Framework para Deduplicação de Dados com Apache Spark

César Magrin - magrin@inf.ufpr.br
Disciplina: Metodologia Científica - CI860
Professor: Alexandre Direne

Sumário

Qualidade de Dados (Data Quality)

- Grande volume de dados.
- Dados de varias fontes: Dispositivos moveis, redes sociais, vendas, consultas medicas, dados bancários, entre outros.
- Data Cleaning.
- KDD (Knowledge Discovery in Databases).

Deduplicação

- Identificar registros diferentes ou múltiplos referentes a um único objeto no mundo real.
- Razões:
  - Entrada de dados incorretos.
  - inconsistências por diferentes formatos de entrada.Ex: “Cesar Magrin” vs “Cesar M”.
  - Informações incompletas.

Motivação

- Dissertação: “*Processo Complementar para Detecção de Registros Duplicados em Bases CADSUS*” - MSc. Osvaldo M. Cavalieri[3]
- Paraná - Cadastros CADSUS x População

Data	Total CADSUS	População IBGE	Percentual
13/01/2010	13.348.838	10.444.526	127,80%
03/01/2012	14.292.902	10.577.755	135,12%

Fonte: MS/Datasus e IBGE

Processo de Christen

- Método de Fellegi e Sunter
  - Calcular pesos parciais para cada par de atributos.
  - Utilizado para definir grau de semelhança entre registros.
  - Classifica em: Duplicados, possivelmente duplicados ou diferentes.
- Baseado neste modelo, Christen[1,2] apresentou um modelo para deduplicação. Divido em etapas, podendo em cada etapa utilizar diferentes algoritmos.

Processo de Christen

1. Pré-Processamento
2. Blocagem
3. Comparação
4. Classificação
5. Avaliação e Revisão Manual

1-Pré-processamento

- A etapa de pré-processamento consiste no trabalho de converter dados para um formato padronizado.
- Algumas etapas:
  - Exclusão ou substituição de caracteres desnecessários.
  - Busca e correção de erros comuns de digitação.

2-Blocagem

- Comparação mxn se torna inviável.Ex: 20k registro, geram 400.000.000 de comparações.
- Os registros devem ser separados em blocos de similaridade, afim de diminuir o numero de comparações.
- Escolha da chave de bloco, codificação da chave (soundex)

3-Comparações

- Verificar bloco a bloco e comparar os registros entre si para identificar o grau de similaridade.
- Há diversos algoritmos para calculo de similaridade: comparação de strings, comparação de datas,...
- Etapa que exige o maior esforco computacional.

4-Classificação

- nesta etapa é calculado o peso final da similaridade que será comparada a limiares de corte.
- Função de Similaridade
  - sim(a,b) =1 -> ⟹ Atributos iguais
  - sim(a,b) =0 -> ⟹ Atributos diferentes
  - 0< sim(a,b) >1 ⟹ Grau de similaridade

Resultados

- Testes realizados na dissertação do Osvaldo[3].
- Base de dados: 238.691 cadastros (CADSUS - Colombo)
- Febrl[2]
- Etapa de blocagem:

Tamanho	Tempo		Quantidade	
Sufixo	Codificação	Indexação	Blocos	Pares registros
5	18,55s	7,24s	94.007	59,05 milhões

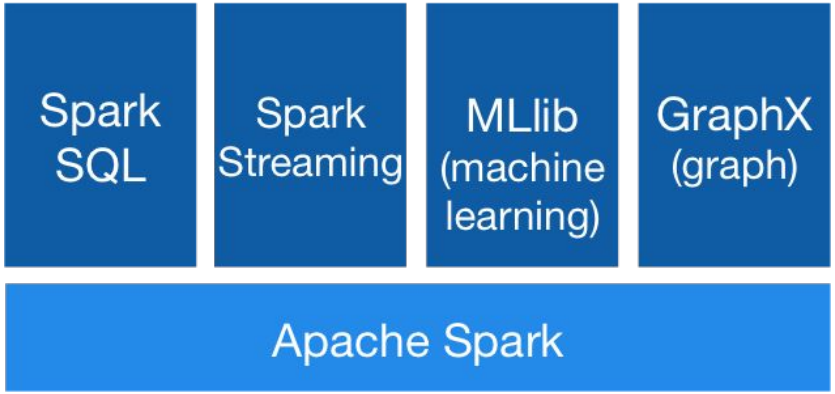
Resultados(cont.)

- Etapa de comparação: etapa que exige maior custo computacional, a etapa anterior conseguiu diminuir o numero de comparações para aprox. 59milhões. Sem ela seria em torno de 50 bilhões.

