XXXVIII Congresso da Sociedade Brasileira de Computação
#ComputaçãoeSustentabilidade
22 a 26 de julho | Centro de Convenções | Natal-RN

Realizado por: SBC

Rumo à Otimização de Operadores sobre UDF no Spark

João Antonio Ferreira[1], Fábio Porto[2], Rafaelli Coutinho[1], Eduardo Ogasawara[1]

[1] CEFET/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
[2] LNCC - Laboratório Nacional de Computação Científica

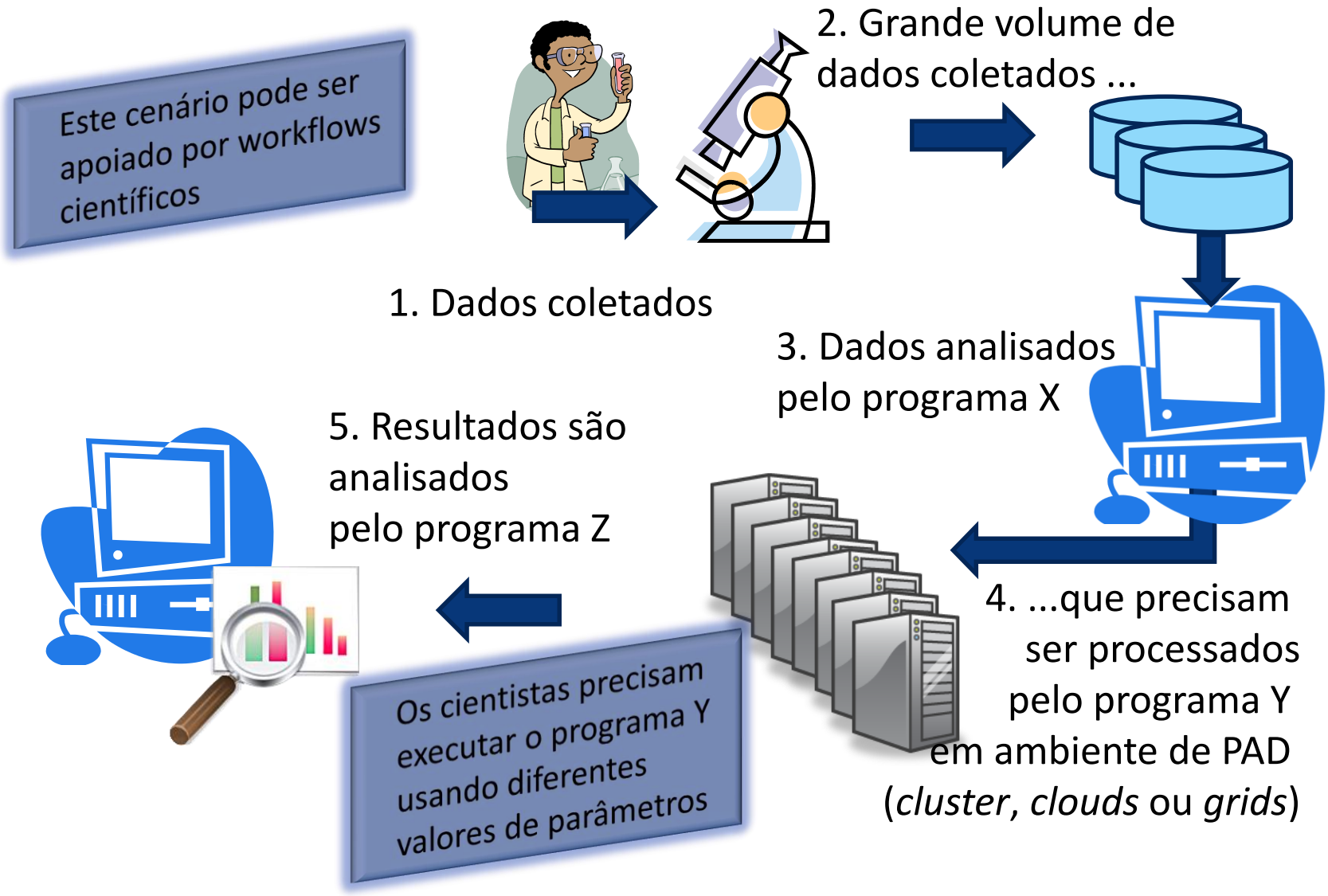
CEFET/RJ

Laboratório Nacional de Computação Científica

Agenda

- Introdução
- Álgebra de Workflows
- Implementação em Spark
- Considerações Finais

