Monografia: Graph 500
Computação Paralela e Distribuída

Renzo Gonzalo Gómez Diaz
NUSP: 7326191

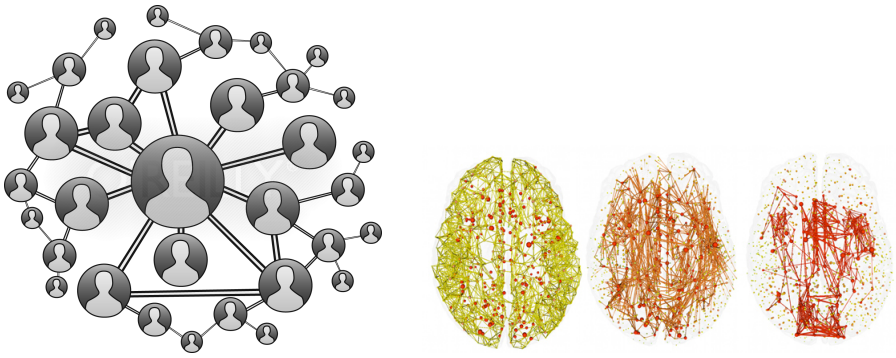
1 Introdução

Desde seus inícios, o desenvolvimento da computação de alto desempenho de larga escala (*"large-scale high performance computing"*) tem se focado no desenvolvimento e melhora da computação que envolve aritmética de ponto-flutuante. Isso se deve a que, maioritariamente, as aplicações para as quais são usadas estes supercomputadores consistem em fazer simulações de algum fenômeno físico. Neste tipo de simulações, a maioria dos cálculos envolve operações com variáveis de ponto-flutuante. Além disso, para obter uma melhor simulação é necessário fazer uma maior quantidade de cálculos. É esta necessidade a que há levado aos pesquisadores a investigar diferentes formas de aumentar este poder de computo.

Por outro lado, este modelo de computação tem motivado o surgimento de *benchmarks* criados especialmente para medir estas capacidades, como por exemplo SPEC, LINPACK, etc. Este último é quem define a lista (ranking) de supercomputadores chamado *Top 500*. Esta lista tenta medir as capacidades de computo de diferentes supercomputadores baseando-se nos resultados do *benchmark* LINPACK. Os supercomputadores são classificados dependendo da quantidade de operações de ponto-flutuante feitas em um segundo. O *Top 500* surgiu em 1993, e ao longo dos anos tem ganhado uma boa reputação entre a comunidade de pesquisadores de HPC.

Porém, nos últimos anos tem surgido problemas que exibem características diferentes às mostradas pela simulação de fenômenos físicos. Estes problemas tem natureza combinatória e, na maioria dos casos, são representados por meio de grafos. Por exemplo, se representarmos uma pessoa por um vértice e uma relação de amizade

por uma aresta, podemos representar uma rede social como um grafo (Figura 1(a)). Neste caso, podemos desejar, dado pessoa, encontrar possíveis amigos dela. O que se traduz a fazer uma busca neste grafo.



(a) *Redes sociais.* **(b)** *Redes neuronais.*

Figura 1: *Exemplos de problemas modelados por meio de grafos.*

De forma similar, podemos representar os neurônios do cérebro humano e suas conexões, e analisar hipóteses cientificas usando este tipo de abstração (Figura 1(b)). Além disso, é importante notar que estes grafos contém uma grande quantidade de dados. No caso do cérebro humano o número de vértices e arestas é da ordem de bilhões, e no caso das redes sociais, o tamanho é praticamente ilimitado. Logo, é necessário o uso de supercomputadores para resolver este tipo de problemas. Porém, este tipo de problemas são diferentes às simulações físicas em muitos aspectos. Por um lado, as simulações físicas precisam fazer grande quantidade de cálculos de ponto-flutuante e as estruturas de dados envolvidas são armazenadas em memória de forma estruturada (vetores, matrizes). Por outro lado, as aplicações com grafos usam operações com inteiros (cálculos de índices) e as estruturas de dados armazenadas em memória não seguem um padrão estruturado. Isso levou a pesquisadores do Sandia Laboratory nos E.E.U.U. a propor uma nova lista que tente classificar os supercomputadores segundo este tipo de aplicações. É desta forma que surgiu a lista *Graph 500*.

