

# BIG DATA



LABDATA



FUNDAÇÃO  
INSTITUTO DE  
ADMINISTRAÇÃO



# **Disciplina: Introdução ao Big Data**

## **Tema da Aula: Spark Core**

### **Coordenação:**

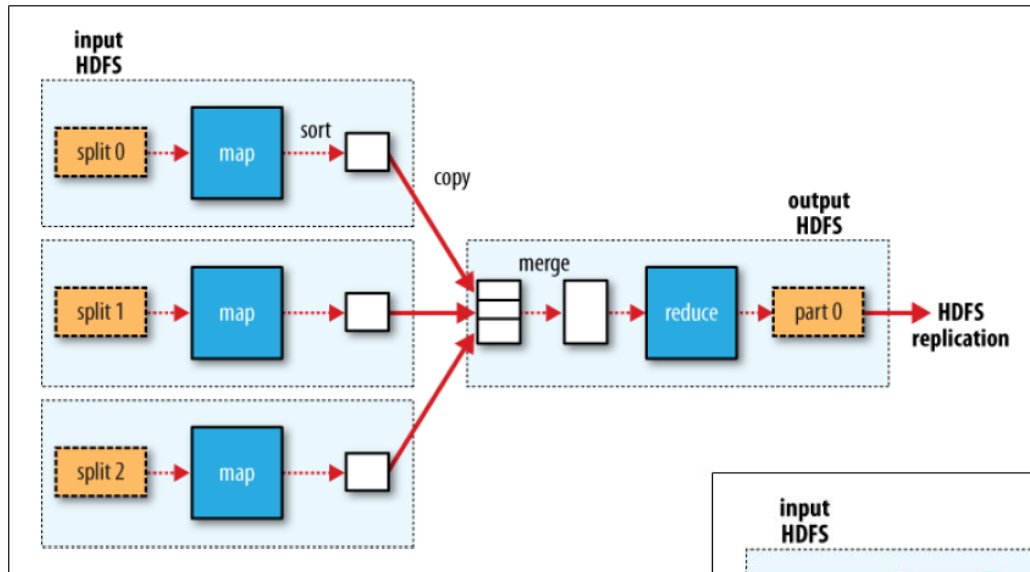
Prof. Dr. Adolpho Walter  
Pimazzi Canton

Profa. Dra. Alessandra de  
Ávila Montini

**Prof. Samuel Otero Schmidt**

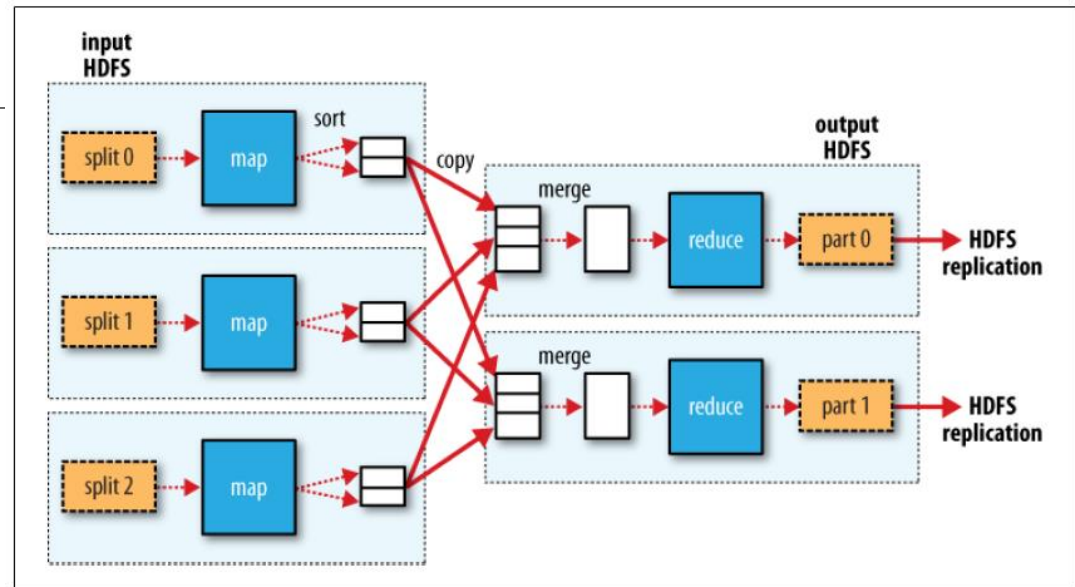
- **Processamento  
MapReduce e Sizing**
- **Extras: Perfis de Profissionais  
de Big Data**

