BULK SYNCHRONOUS PARALLEL K-MEANS IN STREAMING IMAGE DATA

Samuel Amico Fidelis

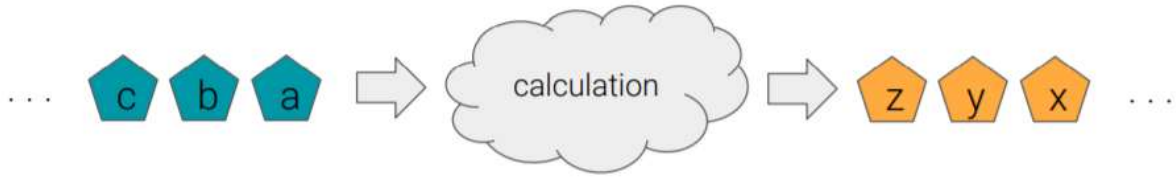
Programa de Pós Graduação em Ciência da Computação
Universidade Federal de Santa Catarina

Disciplina – Programação Paralela
Professor - Dr. Márcio Castro

Sumário

Motivação - *Streaming* de dados

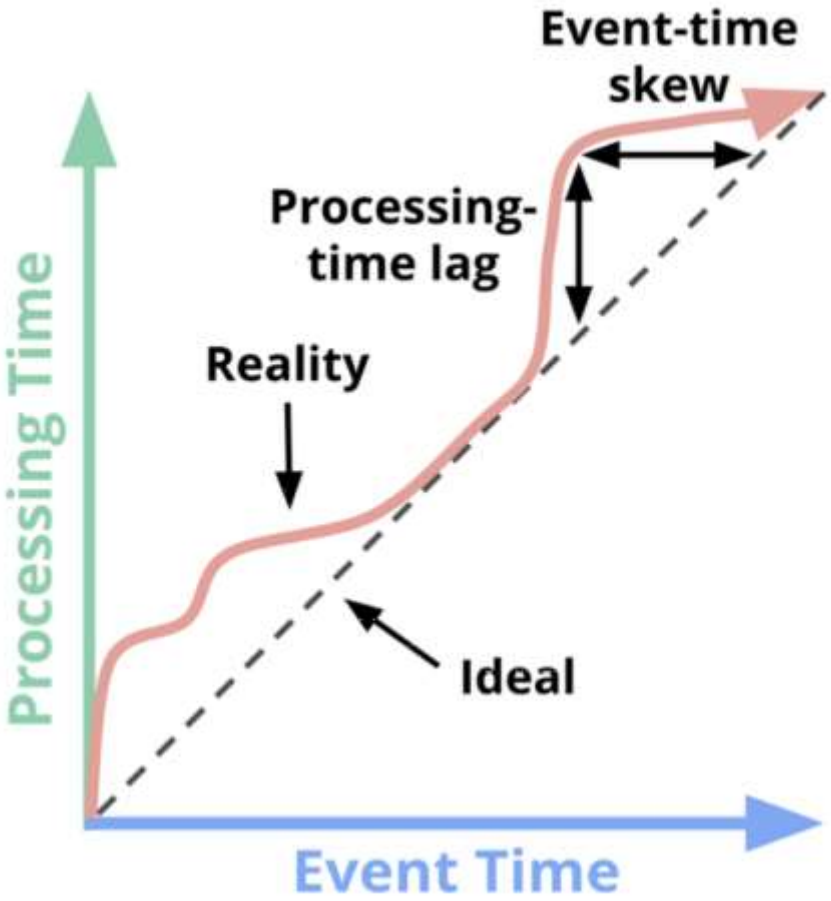
- Processamento de Streaming
- Event Time x Processing Time



- Frameworks de Streaming

Motivação - *Streaming* de dados

- Processamento de Streaming
- Event Time x Processing Time: Δ = PT - ET
- Frameworks de Streaming

