

X ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

Um Modelo para Avaliação de Desempenho de Plataformas de Processamento de Grafos

Daniel N. R. da Silva¹ (dramos@lncc.br)

Klaus Wehmuth¹ (klaus@lncc.br)

Carla Osthoff¹ (osthoff@lncc.br)

Ana Paula Appel² (apappel@br.ibm.com)

Artur Ziviani¹ (ziviani@lncc.br)

¹ Laboratório Nacional de Computação Científica (LNCC)

Av. Getúlio Vargas, 333, Quitandinha

CEP 25651-075 – Petrópolis, RJ – Brasil

² IBM Research Brazil

R. Tutóia, 1157, Vila Mariana

CEP 04007-900 – São Paulo, SP – Brasil

Resumo. *Redes complexas, i.e. grafos cujas características topológicas não são nem puramente aleatórias nem puramente regulares, estão presentes em diversas áreas do conhecimento, p. ex., biologia, economia, química, sociologia e tecnologia. Devido à relevância das análises desses grafos, muitas plataformas destinadas ao seu processamento têm surgido nos últimos anos. Entretanto, a grande quantidade dessas plataformas torna difícil a tarefa dos interessados em escolher aquela mais apropriada ao seu tipo de análise e rede. Soma-se ainda a essa dificuldade a dimensão que alguns desses grafos pode alcançar, o que pode tornar imprescindível o uso de processamento tanto paralelo quanto distribuído. Desse modo, compreender as implicações no desempenho computacional ao serem utilizadas diferentes abordagens no processamento de grafos é um tema de pesquisa pertinente. Neste trabalho, propõe-se um modelo de avaliação de desempenho de plataformas para processamento de grafos que visa atender quatro aspectos chave na compreensão do problema: algoritmos, arquiteturas, plataformas e redes.*

Palavras-chave: Avaliação de desempenho, processamento de grafos, redes complexas.

1 Introdução

Muitas estruturas e fenômenos podem ser descritos através do conceito de redes: entidades que estão ou não conectadas. A Teoria dos Grafos utiliza o conceito de grafo G — um par ordenado (V, E) no qual V é o conjunto de vértices (entidades) e E o de arestas (relacionamentos) — para modelar essas estruturas. Nesse cenário, redes cujos padrões de conectividade não são nem puramente regulares, tampouco exclusivamente aleatórios — ditas complexas — apresentam-se

em diversas áreas do conhecimento, p. ex., biologia, química, sociologia e tecnologia (Newman, 2010). A complexidade nos padrões de ligação e dificuldades relacionadas à escala que essas redes atualmente podem alcançar (até bilhões de componentes) impõem desafios ao seu processamento computacional eficiente (Lovász et al., 2009). Assim, estudos que avaliem o processamento de grafos de larga escala possuem relevância para academia e indústria.

O desafio de compreender o processamento de redes complexas de larga escala envolve o entendimento das relações e implicações de 4 aspectos: (i) **arquitetura**: quais implicações no processamento provém das tecnologias de *hardware* de determinado sistema computacional, p. ex., distribuição de memória, tecnologia de armazenamento e rede. (ii) **algoritmo**: que interferências no processamento provém dos tipos de análises e algoritmos. (iii) **plataforma**: em que influí as características de determinada plataforma de *software*, p. ex., estrutura de dados, modelo de programação e otimizações. (iv) **rede**: em que as características das redes, p. ex., grau de agrupamento, distribuição dos vértices e diâmetro, afetam substancialmente o desempenho do processamento. Pode-se perceber que as dimensões são inter-relacionadas, sendo não somente o desempenho final do processo dependente delas, mas elas próprias interdependentes (Fig. 1).

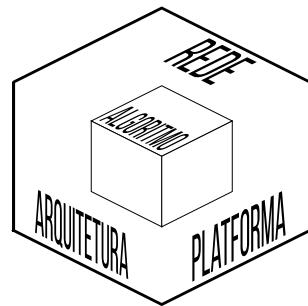


Figura 1: 4 aspectos chave no processamento de grafos.

A contribuição deste artigo é um modelo de avaliação de desempenho do processamento de grafos que ataca o desafio relatado nos 4 aspectos citados. É apresentada a metodologia de escolha de algoritmos, arquitetura, plataformas e redes, estabelecendo diretrizes para experimentos futuros.

2 Trabalhos Relacionados

Uma grande quantidade de plataformas para o processamento de grafos têm surgido no últimos anos. Em um trabalho recente, são revisadas mais de oitenta delas (Doekemeijer e Varbanescu, 2014). Devido à grande quantidade dessas plataformas, observa-se uma crescente na publicação de estudos de avaliação de desempenho de plataformas para o processamento de grafos (Fig. 2).