# Fluxo do Processamento MapReduce



Único Reducer

Dois reducers



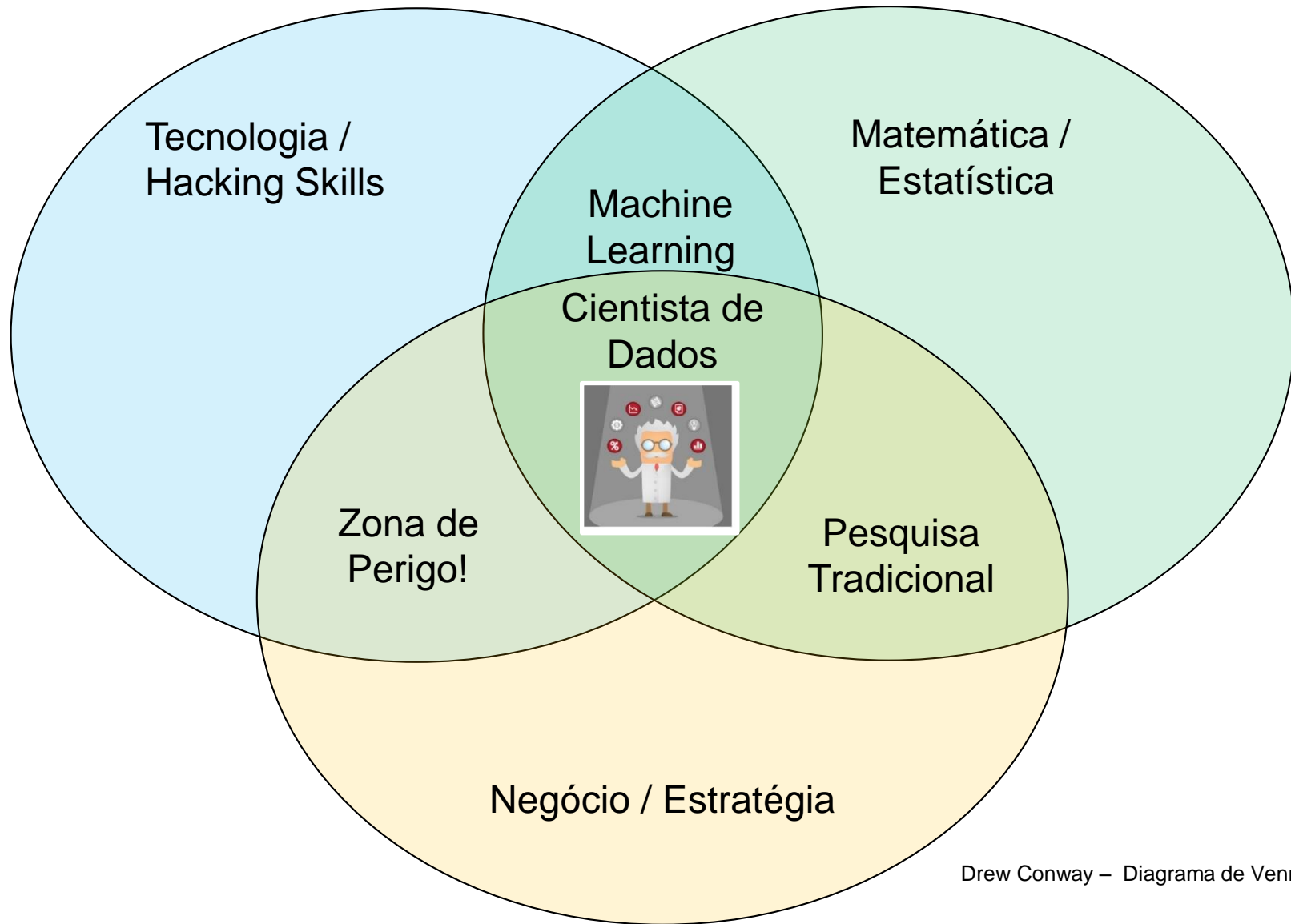
Fonte: White (2012)