Experimentos científicos

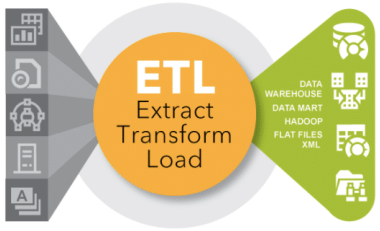


CSBC 2011 - Silva, V., Chirigati, F., Ogasawara, E., Dias, J., Oliveira, D., Porto, F., Valduriez, P., Mattoso, M., (2011), "Uma avaliação da Distribuição de Atividades Estática e Dinâmica em Ambientes Paralelos usando o Hydra". In: *Brazilian e-Science Workshop*, p. 1–8

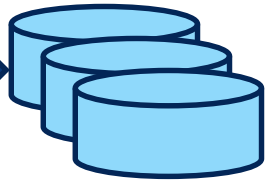
Cenário de Análise de Dados (Data Science)

Este cenário pode ser apoiado por linguagens de programação centrada em dados

1. Dados coletados



2. Grande volume de dados produzidos ...



3. Análise exploratória

4. ...que precisam ser processados por rotinas de MD em ambientes Hadoop/Spark

5. Resultados são analisados

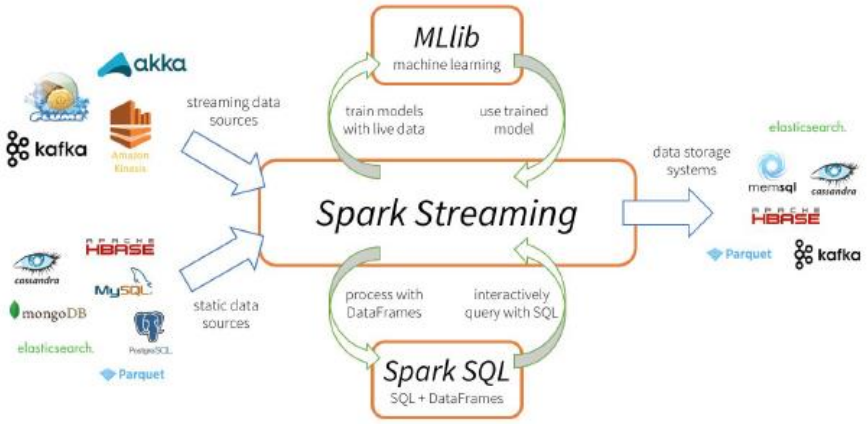
Os cientistas de dados precisam executar tarefas de MD usando diferentes valores de parâmetros

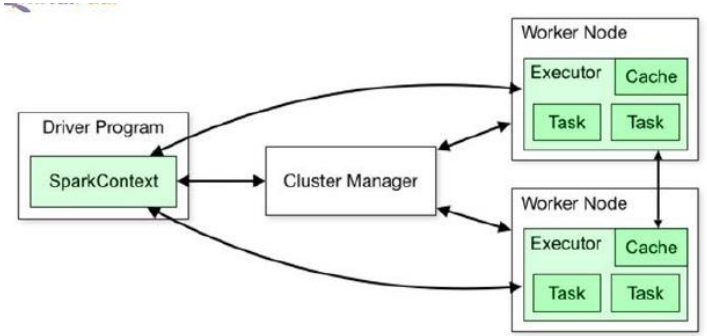
Data Intensive Scalable Computing

- Convergência de Aplicações e Infraestrutura
  - Vários cenários de aplicações nas Ciências-Empresas-Governos
  - Amplo espectro de ferramentas para análise de dados
- Há demandas em aberto
  - Otimização de execução
  - Rastreabilidade
  - *Steering*
  - …

Bryant, R. E., (2011), "Data-intensive scalable computing for scientific applications", *Computing in Science and Engineering*, v. 13, n. 6, p. 25–33.