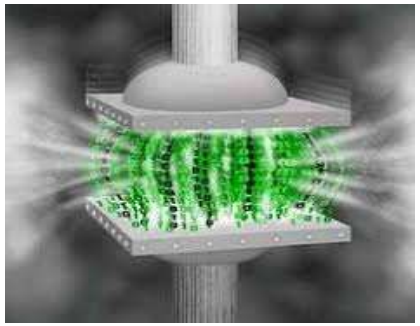


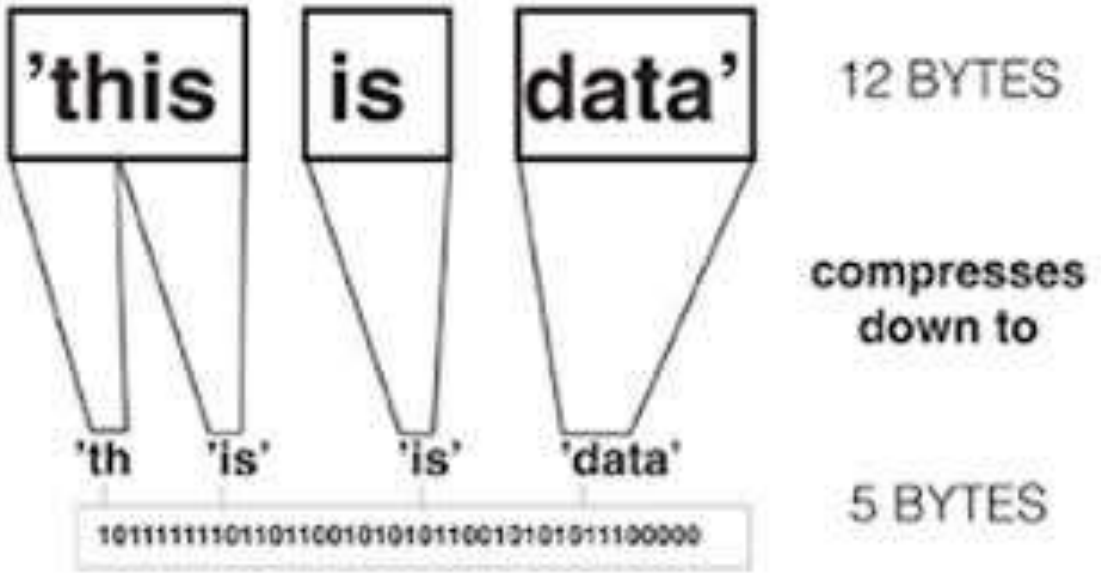
Motivação – *Streaming de dados*

- Processamento de Streaming
- Event Time x Processing Time
- Frameworks de Streaming

Motivação – Compressão de dados

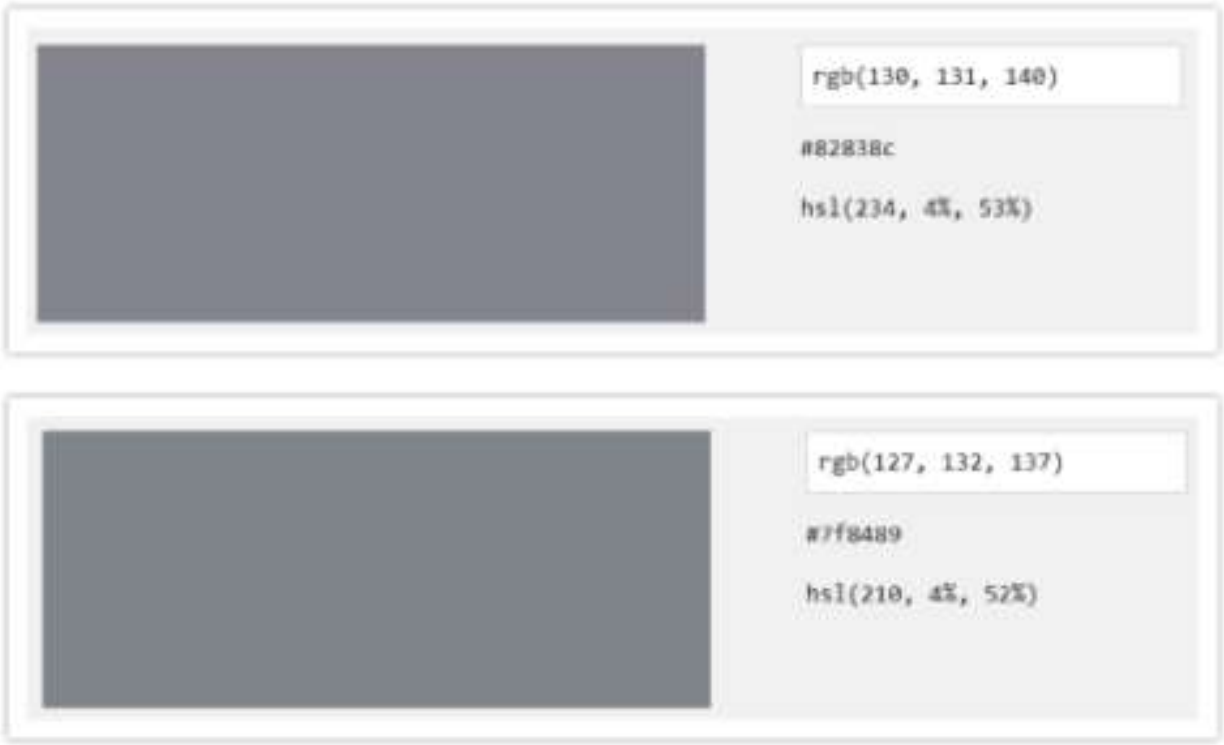
- A compressão de dados é útil para streaming ?
  - Verificar se o fluxo de dados é compactáveis
  - Se a redução do volume de dados motivado pela compressão melhora o throughput
  - O algoritmo usado tem uma baixa sobrecarga no fluxo e se é projetado para dados ilimitados.





