Trabalho Final

Pretende-se com este trabalho a utilização do modelo de programação *MapReduce* [1] para resolução de problemas computacionais de larga escala (*Big Data*). A implementação do modelo *MapReduce* é suportada na plataforma Apache Hadoop [2].

O problema a resolver fica ao critério dos alunos. No entanto, a resolução do problema deverá incluir os mecanismos e características do modelo de programação *MapReduce* suportado pela plataforma Apache Hadoop, nomeadamente:

- Utilização do sistema de ficheiros HFDS (*Hadoop Distributed File System*);
- Possibilidade de configurar a aplicação desenvolvida utilizando propriedades (especificadas na submissão da aplicação) ou através de ficheiros de configuração;
- Utilização de dados comprimidos;
- Recolha de dados estatísticos utilizando contadores;
- Utilização da cache distribuída;
- Ordenação dos resultados produzidos;
- Possibilidade de os dados (entrada e/ou saída) terem um formato próprio, o que implica o desenvolvimento de novas classes derivadas de `InputFormat` e `OutputFormat`.

Como ponto de partida para o problema a resolver podem ser considerados os seguintes casos:

- Contagem de palavras de dimensão [3] *n* (*n*-gramas) que existem num conjunto de documentos de modo a produzir informação estatística relevante:
  - Tabela de frequências dos *n*-gramas
  - Percentagem de *n*-gramas que ocorrem uma única vez (também denominados de *singletons*)
  - Cálculo da medida estatística TF-IDF [4] (*term frequency–inverse document frequency* ou frequência do termo–inverso da frequência).
- Processamento de imagem ou vídeo [5, 6]:
  - Identificação de pessoas (ou objetos)
  - Transformação do espaço de cores (das imagens ou vídeos)
- Processamento de formulários
- Extração de informação de faturas

Os alunos podem propor novos problemas. No entanto, o problema deve ser validado com o docente da Unidade Curricular.

O relatório é apresentado na forma de artigo, escrito em português ou em inglês (no formato IEEE a duas colunas), com um limite máximo de 16 páginas. Todo o código desenvolvido será incluído como apêndice do artigo (as páginas do apêndice não contam para o limite de 16 páginas do artigo).

Sugere-se que o relatório tenha a seguinte organização:

1. Resumo (*Abstract*)
   Breve apresentação do trabalho desenvolvido
2. Introdução (*Introduction*)
   Apresentação do problema que se pretende resolver
3. Trabalho Relacionado (*Related Work*)
   Apresentação de outros trabalhos relacionadas com o problema que se pretende resolver. Neste ponto podem ser indicadas/apresentadas as vantagens da solução que se vai propor
4. Implementação (*Implementation*)
   Descrição do trabalho desenvolvido
5. Resultados (*Results*)
   Discussão dos resultados obtidos
6. Conclusões e Trabalho Futuro (*Conclusions and Future Work*)
7. Bibliografia (*Bibliography*)
8. Apêndices (*Appendixes*)

Referências

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[2] A. S. Foundation. (2019, Dec.) Apache hadoop. [Online]. Available: http://hadoop.apache.org

[3] J. Lin and C. Dyer, *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers, 2010.

[4] K. Sparck Jones, "Document retrieval systems," P. Willett, Ed. London, UK, UK: Taylor Graham Publishing, 1988, ch. A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pp. 132–142. [Online]. Available: http://dl.acm.org/citation.cfm?id=106765.106782

[5] H. Tan and L. Chen, "An approach for fast and parallel video processing on apache hadoop clusters," 07 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6890135

[6] (2019, Dec.) Hipi - hadoop image processing interface. University of Virginia. [Online]. Available: https://github.com/uvagfx/hipi