Considere o cenário de Análise de Dados usando Apache Spark






```
object RandomForest {

  def run(idf: DataFrame, attribute: String, params: HashMap[String, Any] = null, callback: String => Double): entities.Entities.NumericImputationResult = {

    val attributes: Array[String] = params("attributes").asInstanceOf[Array[String]]

    val removeCol = idf.columns.diff(attributes).filter(_ != "lineId")
    val remidf = idf.drop(removeCol: _*)

    val context = remidf.sparkSession.sparkContext

    val calcCol = attributes.filter(_ != attribute)

    var fidf = context.broadcast(utility.Utility.filterNullAndNonNumeric(remidf, calcCol))

    val columns = fidf.value.columns

    val lineIdIndex = columns.indexOf("lineId")
    val attributeIndex = columns.indexOf(attribute)

    val vectorsRdd = fidf.value.rdd.map(row => {

      val lineId = row.getLong(lineIdIndex)
      val attributeValue = row.getString(attributeIndex)
```

Zona de conforto!

Saindo da Zona de conforto

- *Hard mode* para Cientista de Dados
  - Implementar “cegamente” as rotinas em Spark
- Tem-se disponíveis diversas bibliotecas escritas em Python (*NumPy, SciPy, Scikit-learn*) e em pacotes R
  - proveem aos cientistas uma série de soluções numéricas para análise exploratória, cálculo vetorial, álgebra linear, métodos estatísticos, visualização de dados e aprendizado de máquina
- *Hard mode* for Spark
- O uso de tal ferramental nos frameworks como Apache Spark comumente recai na implementação de funções definidas pelo usuário (UDF)

UDFs

- O uso de UDF traz desafios na otimização de execução
  - Dificuldade de se estabelecer semântica clara sobre seus comportamentos
    - Via de regra, possuem códigos ad-hoc relacionados ao domínio
- A falta de conhecimento sobre a semântica da UDF, faz com que *frameworks* como o Spark sejam incapazes de otimizar a execução
- Rheinlander et al. (2017) dedicaram um estudo extenso a questão da otimização de execução de UDF em dataflow
  - Apontam principais desafios
  - Possíveis para solução, dentre as quais a álgebra de workflow [Ogasawara et al. 2011, Rheinlander et al. 2015]

Rheinländer, A., Leser, U., Graefe, G., (2017), "Optimization of complex dataflows with user-defined functions", *ACM Computing Surveys*, v. 50, n. 3.
Ogasawara, E., de Oliveira, D., Valduriez, P., Dias, J., Porto, F., Mattoso, M., (2011), "An algebraic approach for data-centric scientific workflows", *Proceedings of the VLDB Endowment*, v. 4, p. 1328–1339.
Rheinlander, A., Heise, A., Hueske, F., Leser, U., Naumann, F., (2015), "SOFA: An extensible logical optimizer for UDF-heavy data flows", *Information Systems*, v. 52, p. 96–125.

Álgebra de Workflows

- Relações como modelo para consumo e produção de dados
- Operações trazendo semântica às atividades

Operação	UDF	Operandos adicionais	Resultado	Cons. $\times$ Prod. tuplas
Map	programa	Relação R	Relação S	$1:1$ *por* $\lvert R\rvert$
SplitMap	programa	Relação R	Relação S	$1:m$ *por* $\lvert R\rvert$
Reduce	programa	Relação R e atrs	Relação S	$n:1$ *por* $\lvert\pi_{atrs}\rvert$
Filter	programa	Relação R	Relação S	$1:(0-1)$ *por* $\lvert R\rvert$
SRQuery	exp. relacional	Relação R	Relação S	$n:m$
MRQuery	exp. relacional	Relações $(R_1 \cdots R_i)$	Relação S	$(n_1 \cdots n_i):m$

