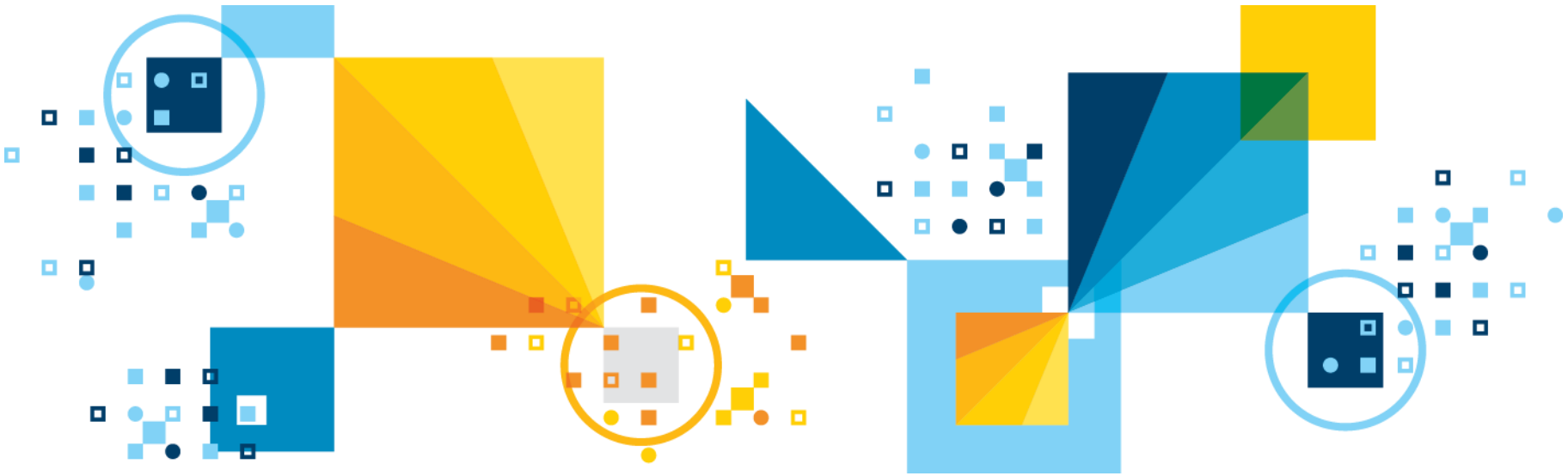
Marcos Haraoka Morais
marcoshm@br.ibm.com

