tweets a cada 2s

Aplica filtros nos tweets extraindo informações sobre hashtags e trendings



Exibe dashboards interativos sobre a movimentação do twitter em tempo real