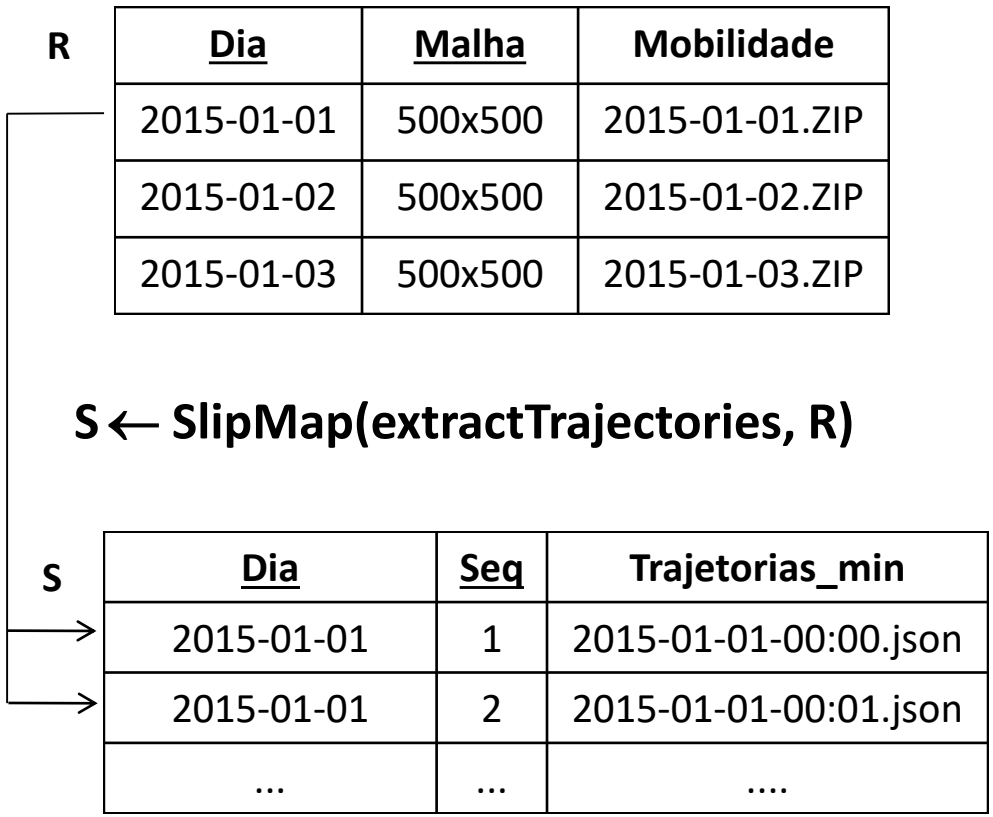
Formalização da otimização usando operações algébricas

Relações como modelo de dados para consumo e produção das atividades

- Relações são definidas como um conjunto de tuplas de tipos primitivos (integer, float, string, date) e tipos de dados complexos (e.g. referencias a arquivos)
- Examplo: R($\mathcal{R}$)
- $\mathcal{R}$ = (dia: Date, malha: String; mobilidade: FileRef)

Dia	**Malha**	**Datafiles**
2015-01-01	500x500	2015-01-01.ZIP
2015-01-02	500x500	2015-01-02.ZIP
2015-01-03	500x500	2015-01-03.ZIP

Split Map Activity (SplitMap)
T ← SplitMap(Y, R)