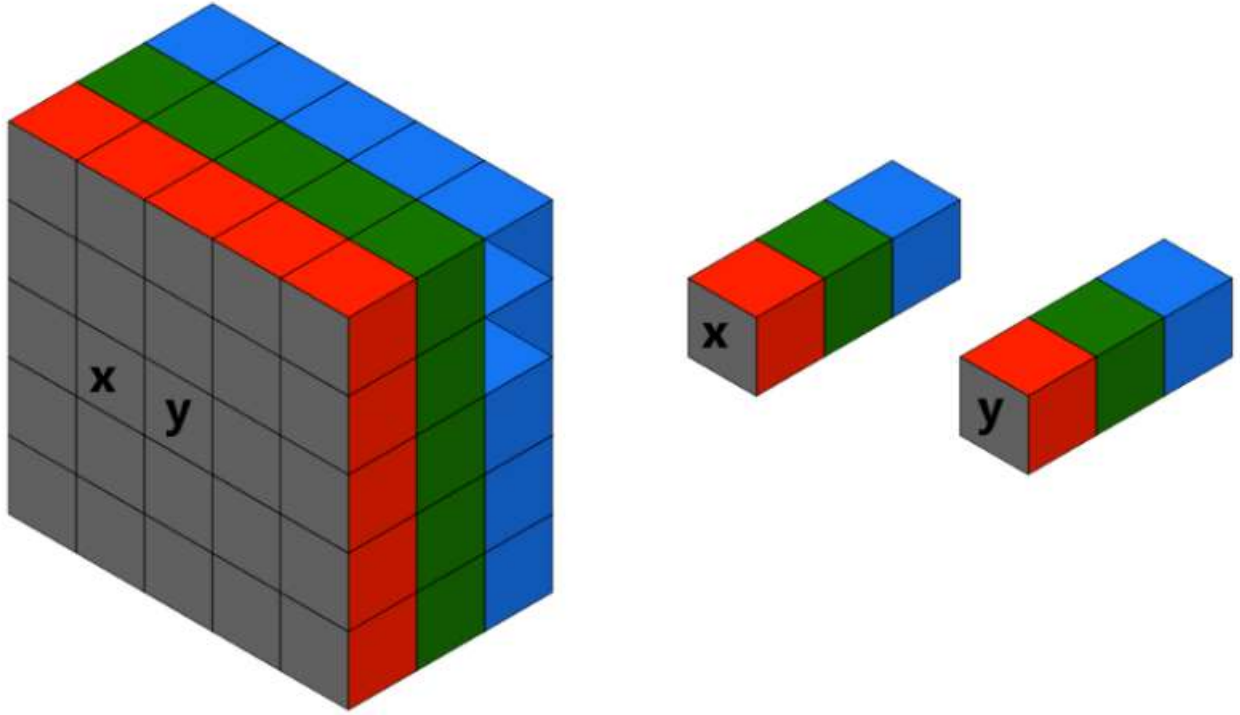
Motivação – Compressão de Imagem

- A compressão de Imagem
  - Compressão de dados em 2D
  - 3 tipos de redundância:
    - Codificação
    - Espacial e Temporal
    - Informações Irrelevantes



Motivação – Compressão de Imagem

- A compressão de Imagem
  - Compressão de dados em 2D
  - 3 tipos de redundância:
    - Codificação
    - Espacial e Temporal
    - Informações Irrelevantes

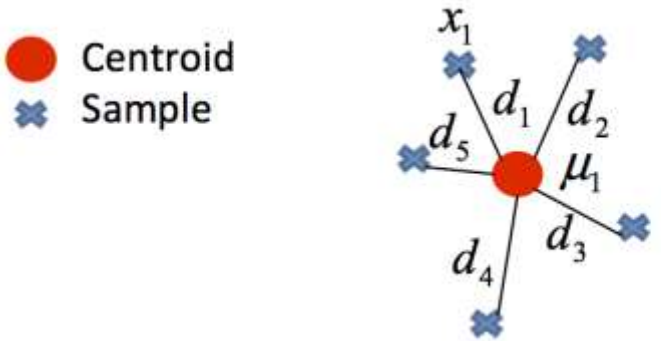


Motivação – Compressão de Imagem

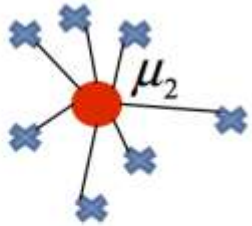
- Informações Irrelevantes
- Codificação por Transformação
- K-Means

