

Avaliação e paralelização de algoritmos de mapeamento para grafos de visibilidade

Matheus Nascimento Tavares

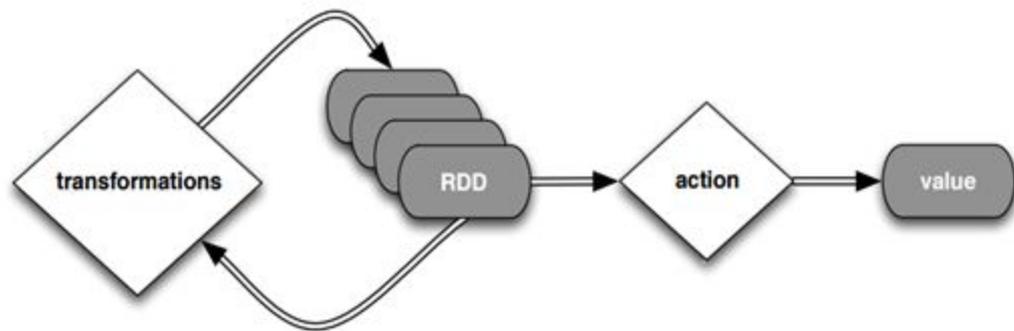
Apache Spark

- Plataforma de computação distribuída
- Combina diferentes tipos de componentes
 - SparkSQL, SparkStreaming, MLlib e GraphX.
- Linguagens
 - *Java, Python e Scala.*
- Utilizado por várias empresas
 - Amazon, eBay, IBM, Nasa, Samsung , Yahoo! , entre outras.

Apache Spark

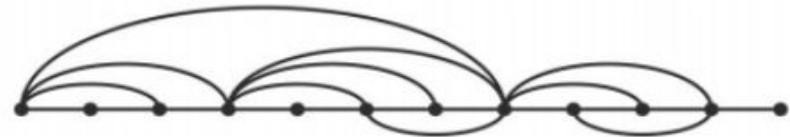
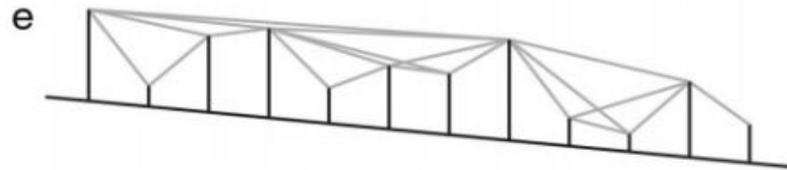
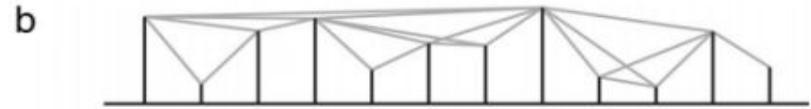
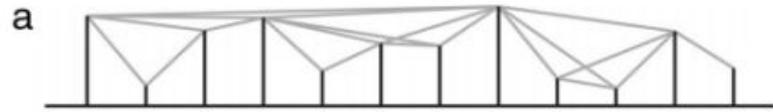
RDD - Resilient Distributed Dataset

- Imutável e persistente;
- Tolerante a falhas;
- Todas as transformações são *lazy*.



Grafo de visibilidade

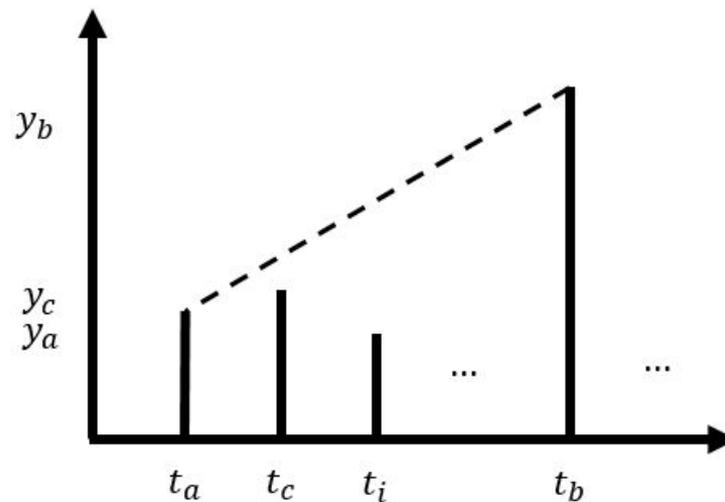
Cada elemento da série é representado por um vértice e as arestas representa a visibilidade entre os vértices.



Grafo de visibilidade

Podemos descrever a técnica de visibilidade com a função de mapeamento proposta por [Lacasa et al. 2008], que utiliza a equação (1) como condição de visibilidade entre dois pares de pontos.