Reduce Activity (Reduce)
$T \leftarrow Reduce(Y, g_A, R)$

S

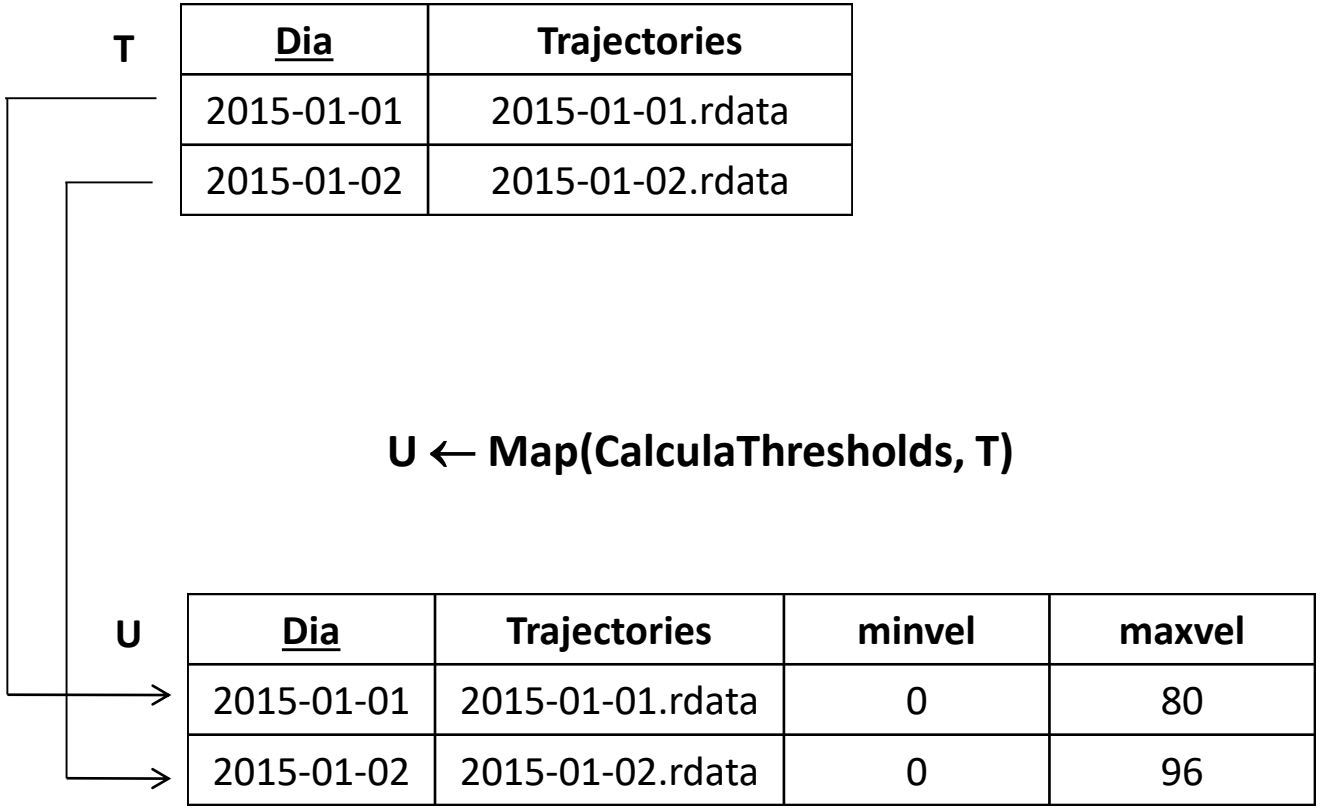
Dia	**Seq**	**Trajetorias_min**
2015-01-01	1	2015-01-01-00:00.json
2015-01-01	2	2015-01-01-00:01.json
…	…	….

T ← Reduce(ProduceRData, {'Dia'}, S)

T

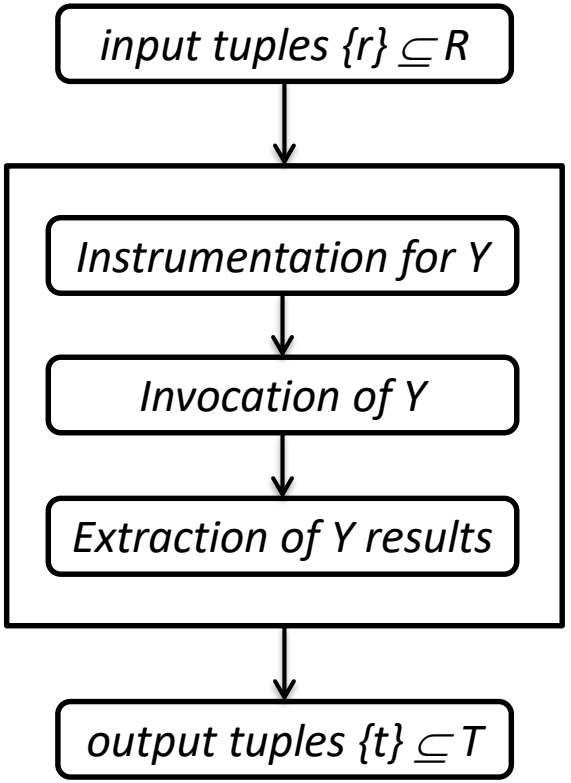
Dia	**Trajetorias**
2015-01-01	2015-01-01.rdata
…	…