Algoritmo – K-Means

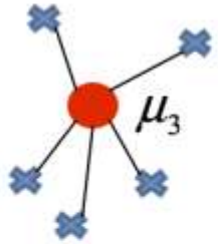
- Clustering
- Erro quadrático



$$\sum_{x_j \in S_2} d_j^2$$



$$\sum_{x_j \in S_1} d_j^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2$$



$$\sum_{x_j \in S_3} d_j^2$$

number of clusters — number of cases — centroid for cluster j — case i

objective function $\leftarrow J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$

Distance function

$$\min_S E(\mu_i) = \sum_{x_j \in S_1} d_j^2 + \sum_{x_j \in S_2} d_j^2 + \sum_{x_j \in S_3} d_j^2$$

Algoritmo – K-Means

- Clustering
- Erro quadrático

Algoritmo – K-Means

- Passo 1)

Fornecer valores para os centroides: Neste passo os k centroides devem receber valores iniciais. No início do algoritmo geralmente escolhe-se os k primeiros pontos da tabela. Também é importante colocar todos os pontos em um centroide qualquer para que o algoritmo possa iniciar seu processamento.

- Passo 2)

Gerar uma matriz de distância entre cada ponto e os centroides: Neste passo, a distância entre cada ponto e os centroides é calculada. A parte mais 'pesada' de cálculos ocorre neste passo pois se temos N pontos e k centroides teremos que calcular N x k distâncias neste passo.

- Passo 3)

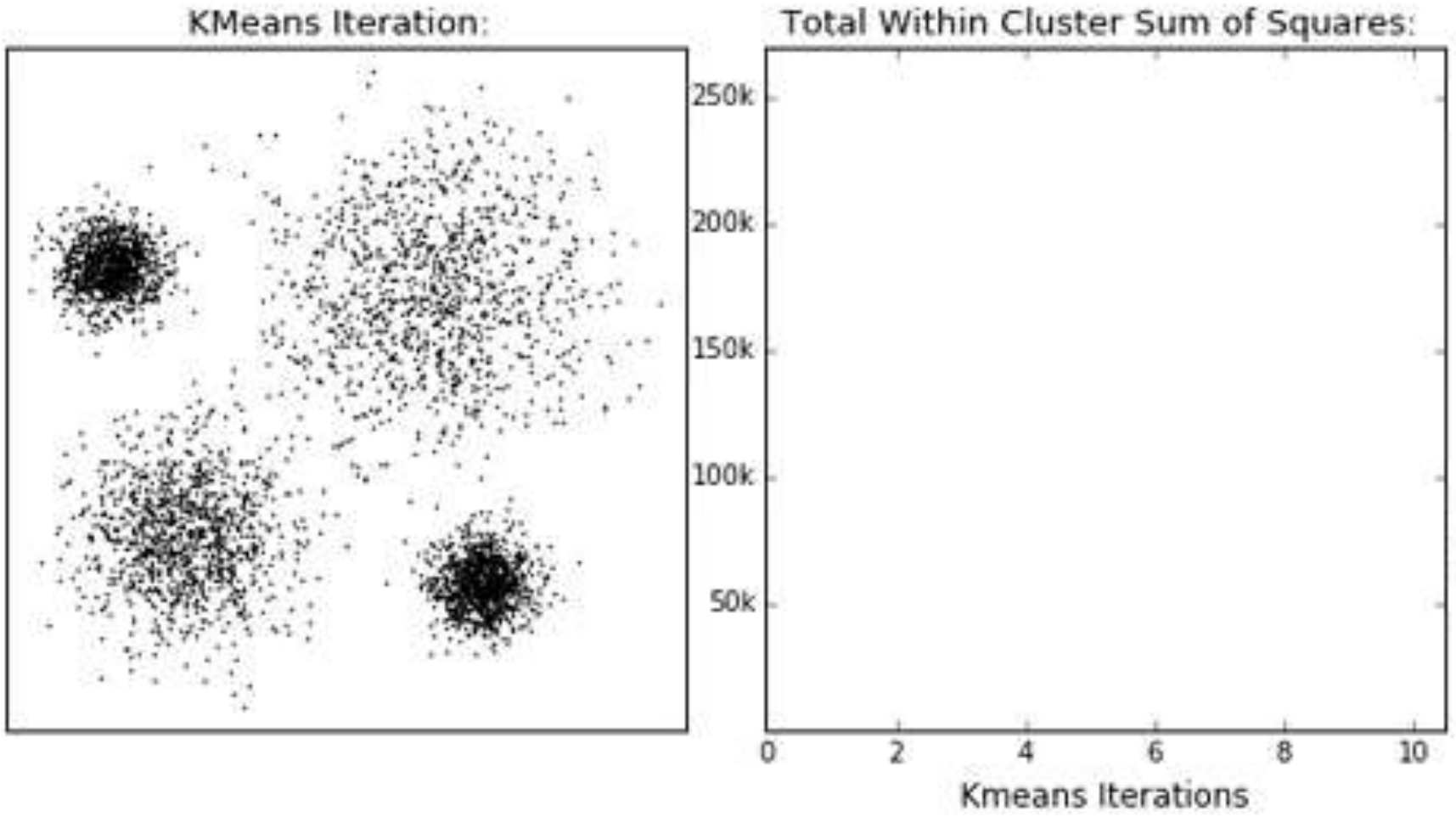
Colocar cada ponto nas classes de acordo com a sua distância do centroide da classe: aqui, os pontos são classificados de acordo com sua distância dos centroides de cada classe. A classificação funciona assim: o centroide que está mais perto deste ponto vai 'incorporá-lo', ou seja, o ponto vai pertencer à classe representada pelo centroide que está mais perto do ponto. É importante dizer que o algoritmo termina se nenhum ponto 'mudar' de classe, ou seja, se nenhum ponto for 'incorporado' a uma classe diferente da que ele estava antes deste passo.