# Racional para definir Sizing do HDFS

Identificação do Consumo	Descrição	Total Disponível (TBs)
1	Espaço Bruto (10 DNs x 12 Discos x 4 TBs)	480
2	Saldo = (ID 1) - (Espaço consumido por formatação do disco e file systems ext4 = 10%)	432
3	Saldo = (ID 2) - (Espaço reservado para Tarefas MapReduce - Arquivos intermediários - área de work. Parâmetro: dfs.datanode.du.reserved = 25% de cada disco = 0,9 TB))	324
4	Saldo = (ID 3) / (Quantidade de replicas = 3)	108
5	Saldo = (ID 4) - (ID 4)*(% de espaço reservado backup)	54
6	Saldo final disponível para uso	54

# Cientista de Dados (Data Scientist)



Drew Conway – Diagrama de Venn, 2010

# Perfis profissionais para atuar em iniciativas de Big Data



Fonte: Izotov (2015)



# Spark Core

# Processamento com o RDD

- RDD básico

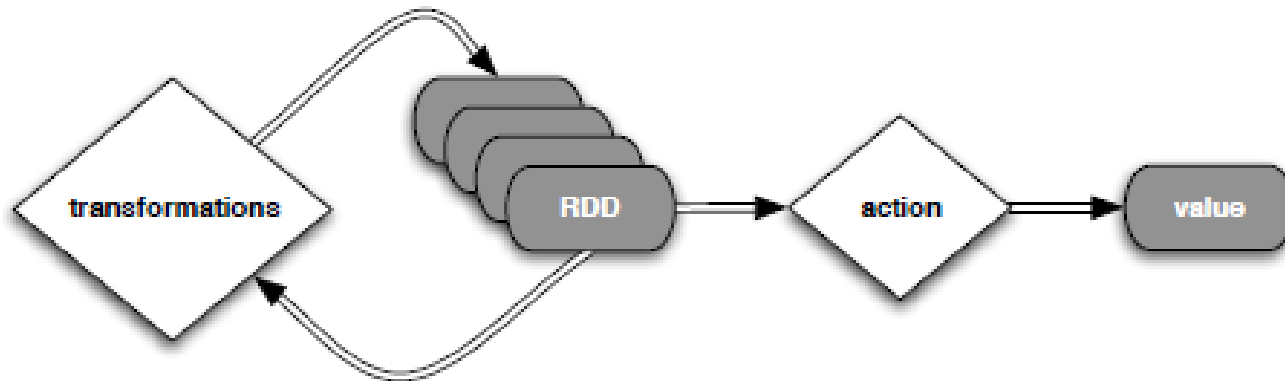
```
logs= sc.textFile("file:/home/training/webserver.log")
```

- Transformação RDD

```
hora23=logs.filter(lambda x: "2014:23:00" in x)
```

- Ação RDD

```
hora23.take(2)
```



Fonte: ZAHARIA et al (2015) e (2010)

# Transformações Básicas

- São operações que retornam um novo RDD.
- `numeros = sc.parallelize([1, 2, 3])`

Envia cada elemento por meio de uma função

- `squares = nums.map(lambda x: x*x) # => [1, 4, 9]`

Mantem os elementos apenas que passam no filtro

- `even = squares.filter(lambda x: x % 2 == 0) # => [4]`

Similar ao map, mas pode fazer com que cada item fique mapped com 0 ou mais itens.

- `numeros.flatmap(x => 1 to x) # => [1, 1, 2, 1, 2, 3]`

Fonte: ZAHARIA et al (2015) e (2010)

# Ações Básicas

- Retorna um valor final do programa.
- `numeros = sc.parallelize([1, 2, 3])`

Recupera o conteúdo de um RDD como se fosse uma coleção local.

- `numeros.collect() # => [1, 2, 3]`

Retorna os primeiro X elementos

- `numeros.take(2) # => [1, 2]`

Retorna a quantidade de elementos

- `numeros.count() # => 3`

Merge dos elementos com uma função associativa

- `numeros.reduce(lambda x, y: x + y) # => [1, 2, 3]`

Escrever os elementos em um arquivo texto

- `numeros.saveAsTextFile("hdfs://arquivo.txt")`

Fonte: ZAHARIA et al (2015) e (2010)

# Criando Contexto do Spark

Scala

```
import spark.SparkContext
import spark.SparkContext._

val sc = new SparkContext("masterUrl", "name", "sparkHome", Seq("app.jar"))
```