Atividade de Map (Map)
U ← Map(calculaThresholds, T)



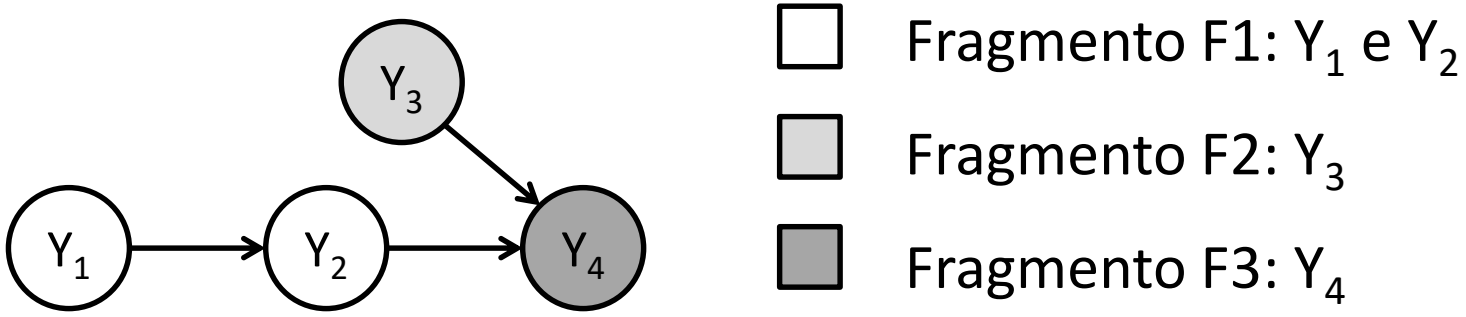
Ativação de Atividade

- Uma ativação é um objeto autocontido contendo toda a informação necessária para execução da atividade em uma unidade computacional (i.e. o programa a executar e o dados em sua granularidade mínima)



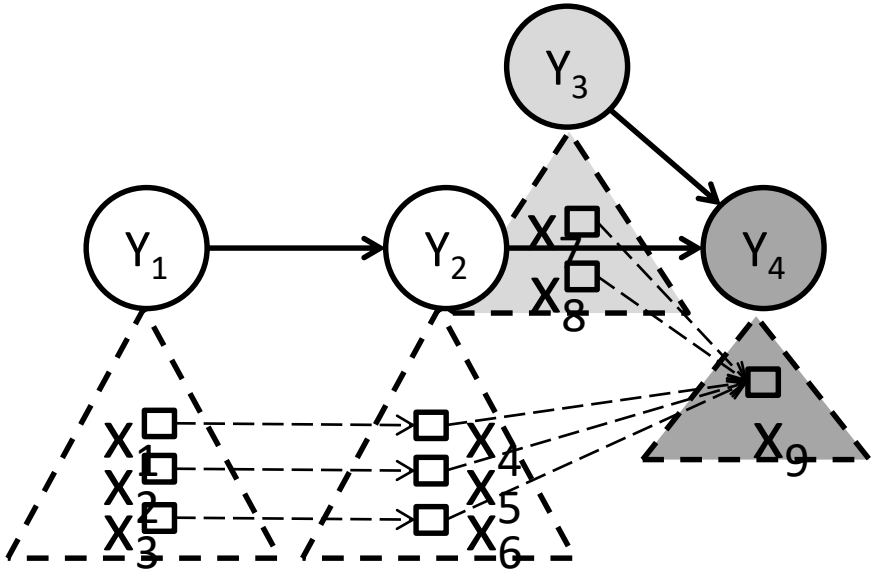
Fragmento de Workflows

- Um fragmento *F* de um workflow W é um subconjunto de atividades de W tal que:
  - ou *F* é um conjunto unitário
  - or $\forall Y_j \in F, \ \exists\, Y_i \in F \mid (Dep(Y_i, Y_j)) \vee (Dep(Y_j, Y_i))$.