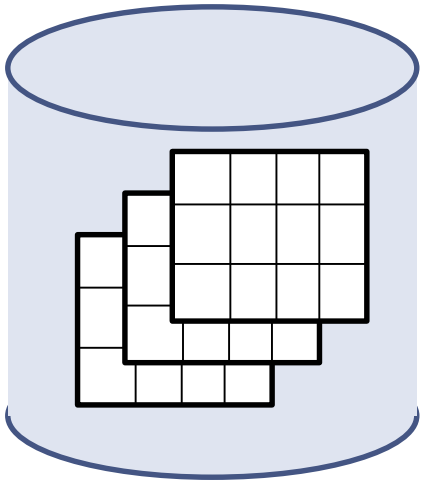
Como Transformar Dados em Insights com



© 2015 IBM Corporation

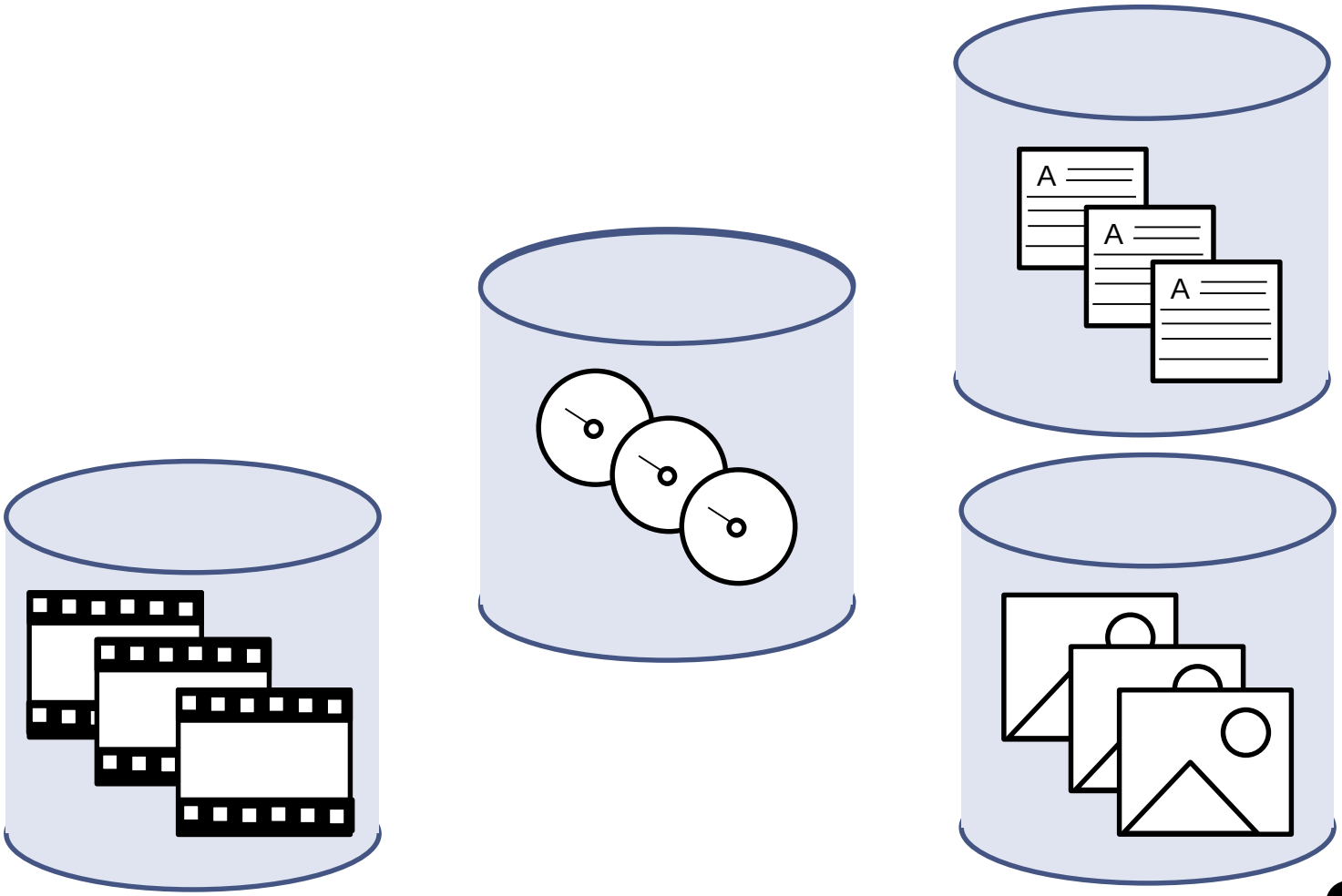
Por que Hadoop?