Java

```
import spark.api.java.JavaSparkContext;

JavaSparkContext sc = new JavaSparkContext(
    "masterUrl", "name", "sparkHome", new String[] {"app.jar"});
```

Python

```
from pyspark import SparkContext

sc = SparkContext("masterUrl", "name", "sparkHome", ["library.py"])
```

Fonte: ZAHARIA et al (2015) e (2010)

# Operações com Chave-Valor

- Retorna um valor final do programa.
- `animais = sc.parallelize([(elefante, 1), ( gato, 1), (gato, 2)])`

Observação - Implementa automaticamente o combiner no lado do map:

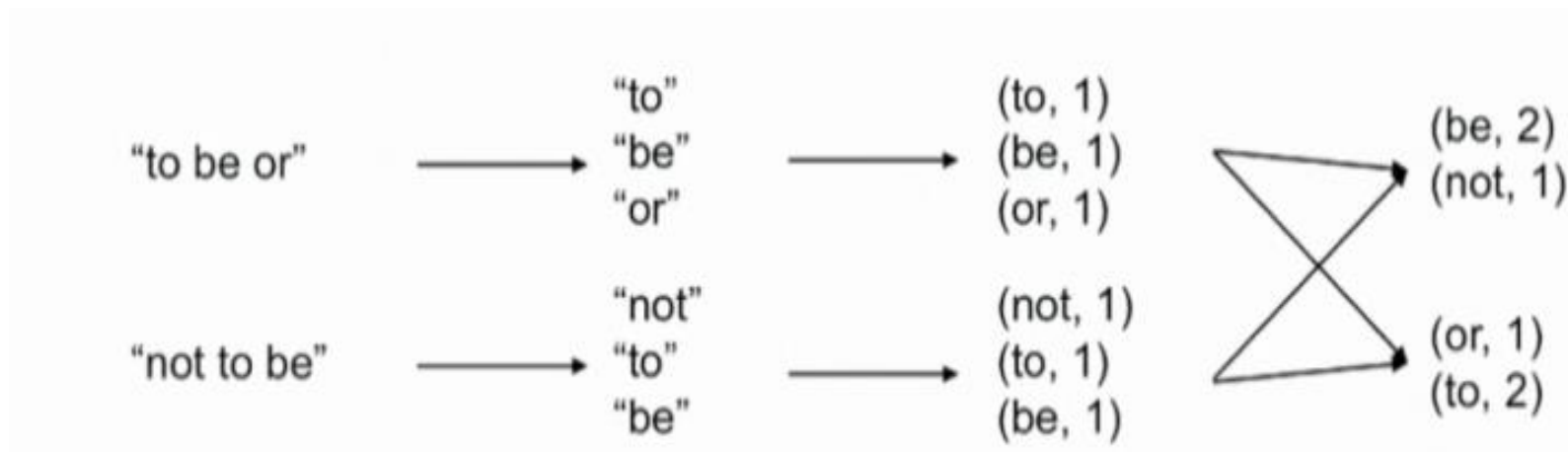
- `animais.reduceByKey(lambda x, y: x + y) # => [gato ,3], [elephante ,1]`
- `animais.groupByKey() # => [gato, Seq(1, 2)], [elefante, Seq(1)]`
- `animais.sortByKey() # => [gato, 1], [gato, 2], [elefante, 1]`

Fonte: ZAHARIA et al (2015) e (2010)

# Exemplo: Conta Palavras

- linhas = sc.textFile("hamlet.txt")

```
contagem = linhas.flatMap(lambda linha: linha.split(" ") \
    .map(lambda palavra: (palavra, 1)) \
    .reduceByKey(lambda x, y: x + y))
```



Fonte: ZAHARIA et al (2015) e (2010)

# Pares de RDDs

- Pares de RDD são formas especiais de RDD
  - Cada elemento tem que ser um par de chave-valor
  - Chave e valor podem ser de qualquer tipo
  - Utiliza algoritmos MapReduce
  - Funções comuns para processamento de dados podem ser utilizadas, exemplo: sort, join, group by.

(key1,value1)
(key2,value2)
(key3,value3)
...