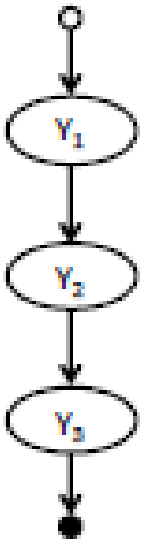
Ativações em um modelo de execução

- Dado um workflow W, um conjunto de ativações $X=\{x_1, \ldots, x_k\}$ é criado para a sua execução
- Cada ativação x_i pertence a uma atividade particular Y_j, representado por $Act(x_i) = Y_j$ e com relação de dependência entra as suas atividades



Plano de Execução de Workflows

- Expressões Algébricas + Modelo de Execução
  → Expressões Algébricas Otimizadas
- Mapeamento Físico para Operações em Spark



(a)

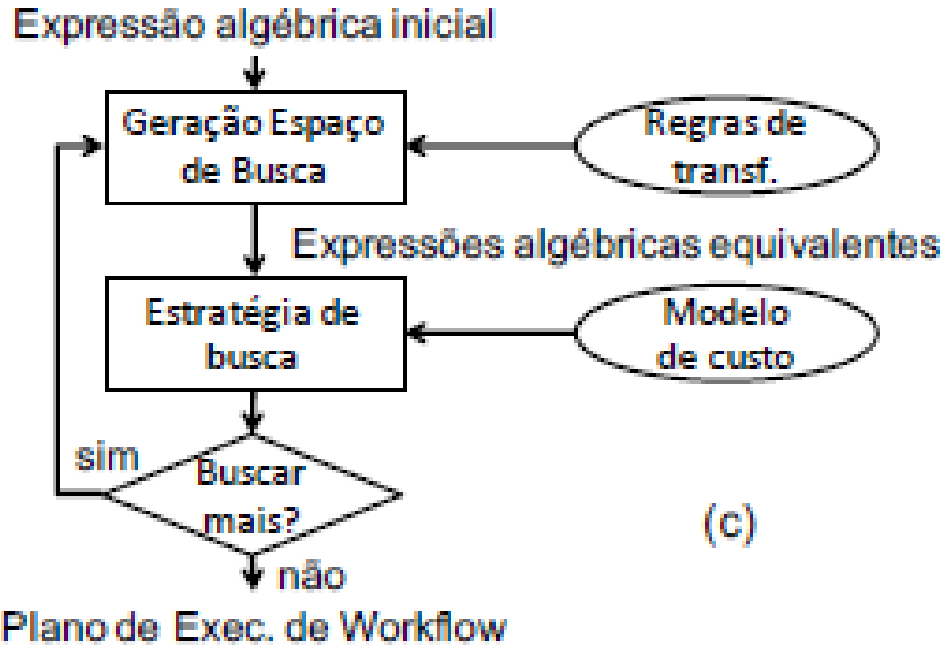
$S \leftarrow Map(Y_1, R)$

$T \leftarrow Filter(Y_2, S)$

$U \leftarrow Filter(Y_3, T)$

ℜ(id, a, t)
ℌ(id, a, b, t)
ℑ(id, a, b, t)
ℵ(id, a, b, t)

(b)



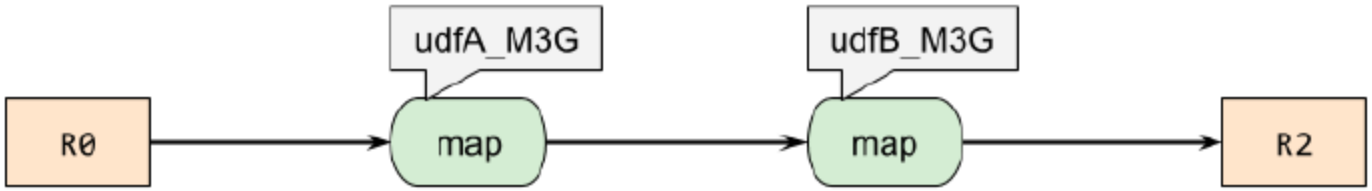
(c)