RDBMS



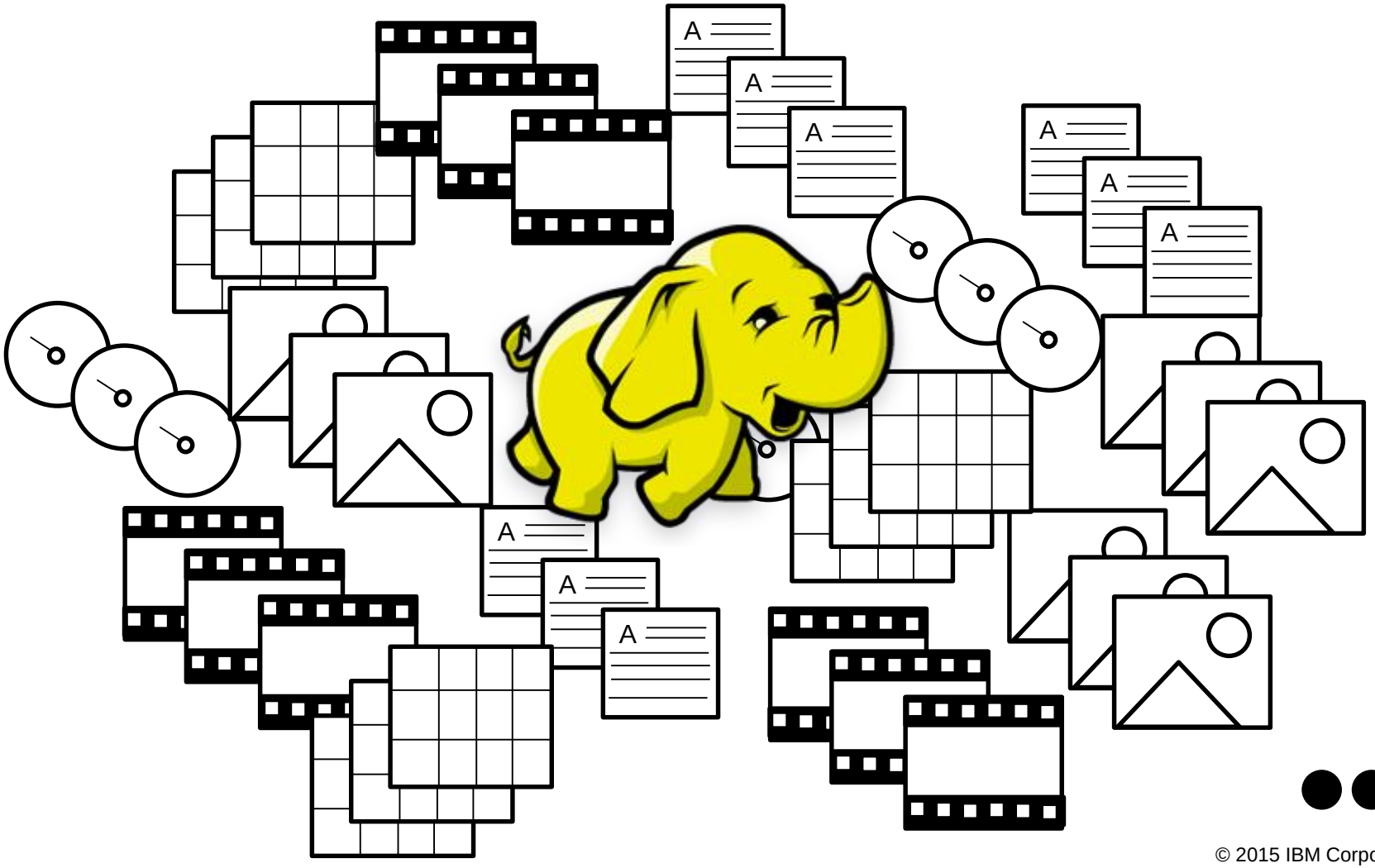
© 2015 IBM Corporation

Por que Hadoop?



© 2015 IBM Corporation

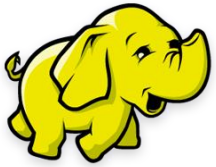
Por que Hadoop?



© 2015 IBM Corporation

O que é Hadoop?

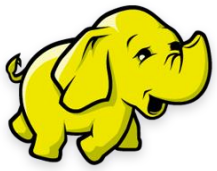
- Projeto de software open-source que permite processamento distribuído de grande conjuntos de dados.
- Componentes principais:
  - MapReduce
  - HDFS (Hadoop Distributed File System)
- Outros projetos Apache:
  - Hive (SQL)
  - Storm (Streaming)
  - Mahout (Machine Learning)
  - Giraph (Grafos)



© 2015 IBM Corporation

Vantagens do Hadoop

Escala ilimitada	• Múltiplas fontes de dados • Múltiplas aplicações • Múltiplos usuários
Plataforma Empresarial	• Confiabilidade • Resiliência • Segurança
Vasta gama de formato de dados	• Arquivos • Semi-estruturados • Banco de dados



© 2015 IBM Corporation

Desafios do Hadoop MapReduce

- Profundo conhecimento Java
- Poucas abstrações disponíveis para analistas

Facilidade no desenvolvimento

Performance In-memory

- Se[illegible] framework In-memory
- Tarefas de aplicativo são gravadas no disco a cada ciclo