# Criando Pares de RDDs

- O primeiro passo do workflow de processamento é obter os pares de chave-valor.
- Operações frequentemente utilizadas para criar os pares de RDD: map, flatMap, keyBy.
- Exemplo, criando pares de RDD separado por tabulação:

Python

```
> users = sc.textFile(file) \  
    .map(lambda line: line.split('\t')) \  
    .map(lambda fields: (fields[0],fields[1]))
```

Scala

```
> val users = sc.textFile(file) \  
    .map(line => line.split('\t')) \  
    .map(fields => (fields(0),fields(1)))
```

```
user001\tFred Flintstone  
user090\tBugs Bunny  
user111\tHarry Potter  
...
```



(user001,Fred Flintstone)
(user090,Bugs Bunny)
(user111,Harry Potter)
...

Fonte: Cloudera

# Exemplo de Web Logs – Chave pelo ID do usuário

Python

```
> sc.textFile(logfile) \  
    .keyBy(lambda line: line.split(' ')[2])
```

Scala

```
> sc.textFile(logfile) \  
    .keyBy(line => line.split(' ')[2])
```

User ID

```
56.38.234.188 - 99788 "GET /KBDOC-00157.html HTTP/1.0" ...  
56.38.234.188 - 99788 "GET /theme.css HTTP/1.0" ...  
203.146.17.59 - 25254 "GET /KBDOC-00230.html HTTP/1.0" ...  
...
```



```
(99788, 56.38.234.188 - 99788 "GET /KBDOC-00157.html...)
```

```
(99788, 56.38.234.188 - 99788 "GET /theme.css...)
```

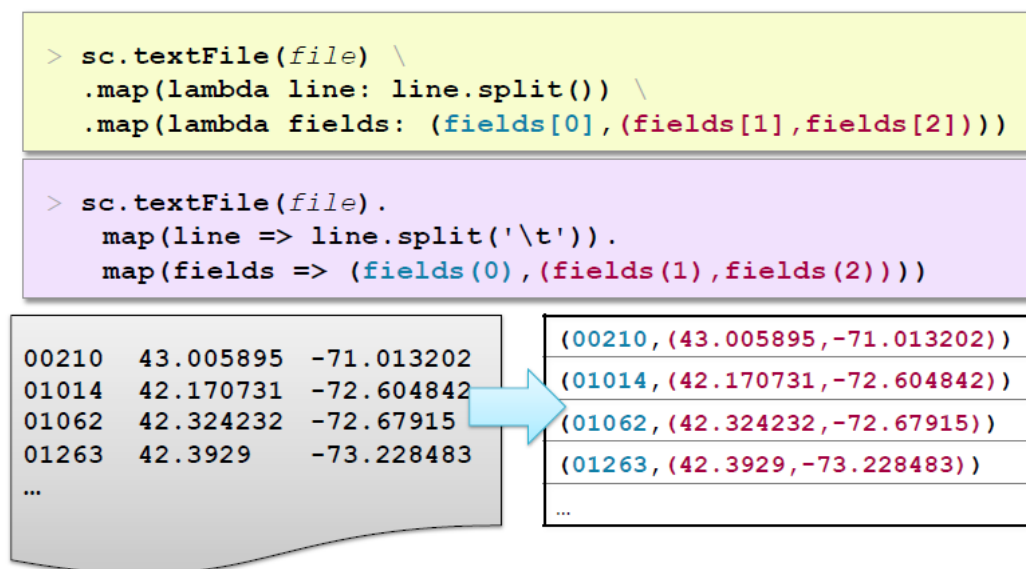
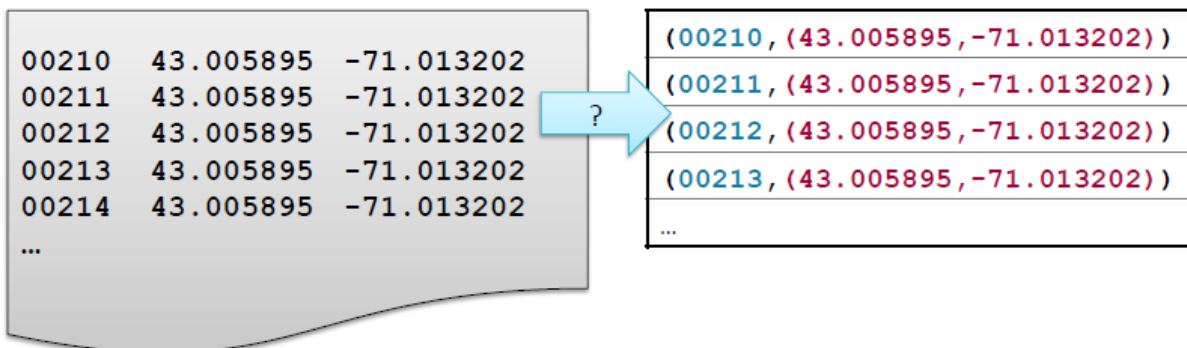
```
(25254, 203.146.17.59 - 25254 "GET /KBDOC-00230.html...)
```

```
...
```