Dentre os estudos de avaliação de desempenho, foi proposta uma metodologia de avaliação que contempla os 4 aspectos citados na Seção 1 que culminou no *benchmark Graphalytics* (Iosup et al., 2016). Entretanto, no panorama de plataformas para grafos, grande parte dos estudos de avaliação de desempenho não contempla de modo diversificado os 4 aspectos do problema (Silva et al., 2016). Assim, é substancial a abordagem do domínio do processamento de grafos com maior diversidade a fim de que maior contribuição possa ser gerada aos interessados.

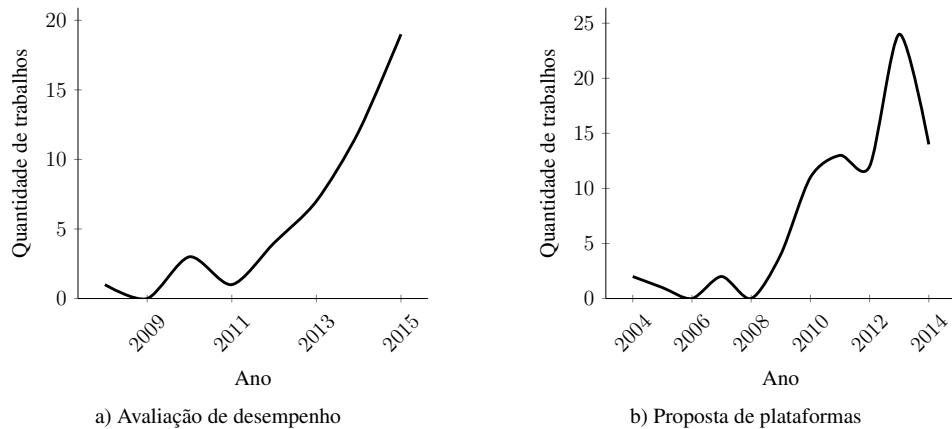


Figura 2: Crescimento na publicação de estudos de a) avaliação de desempenho e b) proposta de plataformas para processamento de grafos.

3 Proposta de Modelo

O objeto de estudo deste trabalho é o comportamento computacional da tarefa de processamento de grafos. Portanto, em uma visão simplificada, define-se tal objeto como um espaço 4-dimensional: **algoritmo, arquitetura, plataforma e rede**. Ao avaliar tal objeto de estudo é essencial que sejam escolhidos pontos que representem de maneira adequada cada aspecto, no intuito de que a análise não se dê de forma tendenciosa. Para esse fim, criteriosamente escolheu-se representantes de cada aspecto, conforme descrito posteriormente nesta seção, visando-se a diversidade de características, assim como, equilíbrio entre abstração e realidade.

Além disso, o presente trabalho lança mão à técnica de medição de desempenho em contraste à modelagem analítica e por simulação. Essa técnica foi escolhida uma vez que geralmente provê os melhores resultados (Lilja, 2005). Desse modo, dados os conjuntos de A_l (algoritmos), A_r (arquiteturas computacionais), P (plataformas) e R (redes), para cada execução de determinado algoritmo em uma plataforma em um ambiente para certa rede $e_i \in E = A_l \times A_r \times P \times R$ avaliar-se-á cada $m_j \in M$ (métricas). A técnica *Multi Way Analysis of Variance* é a ferramenta de avaliação estatística do conjunto de execuções. Os conjuntos serão descritos a seguir.

3.1 Algoritmos

Um grande número de algoritmos para grafos têm sido implementados para as mais diversas análises, os quais segundo (Guo et al., 2014) podem ser divididos nas classes: *componentes conexos, detecção de comunidades, estatística, evolução de grafos, travessia e outros*. Além disso, esses algoritmos usualmente são utilizados em análises maiores que podem ser classificadas nas categorias: *caminho, centralidade, comunidade e conectividade*.

Nesse cenário, em (Kalavri, Vlassov e Haridi, 2016) são revisados estudos de modelos de programação e avaliação de desempenho de processamento de grafos, no quais os três algoritmos mais avaliados são *Page Rank*, Caminhos mínimos, Componentes conexos e variações desses três algoritmos. Portanto, baseando-se no princípio da diversidade de classes de algoritmos, em trabalhos relacionados de avaliação e na disponibilidade dos mesmos nas plataformas escolhidas para realização de experimentos em trabalhos futuros, foram escolhidos os algoritmos de Busca em Largura, Componentes Conexos, Contagem de Triângulos e *Page Rank* (Tab. 1).

Tabela 1: Algoritmos escolhidos.

| Algoritmo | Tipo de análise | Categoria | CPU | Memória | Rede |
|------------------------|-----------------|---------------------|-------|---------|-------|
| Busca em Largura | Caminho | Travessia | Baixo | Baixo | Baixo |
| Componentes Conexos | Conectividade | Componentes conexos | Baixo | Médio | Médio |
| Contagem de triângulos | Comunidade | Estatística geral | Alto | Alto | Médio |
| Page Rank | Centralidade | Outros | Médio | Médio | Alto |

Busca em Largura: É um algoritmo fundamental para construção de algoritmos mais complexos na análise de redes. Dado um vértice origem $v \in V$, esse algoritmo computa a distância de v aos demais vértices $u \in V$.

Componentes Conexos: É um algoritmo utilizado na análise de conectividade das redes. Tem por intuito encontrar os componentes conexos de um grafo G , isto é, subgrafos $G_i \in G$, de modo que exista um caminho p entre cada par de vértice $u, v \in V(G_i)$.

Contagem de Triângulos: Define-se um triângulo em um grafo G três vértices $u \neq v \neq w \in V | (u, v) \in E \wedge (u, w) \in E \wedge (v, w) \in E$. A quantidade de triângulos em G é uma estatística concernente ao grau de agrupamento do mesmo.

Page Rank: É um algoritmo utilizado em análises de centralidade que visar medir a importância de um nó baseado na importância de seus vizinhos. Esse algoritmo ganhou notoriedade tanto na academia quanto na indústria uma vez que o motor de busca do Google baseia-se nele.

3.2 Arquitetura

Segundo a disponibilidade de recursos dos autores, foi definida a realização dos experimentos em um *cluster* de dois nós computacionais, cada um com 2 Intel Xeon E5-2620V3 2.4GHz 6-core, 64GB DDR4 2133MHz, 3 HDD SAS 6Gb/s 4TB 7.2Krpm e Gigabit on-board. Uma vez que o ambiente é modesto, o critério de diversidade adotado se encontra na variação de núcleos de processamento (1, 6 e 12) e nós utilizados (1 e 2) para fins de experimentos de escalabilidade.

3.3 Métricas de Desempenho

As escolhas das métricas de avaliação de desempenho levaram em consideração os princípios de linearidade, confiabilidade, repetibilidade, facilidade de medição, consistência e independência (Lilja, 2005). Desse modo, foram selecionadas 17 métricas de desempenho divididas nas seguintes categorias: a) tempo de resposta, b) utilização, c) eficiência, e d) vazão (Tab. 2).

3.4 Plataformas

Um dos grandes desafios na avaliação de desempenho de plataformas destinadas ao processamento de grandes grafos é a escolha de um conjunto representativo das mesmas. Levando em consideração trabalhos na literatura que analisam o estado da arte desse domínio (Doekemeijer e Varbanescu, 2014; Elshawi et al., 2015; Pokorny, 2015; Yan et al., 2016), naqueles que propõem sistemas para o processamento de grafos, assim como nos que avaliam o desempenho dessas sistemas, foi escolhido um conjunto diversificado de plataformas (Tab. 3).

Tabela 2: Métricas de Desempenho escolhidas.

| Métrica | Descrição |
|--------------------------|--|
| <i>Tempo de Resposta</i> | Tempo de parede (<i>TT</i>) |
| | Tempo de algoritmo (<i>TC</i>) |
| | Tempo de carga (<i>TR</i>) |
| | Tempo de escrita (<i>TW</i>) |
| | $TT - (TC + TR + TW)$ |
| <i>Utilização</i> | Uso de CPU (<i>RC</i>) |
| | Uso de memória (<i>RM</i>) |
| | Uso de rede (<i>RN</i>) |
| | Uso de disco (<i>RD</i>) |
| | Eficiência energética (<i>RE</i>) |
| <i>Eficiência</i> | Escalonamento forte (<i>SS</i>) |
| | Escalonamento fraco (<i>WS</i>) |
| | Escalonamento horizontal (<i>HS</i>) |
| | Escalonamento vertical (<i>VS</i>) |
| | <i>Speedup</i> (<i>S</i>) |
| <i>Vazão</i> | Eficiência paralela (<i>E</i>) |
| | Arestas por segundo (<i>EPS</i>) |
| | $ E /TT$. |
| | Vértices por segundo (<i>VPS</i>) |
| | $ V /TT$. |

Tabela 3: Plataformas escolhidas.

| Plataforma | Breve descrição e principais características |
|---------------------|---|
| Apache Giraph | <i>Framework</i> para o processamento de grafos baseado no paradigma centrado em vértice construído em cima do Apache Hadoop. Paradigma de programação centrado em vértice, implementado em Java, suporta computação em memória externa, sistema para memória distribuída, código livre. |
| Chaos | Sistema de processamento de grafos projetado para análise de grandes grafos utilizando <i>clusters</i> pequenos. Paradigma de programação centrado em aresta, implementado em C++, suporta computação em memória externa, sistema para memória distribuída, código livre. |
| GraphLab Create 2.0 | <i>Framework</i> extensível para análises que utilizam aprendizagem de máquina (possui um <i>toolkit</i> para análise de grafos) que evoluiu a partir de PowerGraph. Paradigma de programação GAS (<i>gather, apply and scatter</i>), implementado em C++ com ligações para Python, sistema para memória distribuída, proprietário. |
| GraphMat | <i>Framework</i> para o processamento de grafos baseado na tradução do paradigma orientado a vértice em multiplicação generalizada entre matriz esparsa e vetor denso. Implementado em C++, sistema para memória compartilhada e distribuída, código livre. |
| GraphX | Uma API para processamento de grafos do <i>framework</i> Apache Spark. Paradigma de programação centrado em vértice, sistema para memória distribuída, código livre. |
| GridGraph | Sistema para o processamento de grandes grafos em apenas uma máquina baseado na nova representação de grafos em grade. Modelo de processamento <i>Streaming - Apply</i> , suporta processamento em disco, implementado em C++, código livre. |
| NetworkKit | Biblioteca de alto desempenho destinada à análise de redes complexas de larga escala. Implementada em C++ com <i>bindings</i> para Python, sistema para memória compartilhada, código livre. |

3.5 Redes

Definiu-se um conjunto de dados de 28 redes, sendo 14 traços de redes reais e 14 redes sintéticas. Ao mesclar dados reais e sintéticos, o poder de análise cresce já que se tem maior equilíbrio entre controle de parâmetros (dados sintéticos) e representatividade (dados reais).

Tabela 4: Mapeamento do conjunto de dados em classes com a escala $s(V, E) = \log(|V| + |E|)$.

| Escala | < 7 | [7, 7.5) | [7.5, 8) | [8, 8.5) | [8.5, 9) | [9, 9.5) | ≥ 9.5 |
|--------|-----|----------|----------|----------|----------|----------|------------|
| Classe | 2XS | XS | S | M | L | XL | 2XL |

Dimensão das redes: Em (Iosup et al., 2016) define-se a escala de um grafo G como $s(V, E) = \log(|V| + |E|)$, mapeando-se s em 7 categorias (Tab. 4). Para cada categoria foram escolhidos 2 traços de redes reais e gerados 2 grafos sintéticos segundo os critérios a seguir.

Traços de redes reais: Coletou-se um grande conjunto de redes dos repositórios¹ efetuando-se pré processamento através da: remoção de redes duplicadas, mantida apenas a topologia da rede, converteu-se redes direcionados em não direcionados, redes com $s < 6$ e $s > 10$ foram desconsideradas, e efetuada computação de estatísticas básicas das redes: a) número de vértices e arestas b) vértices de grau mais alto e mais baixo, c) graus menor, maior e médio, d) contagem de padrões de garra e cunha, e) exponente de lei de potência e f) Veto de características obtido através da aplicação do método *Degree Distribution Quantification and Comparison* (Aliakbary, Habibi e Movaghbar, 2015). Para cada classe em s foi criado um espaço de características com as estatísticas sendo escolhidas 2 redes mais distantes entre si pela distância de Manhattan.

Redes sintéticas: Para geração das redes sintéticas, adotou-se um modelo generativo de rede pertencente à família RHG (*Random Hyperbolic Graphs*) (Krioukov et al., 2010), o TRHG (*Threshold Random Hyperbolic Graphs*) (Looz et al., 2016). A escolha desse modelo baseou-se no fato de que a família de Grafos Aleatórios Hiperbólicos captura características presentes em redes reais, tais como, distribuição de grau seguindo uma lei de potência e forte efeito de comunidade. Esse modelo permite a manipulação de três parâmetros n (número de vértices), γ (exponente de lei de potência relacionado a distribuição do grau dos vértices) e k (grau médio esperado). Para n optou-se pelos valores $10 \times 10^4, 10 \times 10^5, 20 \times 10^5, 10 \times 10^6, 20 \times 10^6, 10 \times 10^7, 20 \times 10^7$. Além disso, para todas as redes utilizou-se o grau médio $k = 25$. Em relação a γ adotou-se 2 valores 2.2 e ∞ . Sendo que para ∞ , como TRHG degenera para o modelo de grafo aleatório Erdos Renyii (ER) (Erdos e Renyi, 1959) segundo (Aldecoa, Orsini e Krioukov, 2015), foi adotado o modelo ER com o parâmetro p ajustado para se ter $k = 25$.

4 Considerações finais

Este artigo define um modelo de avaliação de desempenho para plataformas de processamento de grafos considerando quatro aspectos chave: algoritmos, arquiteturas, plataformas e redes. Ao abordar esse problema atentando para os aspectos citados, assim como a escolha criteriosa dos representantes de cada um deles, leva-se em conta múltiplas perspectivas na análise em contraste com as alternativas da literatura que tipicamente focam em uma análise sob uma única perspectiva. Como trabalho futuro, pretendemos realizar uma série de experimentos computacionais com base no modelo proposto, culminando em uma análise multi-perspectiva do desempenho de plataformas de processamento de grafo, delineando assim diretrizes para a seleção da plataforma mais adequada conforme a análise pretendida e o conjunto de redes complexas a ser analisado.

Agradecimentos: Este trabalho foi parcialmente financiado pelo CNPq.

¹9th DIMACS, Laboratory for Web Algorithms, Microsoft Academic Graph, Network Data Repository, Stanford Large Network Dataset Collection, The Koblenz Network Collection, The University of Florida Sparse Matrix Collection

REFERÊNCIAS

- Aldecoa, R., C. Orsini e D. Krioukov (2015). “Hyperbolic graph generator”. Em: *Computer Physics Communications* 196, pp. 492–496.
- Aliakbary, Sadegh, Jafar Habibi e Ali Movaghar (2015). “Feature extraction from degree distribution for comparison and analysis of complex networks”. Em: *The Computer Journal*, bxv007.
- Doekemeijer, N. e A.L. Varbanescu (2014). *A survey of parallel graph processing frameworks*. Rel. téc. Delft University of Technology.
- Elshawi, Radwa et al. (2015). “Big Graph Processing Systems: State-of-the-Art and Open Challenges”. Em: *Proc. of the IEEE Int. Conf. on Big Data Computing Service and Applications (BigDataService)*, pp. 24–33.
- Erdos, P e A Renyi (1959). “On random graphs I”. Em: *Publ. Math. Debrecen* 6, pp. 290–297.
- Guo, Y. et al. (2014). “How Well Do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis”. Em: *Proc. of the IEEE Int. Parallel and Distributed Processing Symp. (IPDS)*, pp. 395–404.
- Iosup, A. et al. (2016). “LDBC Graphalytics: A Benchmark for Large-scale Graph Analysis on Parallel and Distributed Platforms”. Em: *Proc. of the VLDB Endow.* 9.13, pp. 1317–1328.
- Kalavri, V., V. Vlassov e S. Haridi (2016). “High-Level Programming Abstractions for Distributed Graph Processing”. Em: *arXiv preprint arXiv:1607.02646*.
- Krioukov, D. et al. (2010). “Hyperbolic geometry of complex networks”. Em: *Phys. Rev. E* 82 (3), p. 036106.
- Lilja, D. J. (2005). *Measuring computer performance: a practitioner’s guide*. Cambridge University Press.
- Looz, M. von et al. (2016). “Generating massive complex networks with hyperbolic geometry faster in practice”. Em: *arXiv preprint arXiv:1606.09481*.
- Lovász, L. et al. (2009). “Very large graphs”. Em: *Current Developments in Mathematics* 2008, pp. 67–128.
- Newman, M. (2010). *Networks: An introduction*. Oxford University Press.
- Pokorny, J. (2015). “Graph Databases: Their Power and Limitations”. Em: *IFIP Int. Conf. on Computer Information Systems and Industrial Management*. Springer, pp. 58–69.
- Silva, D. N. R. da et al. (2016). “Análise de Desempenho de Plataformas de Processamento de Grafos”. Em: *Simpósio Brasileiro de Bancos de Dados*.
- Yan, D. et al. (2016). “Big Graph Analytics Systems”. Em: *Proc. of the Int. Conf. on Management of Data (ACM SIGMOD)*. New York, NY, USA: ACM, pp. 2241–2243.