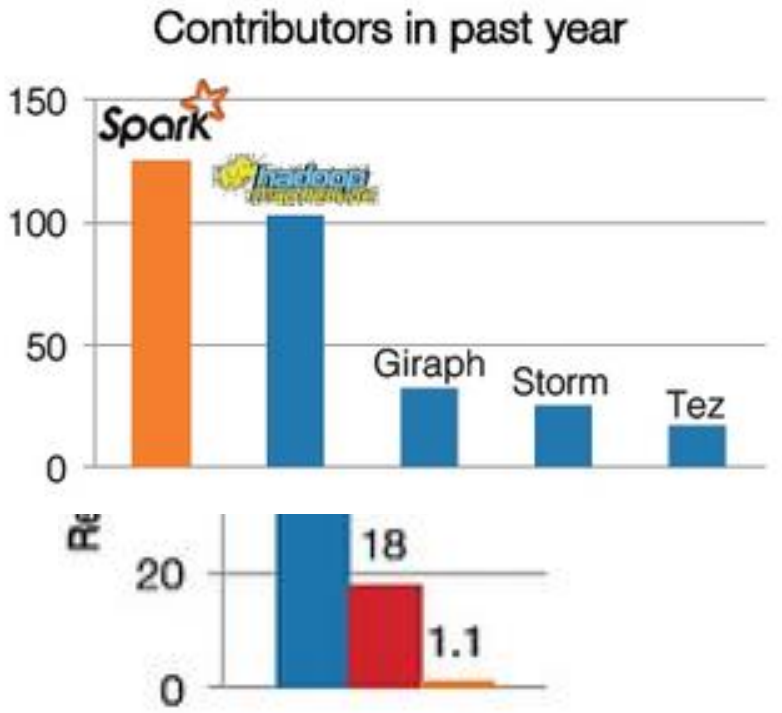
- Adequado apenas para tarefas em batch
- Modelo de processamento rígido

Combina fluxos de trabalho

© 2015 IBM Corporation

O que é Spark?

- Apache Spark é um sistema de computação em cluster rápido, de propósito geral de fácil utilização para processamento de dados em larga escala
  - Mais rápido do que o Hadoop
  - Três formas de funcionamento:
    - Standalone
    - Em conjunto com o Hadoop
    - Cloud
  - Projeto mais ativo na comunidade Apache



© 2015 IBM Corporation

Vantagens do Spark

- APIs simples
- Python, Scala, Java

Facilidade no desenvolvimento

Performance In-memory

- Resilient Distributed Datasets
- Processamento unificado

- Batch
- Interativo
- Algoritmos iterativos
- Micro-batch

Fluxos de trabalho combinados

© 2015 IBM Corporation

Facilidade no desenvolvimento

Código Java MapReduce:

```
public class LineLengthMapper
    extends Mapper<LongWritable,Text,IntWritable,IntWritable> {
  @Override
  protected void map(LongWritable lineNumber, Text line, Context context)
      throws IOException, InterruptedException {
    context.write(new IntWritable(line.getLength()), new IntWritable(1));
  }
}
public class LineLengthReducer
    extends Reducer<IntWritable,IntWritable,IntWritable,IntWritable> {
  @Override
  protected void reduce(IntWritable length, Iterable<IntWritable> counts,Context context)
      throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable count : counts) {
      sum += count.get();
    }
    context.write(length, new IntWritable(sum));
  }
}
```


© 2015 IBM Corporation

Facilidade no desenvolvimento

Código Scala Spark:

```
lines.map(line => (line.length, 1))
val lengthCounts = lines.map(line => (line.length, 1)).reduceByKey(_ + _)
```