Tamanho Sufixo	Pares de Registros	Tempo Processamento	Mémoria Utilizada
8	4.10 milhões	12 min	1.91 GB
7	6.75 milhões	20 min.	3.04 GB
6	16.65 milhões	51 min.	6.91 GB
5	59.05 milhões	720 min.[7]	N. Disponível[8]

Processamento Paralelo e Distribuído

- Spark[5]
  - Um projeto de pesquisa da AMPLab UC Berkeley 2009
  - Um framework open source para processamento paralelo
  - Uma engine empregada, na maioria dos casos, para o processamento de dados em larga escala



Processamento Paralelo e Distribuído(cont.)

- Paradigma MapReduce[4]
  - Adequado para trabalhar com problemas que poder ser particionados ou fragmentados em sub problemas
  - Etapas: Map, Shuffle e Reduce
  - Hadoop[4] vs Spark[5]

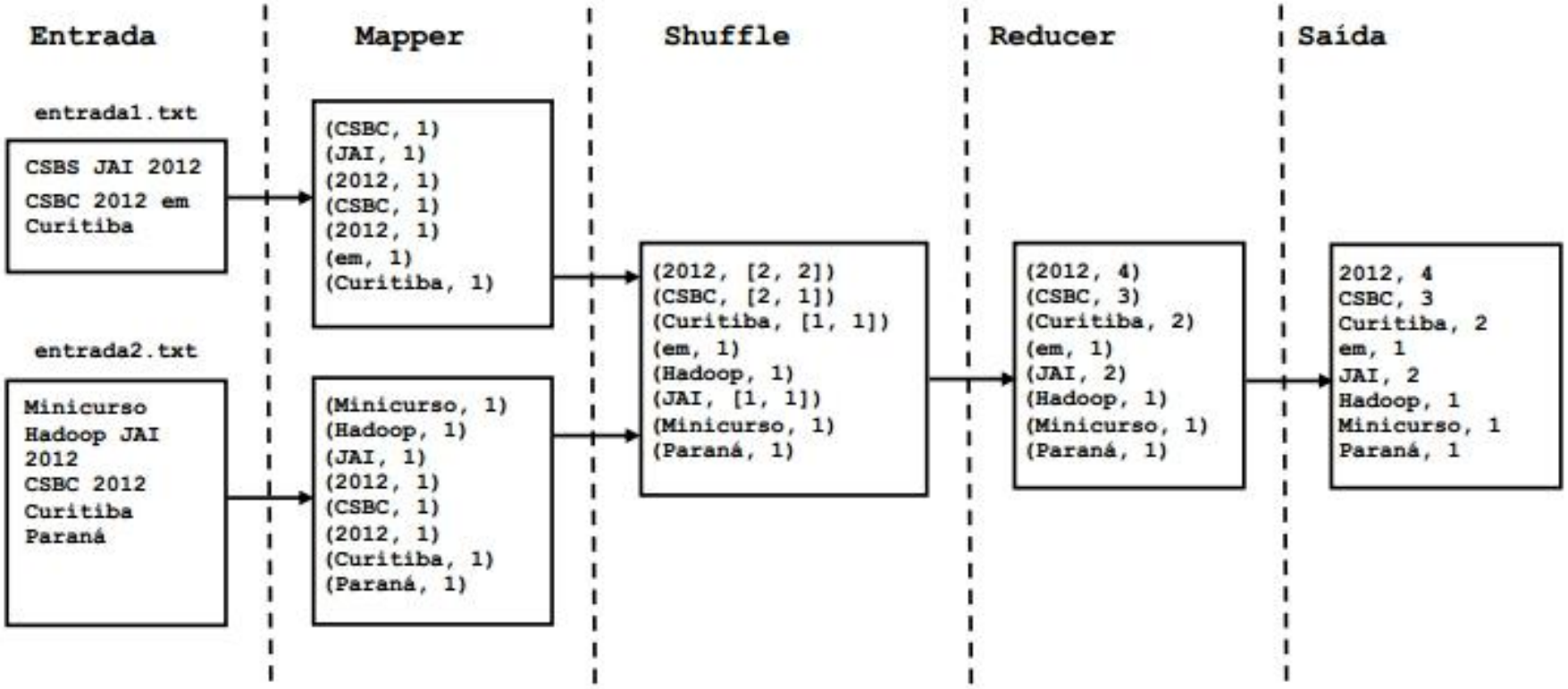
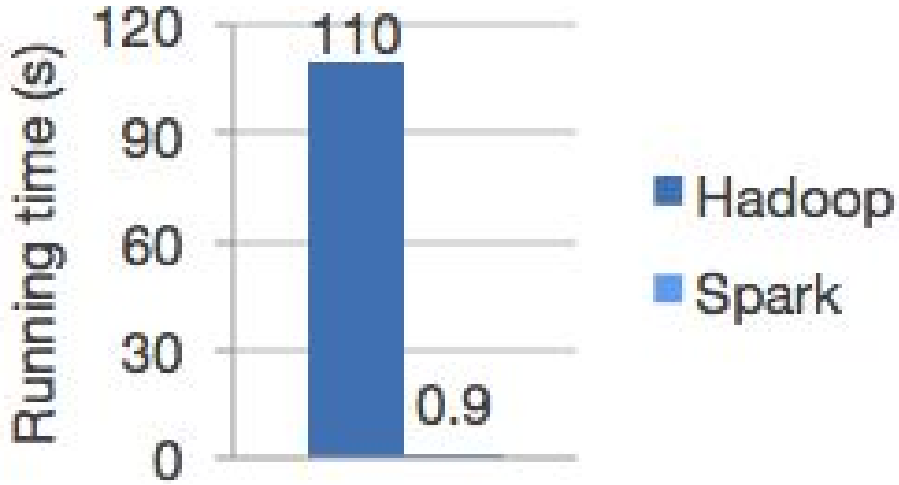