- Passo 4)

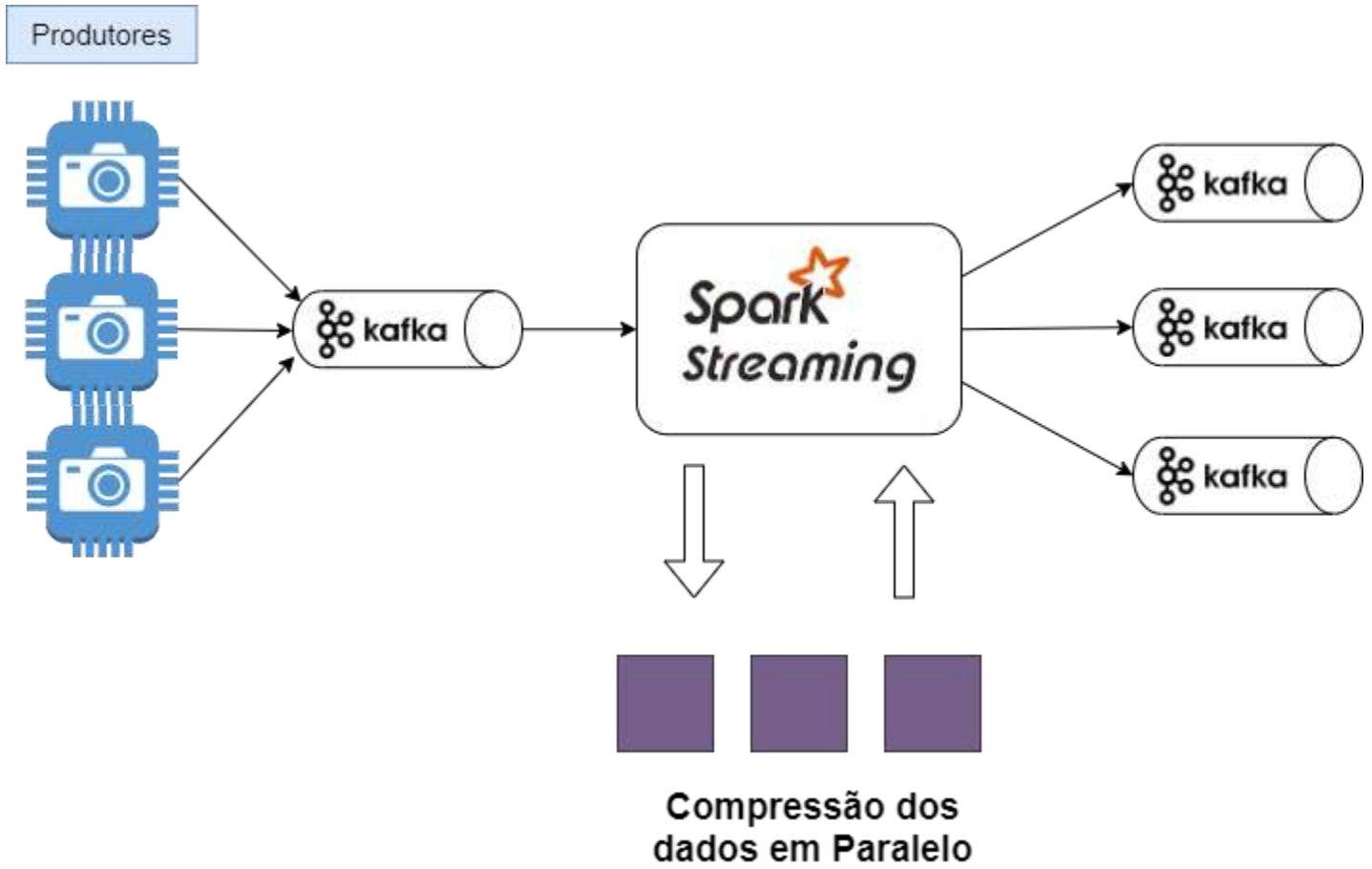
Calcular os novos centroides para cada classe: neste momento, os valores das coordenadas dos centroides são refinados. Para cada classe que possui mais de um ponto o novo valor dos centroides é calculado fazendo-se a média de cada atributo de todos os pontos que pertencem a esta classe.