© 2015 IBM Corporation

Vantagens do Spark

- APIs simples
- Python, Scala, Java

Facilidade no desenvolvimento

Performance In-memory

- Resilient Distributed Datasets
- Processamento unificado

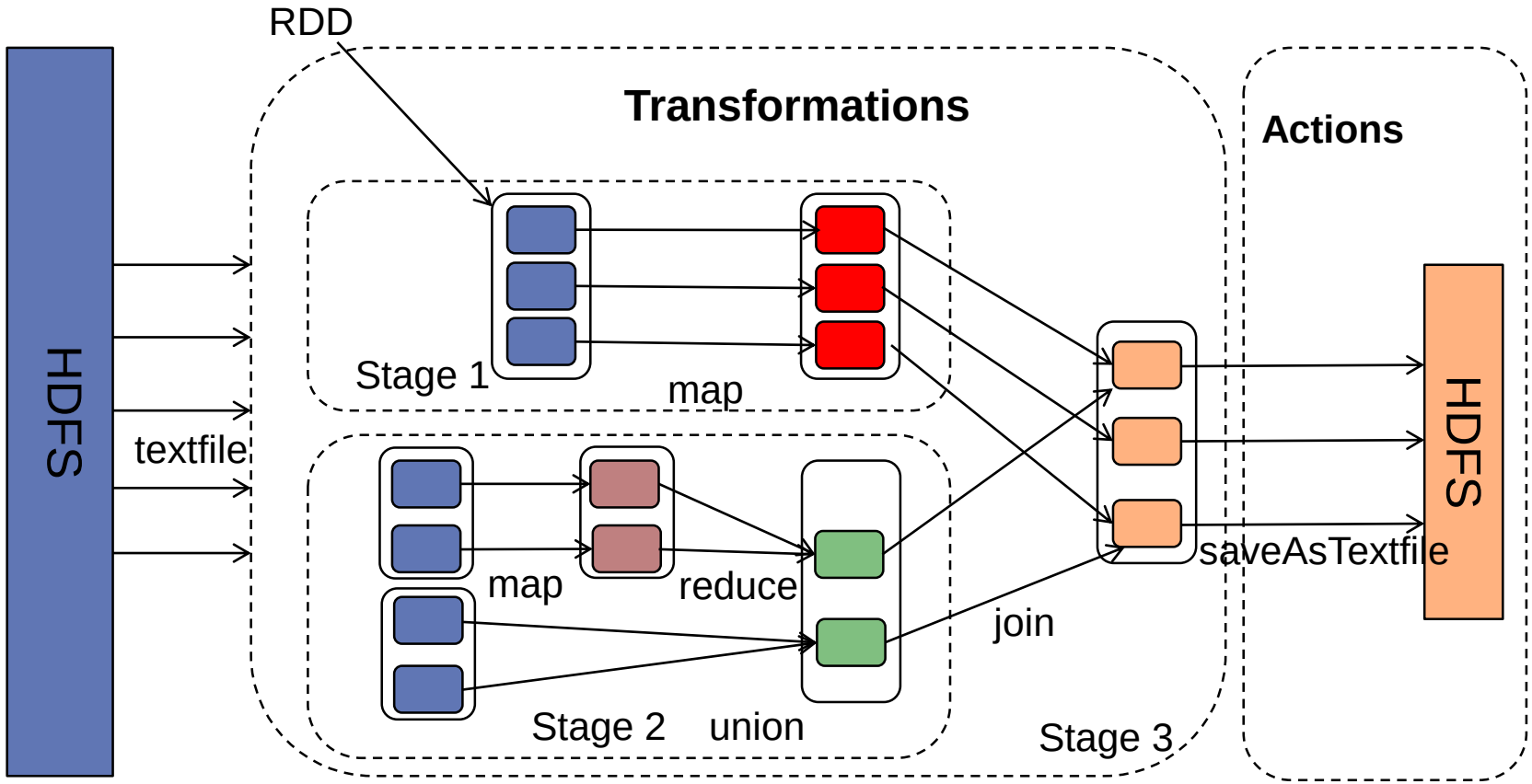
- Batch
- Interativo
- Algoritmos iterativos
- Micro-batch

Fluxos de trabalho combinados

© 2015 IBM Corporation

Performance In-memory

Resilient Distributed Dataset (RDD)



Spark

© 2015 IBM Corporation

Performance In-memory

	Hadoop	Spark	Comparação
Quantidade de dados	102.5 TB	100 TB	
Tempo	72 mins	23 mins	3.1 X <
# Nós	2100	206	10.2 X <
# Cores	50400	6592	7.6 X <
# Reducers	10,000	29,000	2.9 X >
Taxa	1.42 TB/min	4.27 TB/min	3.0 X >
Taxa/nó	0.67 GB/min	20.7 GB/min	30.9 X >
Ambiente	Data center dedicado	Cloud	On-prem/Cloud