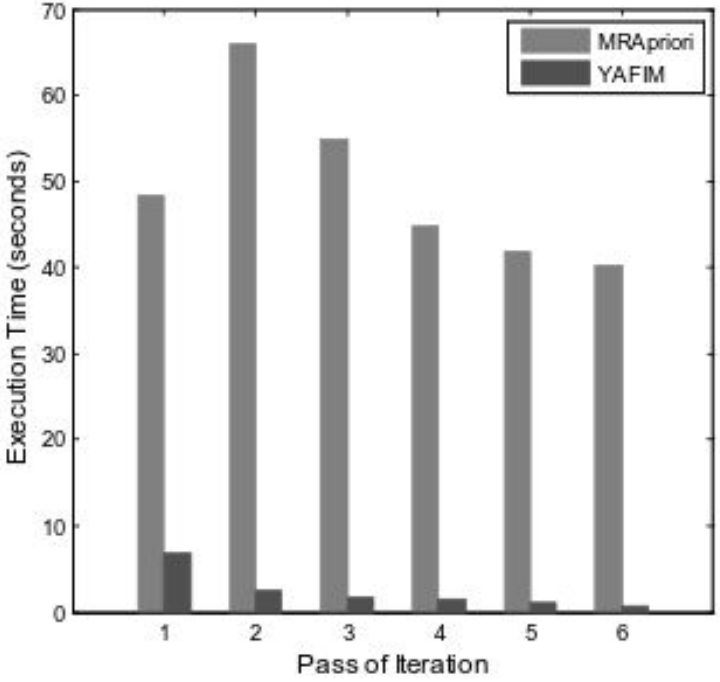
Figura 3.6. Fluxo lógico de execução da aplicação MapReduce do exemplo WordCount

Fonte: Figura 3.6[4]

Exemplos: Hadoop vs Spark



Logistic regression in Hadoop and Spark
Fonte: spark.apache.org



(d) Pumsb_star: Sup = 65%.

Yafim - A Parallel Frequent Itemset Mining Algorithm with Spark

Processamento Paralelo e Distribuído(cont.)

- RDD
  - Resilient - Se os dados em memória forem perdidos, eles podem ser recriados
  - Distributed - Particiona os elementos entre cluster para serem computados paralelamente;
  - Dataset - Os dados iniciais podem ser criados através de um arquivo externo (HDFS, HBase,...) ou paralelizando uma coleção existente.

Processamento Paralelo e Distribuído(cont.)

- RDD(cont.)
  - Coleção de Objetos imutáveis espalhados em clusters;
  - Criado através de transformações paralelas(map, reduce, filter,..);
  - Recriada automaticamente em caso de falha;
  - Controla a persistência (Cache) para reuso.

Conclusão

- A deduplicação é de fato uma tarefa importante no processo de Data Cleaning, sua falta pode acarretar em falhas num processo de tomada de decisão por exemplo.
- Segundo os resultados obtidos na dissertação do Osvaldo, mostra a necessidade de um sistema paralelo e distribuído para a eficacia do processo em bases maiores.
- O Spark aparenta ser a melhor solução para este processo com uso da paradigma MapReduce.

Referências

1. Peter Christen. Data Matching. 2011
2. Peter Christen e Tim Churches. Febrl -Freely extensible biomedical record linkage. 2002
3. Osvaldo, M Cavalieri, Bruno Muller, e Marcos Sfair Sunyê. Processo Complementar para Detecção de registros duplicados em Bases CADSUS.
4. Alfredo Goldman, Fabio Kon, Francisco. Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades.Cap 3.4 .2012
5. https://spark.apache.org/ acesso: 22/10/2015

Obrigado!

César Magrin
magrin@inf.ufpr.br