Repetir até a convergência

Algoritmo – K-Means



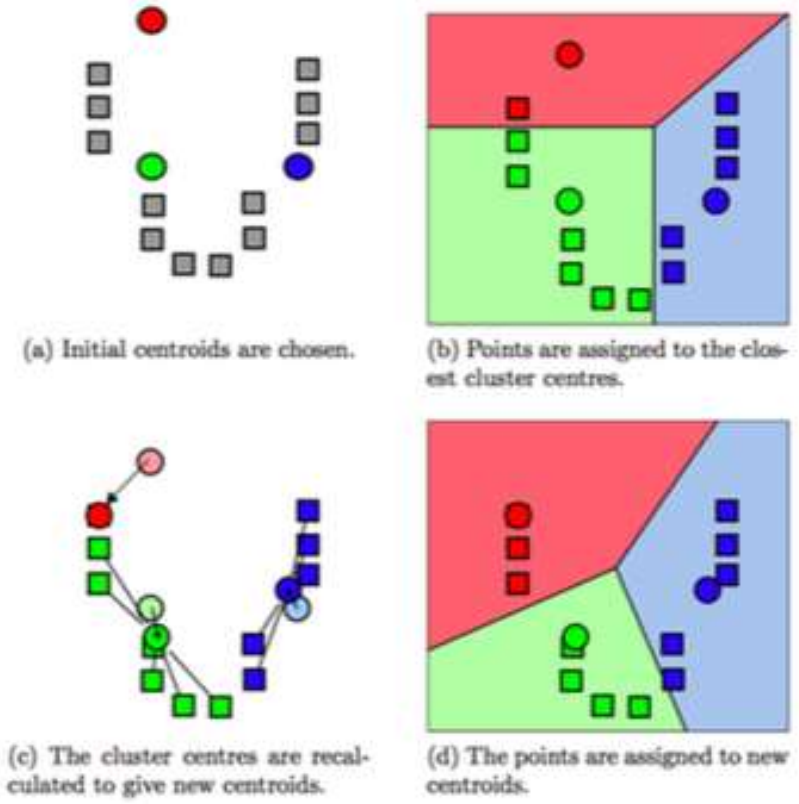
Arquitetura - Problema



Arquitetura - Problema

Paralelização

- Multiprocessadores Baseado em memória compartilhada
  - *Bulk synchronous Parallel algorithm*
  - Erro quadrático médio é calculado em paralelo e depois de todos as threads terminarem a computação, as tarefas são sincronizadas a fim de calcular novos centroides do cluster.
- Multimáquinas - MapReduce



(a) Initial centroids are chosen.

(b) Points are assigned to the closest cluster centres.

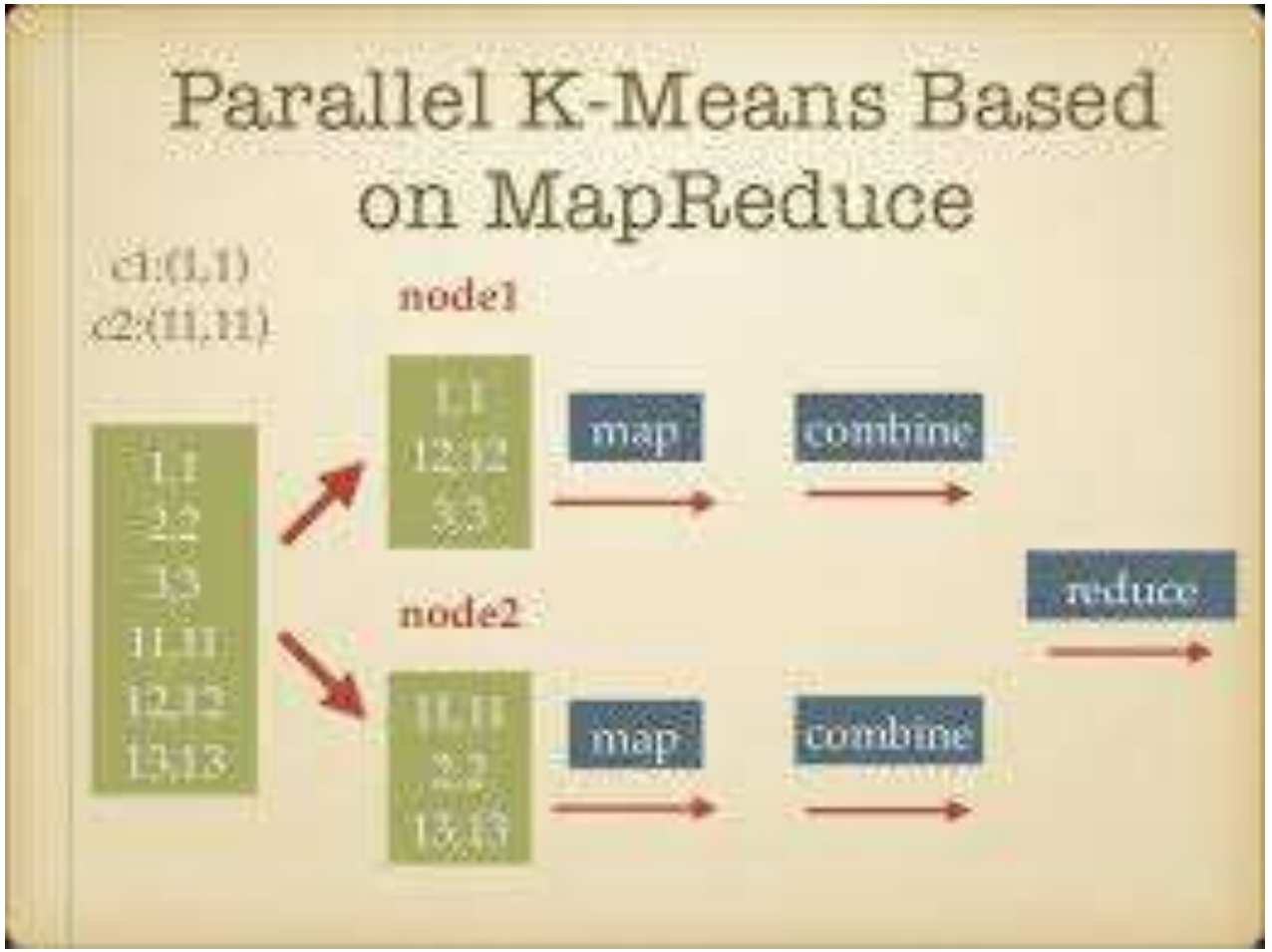
(c) The cluster centres are recalculated to give new centroids.

(d) The points are assigned to new centroids.