Modelo de Execução de Workflow

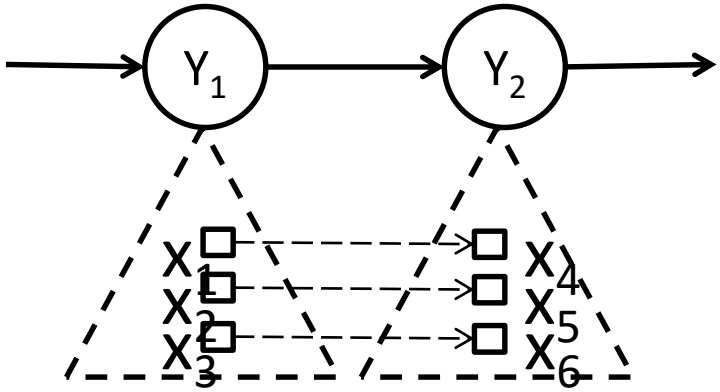
- Modelo depende da plataforma
  - Ativação da atividade
  - Estabelecimento de fragmentos de workflow
  - Estratégias de fluxo de dados e despacho das ativações
- Apache Spark
  - “Distribuição estática” para despacho de ativações
  - Contém um conjunto de otimizações básicas para dataflow buscando pipelines (“fluxo de dados”) sempre que possível (“fragmentos de workflow”)

O que acontece se os Maps das UDFs são extremamente computacionalmente intensivos?

```
myDataset.map(heavyUDF1(some, parameters)).map(heavyUDF2(param, ...))
```

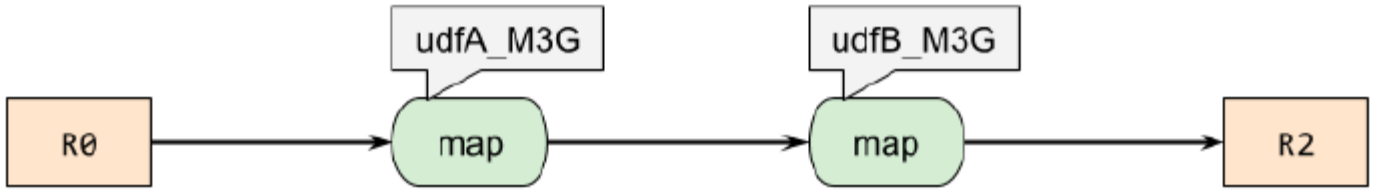


(a)

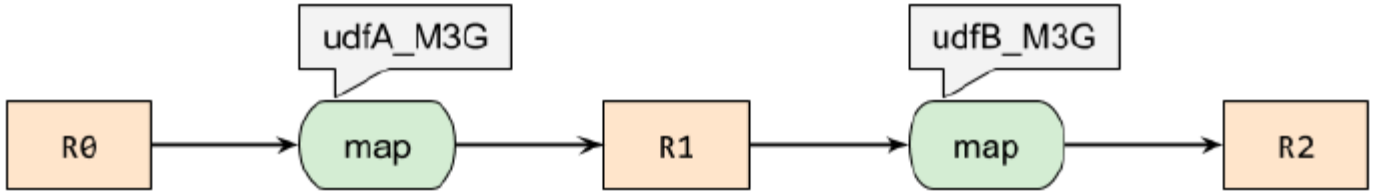


Inclusão de materialização

```
myDataset.map(heavyUDF1(some, parameters)).map(heavyUDF2(param, ...))
```



(a)



(b)

Atualmente não é possível incluir esta etapa de otimização no Catalyst (foi feito na etapa pré-otimização)

Considerações finais

- Fase de implementação consolidada do framework
- Fase de implementação de um conjunto restrito de otimizações integradas no Catalyst

Contato

CEFET/RJ

Eduardo Ogasawara
eogasawara@ieee.org
http://eic.cefet-rj.br/~eogasawara

Agradecimentos

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

CAPES

FAPERJ
Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro