



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

# Análise e processamento distribuído de redes de grande dimensão

Proposta de Projecto

Março de 2014

ORIENTADORES:

Cátia Vaz (ISEL)

Alexandre Francisco (INESC-ID/IST)

ESTUDANTES:

André Mota

aqmota@gmail.com

912209726

Aguinaldo Pontes

pontesaguinaldo15@gmail.com

925102870

## Introdução

Nos últimos anos, o processamento de grandes quantidades de dados tem sido um tópico de grande interesse. Contudo, a análise das estruturas envolvidas no processo é normalmente complexa. De modo a diminuir a complexidade envolvida no tratamento deste tipo de estruturas surgiram algumas plataformas seguindo o modelo *Map Reduce*, como o Apache Hadoop[5].

O Apache Hadoop[5] é uma plataforma que visa facilitar o processamento e análise de estruturas de dimensões consideráveis em ambientes distribuídos, a qual tem sido muito utilizada. A plataforma oferece um conjunto de benefícios tais como a sua interface simples de programação, escalabilidade e de ser tolerante a falhas. Esta plataforma é composta por quatro módulos: Hadoop Common (conjunto de ferramentas que servem de suporte a outros módulos), Hadoop Distributed File System (sistema de ficheiros distribuído), Hadoop Yarn (plataforma que disponibiliza o agendamento de tarefas) e o Hadoop Map Reduce (módulo que usa o modelo de programação Map Reduce para o processamento de dados e agenda tarefas usando o Hadoop Yarn)

Apesar dos benefícios de se usar este tipo de plataforma para alguns tipos de dados, e ser possível a sua utilização para o processamento de redes através múltiplas invocações de Map Reduce, o modelo de programação usado não é o mais adequado para o processamento de grafos devido à existência de uma elevada complexidade envolvida na implementação de algoritmos e um custo computacional indesejado. Para resolver este problema foi proposto pela Google uma plataforma, denominada Pregel[8], que se baseia no modelo de programação Bulk Synchronous

Parallel[12].

Baseando-se na implementação da Google foram surgindo implementações *open-source* como o GPS[6], Apache Hama[3] e Apache Giraph[4]. Estas plataformas exportam uma interface programável com algumas semelhanças assim como uma típica computação de um grafo, em que consiste começar por iniciar o respectivo grafo seguido de um número variável de *supersteps* (iterações) até que todos os vértices estejam inactivos (não têm que participar na computação). Durante cada *superstep* é chamada (paralelamente) para cada vértice do grafo uma função definida pelo utilizador que irá delinear o seu comportamento. Durante o processamento de um vértice, tem-se acesso às mensagens que lhe foram enviadas no *superstep* anterior, sendo também possível enviar mensagens (que irão ser recebidas do próximo *superstep*) para outros vértices que se conheça o seu identificador único (tipicamente vértices vizinhos). Este modelo tem uma barreira de sincronização entre *supersteps*, que faz com que cada um só se inicie após todos os *nodes* entrarem na barreira de sincronização, fazendo com que a performance global seja afectada pelo *node* que demore mais a processar. De qualquer modo, o modelo simplifica a semântica da implementação dos algoritmos e tem normalmente um melhor desempenho que as implementações em Map Reduce devido à facilidade em que há em partilhar o estado entre os vários vértices.

Apesar de existirem alguns algoritmos implementados nos ambientes descritos anteriormente, o objectivo deste projecto é analisar as plataformas baseadas no modelo *Bulk Synchronous Parallel*, de modo a criar uma biblioteca modular que contenha um conjunto de algoritmos e que consiga ser usada em diversas plataformas. As principais plataformas que serão estudadas para o desenvolvimento desta biblioteca será o Apache Hama e o Apache Giraph. O Apache Giraph é uma plataforma de interesse tendo em conta que usa como base o Apache Hadoop para o agendamento de tarefas, o que facilita o seu uso para todas as infraestruturas que usam Apache Hadoop. O Apache Hama proporciona um modelo mais perto do *Bulk Synchronous Parallel* que o Apache Giraph, daí também ser alvo de estudo para esta biblioteca.

## Requisitos

Os requisitos obrigatórios a realizar no âmbito deste projecto serão:

1. Módulos de suporte às plataformas Apache Hama e Apache Giraph
2. Implementação de um conjunto de algoritmos:
  - (a) *heat-kernel*[10]
  - (b) *k-core*[11]
  - (c) *Breadth First Search*[13]

- (d) *Louvain*[1]
  - (e) *Layered Label Propagation*[2]
  - (f) *Betweenness Centrality*[9]
3. Produzir documentação sobre a biblioteca.
  4. Conjunto de testes aos algoritmos implementados.
  5. Análise de performance comparativamente a outras plataformas/bibliotecas utilizando *clusters* disponibilizados pelo INESC.