Daytona GraySort contest 2014

© 2015 IBM Corporation

Vantagens do Spark

- APIs simples
- Python, Scala, Java

Facilidade no desenvolvimento

Performance In-memory

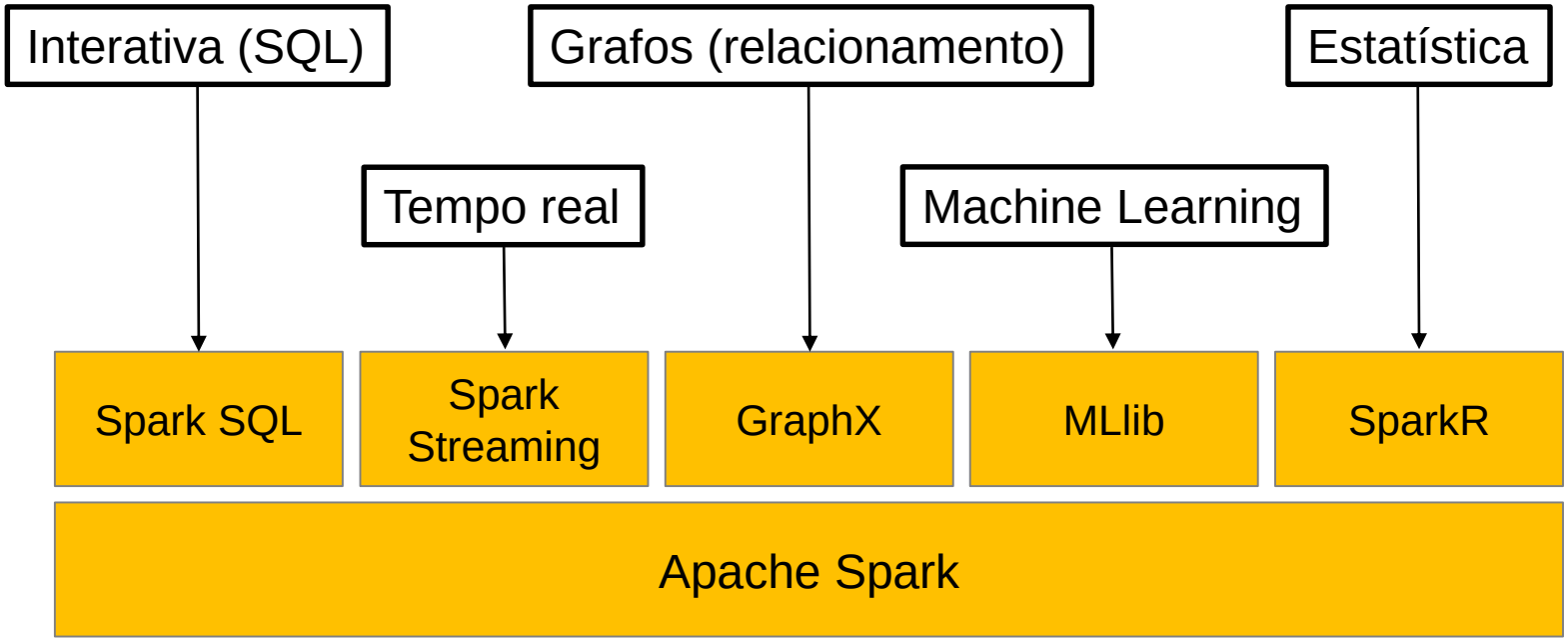
- Resilient Distributed Datasets
- Processamento unificado

- Batch
- Interativo
- Algoritmos iterativos
- Micro-batch

Fluxos de trabalho combinados

© 2015 IBM Corporation

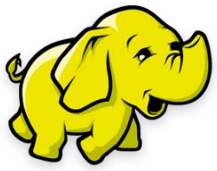
Fluxos de trabalho combinados - Bibliotecas Spark



© 2015 IBM Corporation

Flexibilidade do Spark sobre uma plataforma Hadoop estável

Escala ilimitada	Facilidade no desenvolvimento
Performance In-memory	Plataforma Empresarial
Vasta gama de formato de dados	Fluxos de trabalho combinados



© 2015 IBM Corporation

Casos de Uso

Utilizado para o entendimento dos tipos de produto que seus usuários estão vendendo com o objetivo de selecionar lojas para uma possível parceria de negócio. Optou pelo Spark após seu data warehouse ter problemas de time out rodando consultas na base.

Utilizado para a criação de um mapa de calor para visualização de dados importantes como a velocidade média dos taxis, onde os piores congestionamentos acontecem em NY, auxiliando na melhoria do tráfego nessas áreas problemáticas e aumentando a eficiência das companhias de taxi

© 2015 IBM Corporation

Casos de Uso

Ajudando na personalização das páginas do domínio Yahoo!, mostrando o que mais interessa ao usuário, baseado em seus hábitos.

Utiliza o Spark no sistema de recomendações para os usuários oferecendo conselhos obtidos do processamento de milhões de reviews de viajantes

Utiliza o Spark Streaming para selecionar e otimizar dinamicamente as fontes, enquanto o vídeo está sendo reproduzido, para maximizar a qualidade.

© 2015 IBM Corporation

"Big Data will be standard: everyone will have it."

Matei Zaharia, criador do Apache Spark

© 2015 IBM Corporation