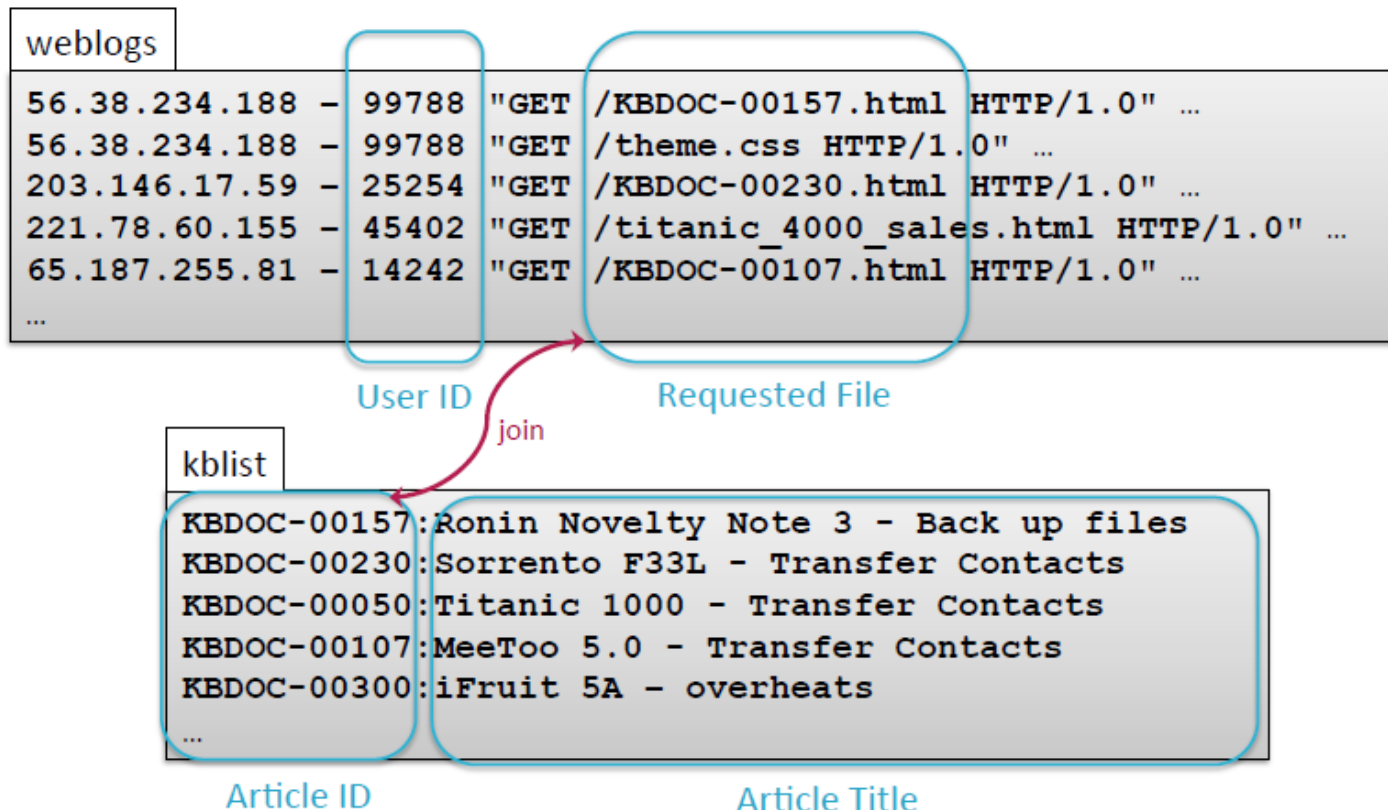
Fonte: Cloudera

# Exemplo de Pares com Valores Complexos

- Entrada: Lista de código postal com as respectivas latitudes e longitudes
- Saída: código postal (chave) e latitude / longitude pares (valores)