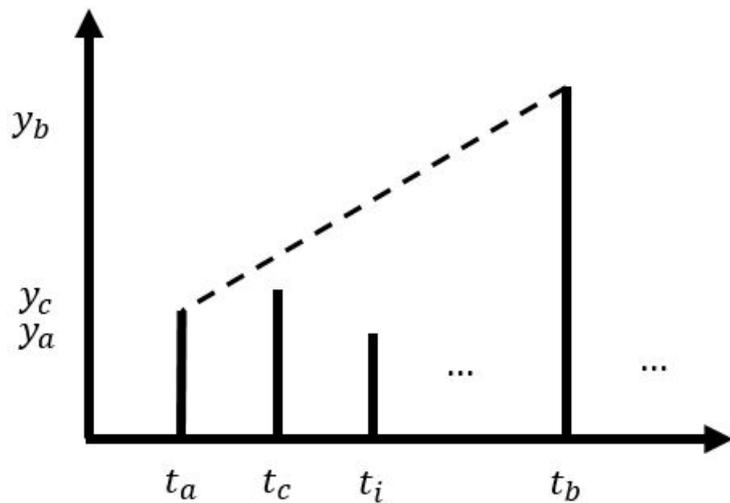
$$(1) \quad y_c < y_b + (y_a - y_b) \frac{t_b - t_c}{t_b - t_a}$$



Algoritmos

- Sequencial $O(n^3)$
 - Utiliza diretamente a função de mapeamento proposta por [Lacasa et al. 2008];
 - Algoritmo naïve para a criação de grafos de visibilidade

- Sequencial $O(n^2)$



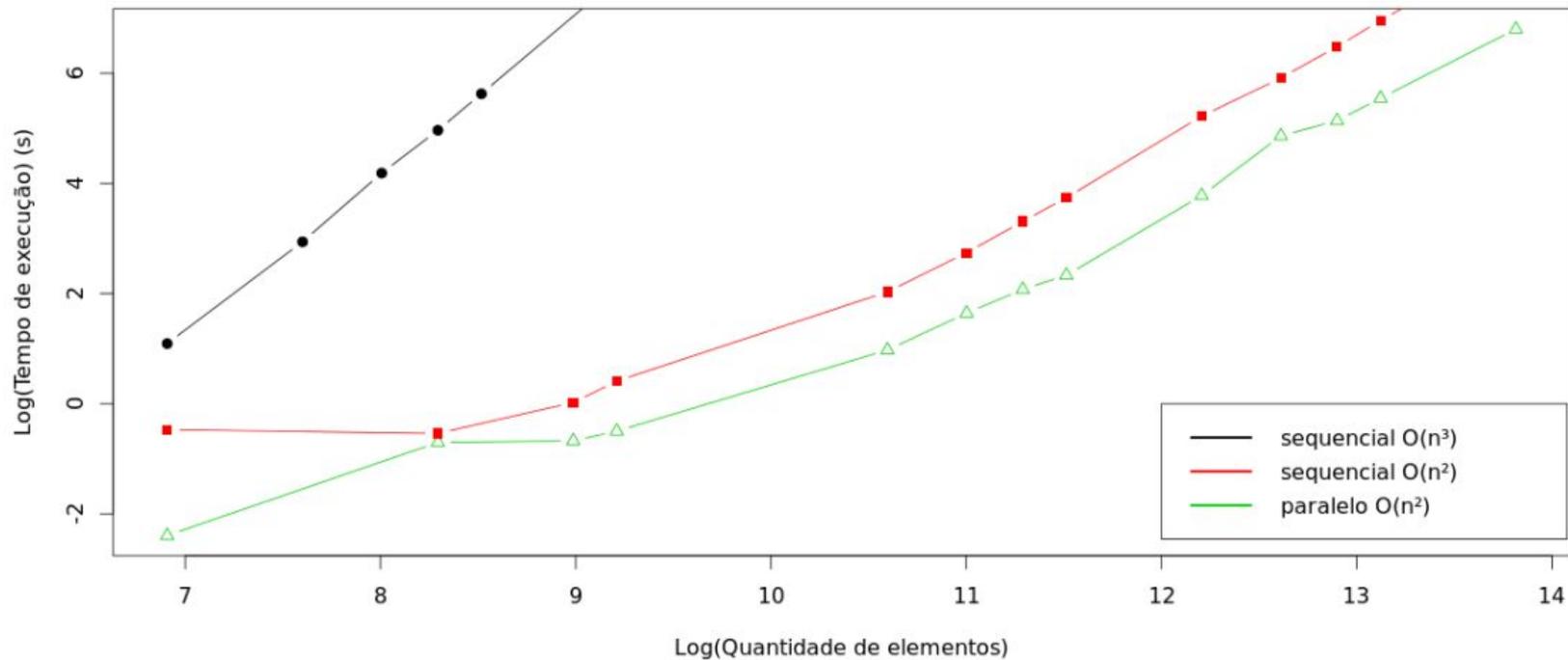
Algoritmos

- Paralelo $O(n^2)$ implementado no *Spark*

Algorithm 1 FUNÇÃO DE VERIFICAÇÃO DE VISIBILIDADE EM PARALELO

```
function Verifica_Visibilidade_Algoritmo_O2_Paralelo(a, b)
  SerieRDD.map((l_1, l_2) =>
    var arestas; var maiorElemento = 0; var valorMaiorElemento;
    serie.foreach((a_1, a_2) =>
      if ( $a_1 > l_1$ ) then
        if  $maiorElemento == 0$  then
          arestas.add(a_1); maiorElemento = a_1; valorMaiorElemento =
a_2;
        else
          if Teste - de - visibilidade then
            arestas.add(a_1); maiorElemento = a_1; valorMaiorElemento = a_2
          end if
        end if
      end if
    )
  end function
```

Resultado



Trabalho futuros

- Aplicação de grafo de visibilidade no método de caracterização estatística;
- Aplicar o algoritmo paralelo para analisar grandes volumes de dados.

Obrigado!