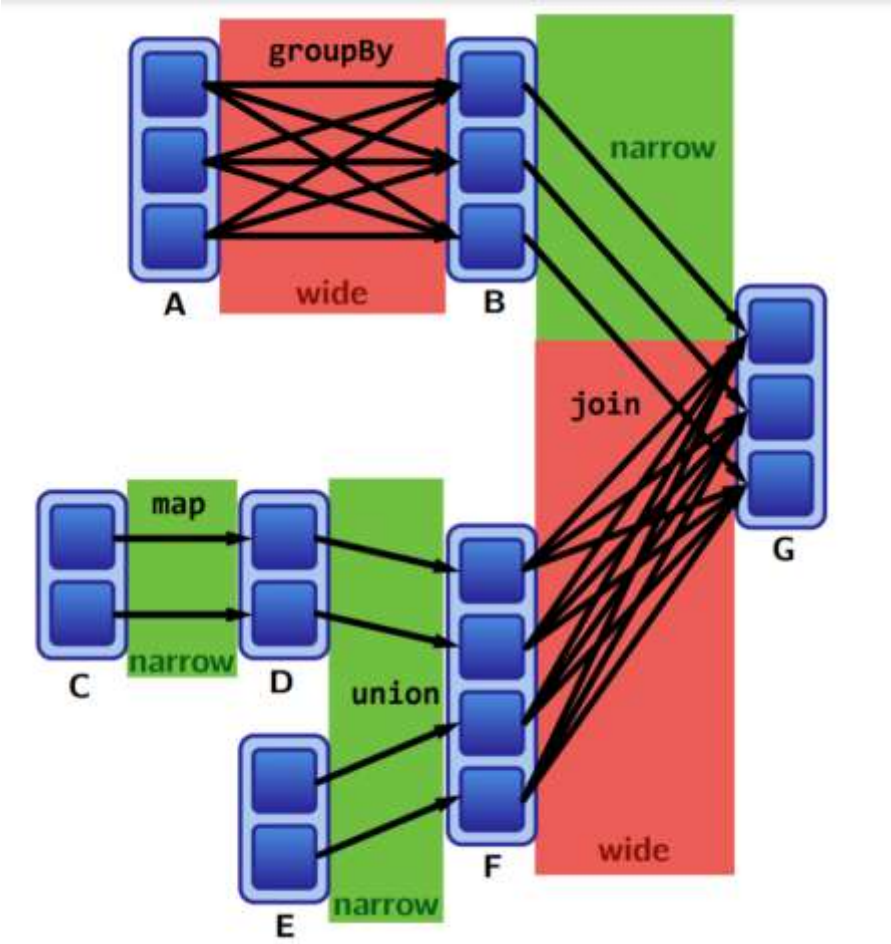
Paralelização

- Multiprocessadores Baseado em memória compartilhada

- Multimáquinas - MapReduce



Arquitetura - Problema



Paralelização

- Metodologias de Benchmarking:
  - Múltiplas Repetições
  - Tratamento estatístico – média, variância e eliminação de outliers
  - WarmUP – JavaWarmUp
  - Prevenir anomalias (GB)
  - Processing Time – Event Time

Referências

[1] LIKAS, Aristidis; VLASSIS, Nikos; VERBEEK, Jakob J. The global k-means clustering algorithm. **Pattern recognition**, v. 36, n. 2, p. 451-461, 2003.
[2] KANUNGO, Tapas et al. An efficient k-means clustering algorithm: Analysis and implementation. **IEEE transactions on pattern analysis and machine intelligence**, v. 24, n. 7, p. 881-892, 2002.
[3] DEHARIYA, Vinod Kumar; SHRIVASTAVA, Shailendra Kumar; JAIN, R. C. Clustering of image data set using k-means and fuzzy k-means algorithms. In: **2010 International Conference on Computational Intelligence and Communication Networks**. IEEE, 2010. p. 386-391.
[4] PUJARI, Arun K. **Data mining techniques**. Universities press, 2001.
[5] ALPAYDIN, Ethem. **Introduction to machine learning**. MIT press, 2020.
[6] PEKHIMENKO, Gennady et al. Tersecades: Efficient data compression in stream processing. In: **2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)**. 2018. p. 307-320.
[7] PEKHIMENKO, Gennady et al. Tersecades: Efficient data compression in stream processing. In: **2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)**. 2018. p. 307-320.
[8] BARGA, Roger S. et al. Consistent streaming through time: A vision for event stream processing. **arXiv preprint cs/0612115**, 2006.
[9] SPARK, Apache. Apache spark. **Retrieved January**, v. 17, p. 2018, 2018.
[10] GRAEFE, Goetz; SHAPIRO, Leonard D. **Data compression and database performance**. University of Colorado, Boulder, Department of Computer Science, 1990.
[11] KUCUKYILMAZ, Tayfun et al. Parallel k-means algorithm for shared memory multiprocessors. **Journal of Computer and Communications**, v. 2, n. 11, p. 15, 2014.
[12] LELEWER, Debra A.; HIRSCHBERG, Daniel S. Data compression. **ACM Computing Surveys (CSUR)**, v. 19, n. 3, p. 261-296, 1987.
[13] WELTON, Benjamin et al. Improving i/o forwarding throughput with data compression. In: **2011 IEEE International Conference on Cluster Computing**. IEEE, 2011. p. 438-445.

Referências

[14] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall, 1988
[15] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression. Boston: Kluwer Academic, 1992.
[16] M. Inaba, N. Katoh, and H. Imai, ªApplications of Weighted Voronoi Diagrams and Randomization to Variance-Based k-clustering,º Proc. 10th Ann. ACM Symp. Computational Geometry, pp. 332-339, June 1994