# Exemplo de Web Logs – Realizando Join entre pares de RDD



Fonte: Cloudera

# Exemplo de Web Logs – Passo 1A)

## Map (docid , userid)

```
> import re
> def getRequestDoc(s):
    return re.search(r'KBD0C-[0-9]*',s).group()

> kbreqs = sc.textFile(logfile) \
    .filter(lambda line: 'KBD0C-' in line) \
    .map(lambda line: (getRequestDoc(line),line.split(' ')[2])) \
    .distinct()
```

```
56.38.234.188 - 99788 "GET /KBD0C-00157.html HTTP/1.0" ...
56.38.234.188 - 99788 "GET /theme.css HTTP/1.0" ...
203.146.17.59 - 25254 "GET /KBD0C-00230.html HTTP/1.0" ...
221.78.60.155 - 45402 "GET /titanic_4000_sales.html HTTP/1.0" ...
65.187.255.81 - 14242 "GET /KBD0C-00107.html HTTP/1.0" ...
...
```

kbreqs

(KBD0C-00157,99788)

(KBD0C-00203,25254)

(KBD0C-00107,14242)

...

Fonte: Cloudera

# Exemplo de Web Logs – Passo 1B)

## Map (docid , nome do artigo)

```
> kblast = sc.textFile(kblastfile) \
  .map(lambda line: line.split(':')) \
  .map(lambda fields: (fields[0],fields[1]))
```

```
KBDOC-00157:Ronin Novelty Note 3 - Back up files
KBDOC-00230:Sorrento F33L - Transfer Contacts
KBDOC-00050:Titanic 1000 - Transfer Contacts
KBDOC-00107:MeeToo 5.0 - Transfer Contacts
KBDOC-00206:iFruit 5A - overheats
...
```



kblast

(KBDOC-00157,Ronin Novelty Note 3 - Back up files)
(KBDOC-00230,Sorrento F33L - Transfer Contacts)
(KBDOC-00050,Titanic 1000 - Transfer Contacts)
(KBDOC-00107,MeeToo 5.0 - Transfer Contacts)
...

Fonte: Cloudera


# Exemplo de Web Logs – Passo 2)

## Join pela chave docid

```
> titlereqs = kbreqs.join(kblist)
```

kbreqs
(KBD0C-00157,99788)
(KBD0C-00230,25254)
(KBD0C-00107,14242)
...

kblist
(KBD0C-00157,Ronin Novelty Note 3 - Back up files)
(KBD0C-00230,Sorrento F33L - Transfer Contacts)
(KBD0C-00050,Titanic 1000 - Transfer Contacts)
(KBD0C-00107,MeeToo 5.0 - Transfer Contacts)
...



(KBD0C-00157,(99788,Ronin Novelty Note 3 - Back up files))
(KBD0C-00230,(25254,Sorrento F33L - Transfer Contacts))
(KBD0C-00107,(14242,MeeToo 5.0 - Transfer Contacts))
...

Fonte: Cloudera

# Exemplo de Web Logs – Passo 3)

## Mapeando os resultados pelo formato desejado: (userid, título)

```
> titlereqs = kbreqs.join(kblist) \
    .map(lambda (docid, (userid, title)): (userid, title))
```

(KBD0C-00157, (99788, Ronin Novelty Note 3 - Back up files))
(KBD0C-00230, (25254, Sorrento F33L - Transfer Contacts))
(KBD0C-00107, (14242, MeeToo 5.0 - Transfer Contacts))
...



(99788, Ronin Novelty Note 3 - Back up files)
(25254, Sorrento F33L - Transfer Contacts)
(14242, MeeToo 5.0 - Transfer Contacts)
...

Fonte: Cloudera



# Exemplo de Web Logs – Passo 4)

## GroupByKey userid - Agrupando por chave

```
> titlereqs = kbreqs.join(kblist) \
    .map(lambda (docid,(userid,title)): (userid,title)) \
    .groupByKey()
```

(99788,Ronin Novelty Note 3 - Back up files)
(25254,Sorrento F33L - Transfer Contacts)
(14242,MeeToo 5.0 - Transfer Contacts)
...



Note: values  
are grouped  
into Iterables

(99788,[Ronin Novelty Note 3 - Back up files, Ronin S3 - overheating])
(25254,[Sorrento F33L - Transfer Contacts])
(14242,[MeeToo 5.0 - Transfer Contacts, MeeToo 5.1 - Back up files, iFruit 1 - Back up files, MeeToo 3.1 - Transfer Contacts])
...