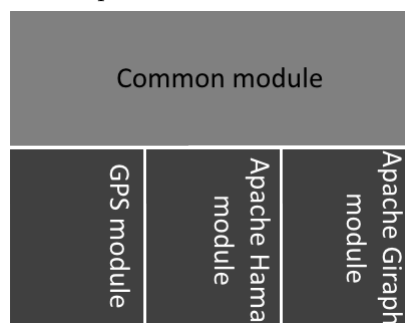
Os requisitos opcionais que serão realizados caso os requisitos obrigatórios serem conseguidos em tempo útil são:

1. Módulo de suporte à plataforma académica GPS.
2. Implementação de algoritmos a definir.

## Arquitectura

A biblioteca vai seguir um modelo de programação modular em que cada módulo irá ter uma responsabilidade diferente. Existirá um modulo comum onde estarão implementados os algoritmos e onde estará definida uma interface que estandardize as interfaces disponibilizadas pelas diversas plataformas. Esta interface estandardizada e comum a todos os módulos terá o objectivo de permitir a implementação dos algoritmos de forma independente da plataforma. Para cada plataforma estudada (Apache Hama, Apache Giraph e GPS) será efectuado um módulo cuja função é a de mapear a interface da respectiva plataforma para a interface estandardizada.

Figura 1: Arquitectura modular da biblioteca.



## Calendarização

O planeamento do projecto é o seguinte:

Tabela 1: Calendarização semanal

Data de Inicio	Semana	Descrição
2 Março	1	Escrita da Proposta
9 Março	2	Finalização da Proposta e iniciação do estudo das Plataformas
16 Março	3	Estudo das Plataformas e levantamento das interfaces programáveis.
23 Março	4	Estudo dos algoritmos <i>k-core</i> , <i>heat kernel</i> e <i>BFS</i>
30 Março	5-6	Estudo dos algoritmos <i>Layered Label Propagation</i> e <i>Louvain</i> . Estruturar os vários módulos da biblioteca.
13 Abril	7-8	Implementação dos algoritmos estudados na 4 <sup>a</sup> semana.
27 Abril	9-10	Preparação da apresentação individual e relatório de progresso
11 Maio	11-12	Implementação dos algoritmos <i>Louvain</i> e <i>Layered Label Propagation</i> .
25 Maio	13-14	Estudar e implementar o algoritmo de <i>Betweenness Centrality</i> .
8 Junho	15	Cartaz e finalização da versão com a implementação dos algoritmos obrigatórios.
15 Junho	16-19	Escolha dos <i>data-sets</i> (que iram ser usados nos testes), testes e comparações com outras plataformas.
13 Julho	20	Finalização do relatório e entrega da versão final.

O relatório irá ser realizado de forma gradual, havendo contribuições em todas as semanas.

## Referências

- [1] Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008): P10008.
- [2] Boldi, Paolo, et al. "Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks." *Proceedings of the 20th international conference on World Wide Web*. ACM, 2011.
- [3] Apache Hama. <https://hama.apache.org/>.
- [4] Apache Giraph. <https://giraph.apache.org/>.
- [5] Apache Hadoop. <http://hadoop.apache.org/>.
- [6] Salihoglu, Semih, and Jennifer Widom. "Gps: A graph processing system." *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*. ACM, 2013.
- [7] Redekopp, Mark, Yogesh Simmhan, and Viktor K. Prasanna. "Optimizations and Analysis of BSP Graph Processing Models on Public Clouds." *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*. IEEE, 2013.
- [8] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010.
- [9] David A. Bader, Shiva Kintali, Kamesh Madduri, and Milena Mihail. Approximating betweenness centrality. In *Proc. 5th Workshop on Algorithms and Models for the Web Graph*, pages 124-137, 2007.
- [10] Chung, Fan. "The heat kernel as the pagerank of a graph." *Proceedings of the National Academy of Sciences* 104.50 (2007): 19735-19740.
- [11] Fortunato, Santo, and Marc Barthélemy. "Resolution limit in community detection." *Proceedings of the National Academy of Sciences* 104.1 (2007): 36-41.
- [12] Valiant, Leslie G. "A bridging model for parallel computation." *Communications of the ACM* 33.8 (1990): 103-111.
- [13] Edward F. Moore. The shortest path through a maze. In *Proceedings of the International Symposium on the Theory of Switching*, pages 285-292. Harvard University Press, 1959.