Fonte: Cloudera

# Exemplo de Web Logs – Passo 5)

## print - Saída (output)

```
> for (userid,titles) in titlereqs.take(10):
    print 'user id: ',userid
    for title in titles: print '\t',title
```

```
user id: 99788
    Ronin Novelty Note 3 - Back up files
    Ronin S3 - overheating
user id: 25254
    Sorrento F33L - Transfer Cont
user id: 14242
    MeeToo 5.0 - Transfer Contact
    MeeToo 5.1 - Back up files
    iFruit 1 - Back up files
    MeeToo 3.1 - Transfer Contacts
```

```
(99788,[Ronin Novelty Note 3 - Back up files,
Ronin S3 - overheating])
(25254,[Sorrento F33L - Transfer Contacts])
(14242,[MeeToo 5.0 - Transfer Contacts,
MeeToo 5.1 - Back up files,
iFruit 1 - Back up files,
MeeToo 3.1 - Transfer Contacts])
...
```

Fonte: Cloudera

# Exercício 5 - Spark

Abra o arquivo de exercícios de Spark.

# Prof. MSc. Samuel Otero Schmidt



[www.linkedin.com/pub/samuel-otero-schmidt/16/358/a98](https://www.linkedin.com/pub/samuel-otero-schmidt/16/358/a98)



[@schmidt\\_samuel](https://twitter.com/schmidt_samuel)



Schmidt\_Samuel / Samuel Otero Schmidt



[schmidt-samuel@usp.br](mailto:schmidt-samuel@usp.br)

# Referências Bibliográficas

- Patrick Wendell. The Future of Apache Spark, Spark Summit, 2014. Disponível em: [e spark-summit.org/wp-content/uploads/2014/07/Future-of-Spark-Patrick-Wendell.pdf](http://spark-summit.org/wp-content/uploads/2014/07/Future-of-Spark-Patrick-Wendell.pdf)
- ZAHARIA, Matei et al. Spark: cluster computing with working sets. In: **Proceedings of the 2nd USENIX conference on Hot topics in cloud computing**. 2010. p. 10-10.
- ZAHARIA, Matei et al. Learning Spark: Lightning-fast Data Analysis, O'Reilly Media, 2015.  
<http://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html>
- Ryza , Sandy; Uri Laserson; Sean Owen; Josh Wills. Advanced Analytics with Apache Spark: The Book, O'Reilly Media, 2015.
- MapR. The Future of Hadoop *is Right Now*, 2014. Disponível em: <https://www.mapr.com/wwgd>
- Kestelyn, J.** How Edmunds.com Used Spark Streaming to Build a Near Real-Time Dashboard, 2014. Disponível em: <http://blog.cloudera.com/blog/2015/03/how-edmunds-com-used-spark-streaming-to-build-a-near-real-time-dashboard/>
- Lee, Jungryong. Big Telco, Bigger real-time Demands: Moving Towards Real-time analytics - Jungryong Lee (SK Telecom), 2014. Disponível em: <https://spark-summit.org/2014/talk/big-telco-bigger-real-time-demands-moving-towards-real-time-analytics>
- Shapira, Gwen, Jonathan Seidman, Ted Malaska, Mark Grover, Hadoop Application Architectures, O'Reilly Media, Inc., 2015.  
[http://stanford.edu/~rezab/sparkclass/slides/itas\\_workshop.pdf](http://stanford.edu/~rezab/sparkclass/slides/itas_workshop.pdf)
- WHITE, Tom. Hadoop: The definitive guide. " O'Reilly Media, Inc.", 2012.
- Sawant, N; Shah, H. Big Data Application Architecture Q&A, 2013
- Izotov, I. Finding the right people for your Hadoop initiative, 2015. <https://www.linkedin.com/pulse/finding-right-people-your-hadoop-initiative-igor-izotov>
- Conway, D. The Data Science Venn Diagram, 2010. <http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>
- WHITE, Tom. Hadoop: The definitive guide. " O'Reilly Media, Inc.", 2012.
- ITAS - [http://stanford.edu/~rezab/sparkclass/slides/itas\\_workshop.pdf](http://stanford.edu/~rezab/sparkclass/slides/itas_workshop.pdf)