A estrutura desta monografia é a seguinte. Na Seção 2 tratamos sobre a motivação e objetivos que levaram à criação do *Graph 500*. Depois, na Seção 3, falamos sobre o *benchmark* usado para classificar os supercomputadores na lista. Na Seção 4,

mostramos alguns resultados da lista *Graph 500*. Além disso, mostramos uma comparação dos resultados ao longo do tempo e uma comparação com a lista *Top 500*. Finalmente, na Seção 5, damos as conclusões deste trabalho.

2 Motivação e objetivos

2.1 Motivação

A lista *Graph 500* tem seu origem nos resultados obtidos por investigações feitas por pesquisadores do laboratório Sandia dos E.E.U.U, em particular do Richard Murphy. Em primeiro lugar, Richard Murphy pesquisou a diferença no padrão de acesso a memória de *benchmarks* focados em operações de ponto-flutuante, em operações com inteiros e outras aplicações usadas no laboratório Sandia [4]. Nestes experimentos se analisam características como a localidade espacial, localidade temporal e a quantidade de dados diferentes acessados por cada aplicação. Na Figura 2, mostramos um gráfico que resume os resultados obtidos nesse artigo [4], notamos que a área da circunferência representa a quantidade de dados distintos acessados por cada aplicação.

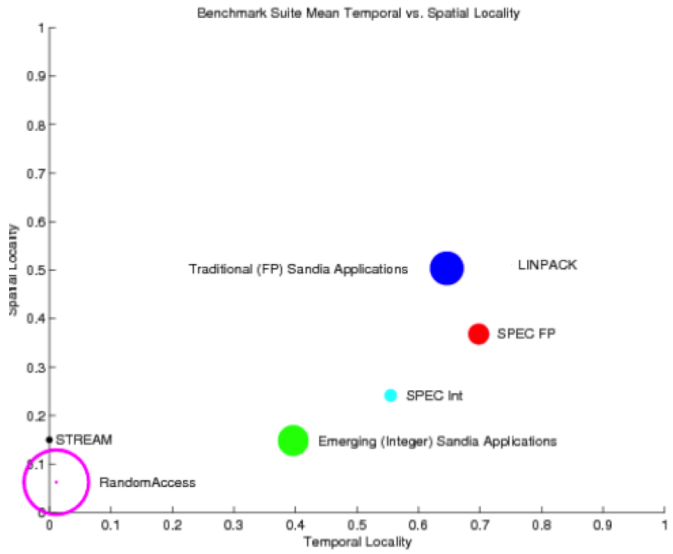


Figura 2: *Comparação entre localidade espacial, localidade temporal e uso de quantidade de dados.*

Dentre os resultados mais importantes, os autores obtiveram evidência de que aplicações em problemas de grafos e outros problemas que envolvem operações com inteiros exibem

- menor localidade espacial,
- menor localidade temporal e
- maior acesso a dados distintos em memória,

que as aplicações focadas em operações de ponto-flutuante.

Por outro lado, Richard Murphy também estudou o impacto que tem a banda passante (*"bandwidth*) e a latência do acesso a memória em aplicações de larga escala do laboratório Sandia [3]. Ele comparou aplicações baseadas em operações de ponto-flutuante com aplicações emergentes que usam, principalmente, operações com inteiros (buscas em grafos, caminhos mínimos, etc.) Os resultados mostram que as aplicações baseadas em operações com inteiros são mais sensíveis à banda passante e à latência do que as aplicações baseadas em operações de ponto-flutuante. Estes e outros resultados foram os que levaram a Richard Murphy e outros pesquisadores do laboratório Sandia a propor a lista do *Graph 500* [5].

2.2 Objetivos

Os objetivos principais da lista *Graph 500* são

- Ser um complemento para a lista *Top 500*.
- Incentivar pesquisa e desenvolvimento de outras arquiteturas de computadores para melhorar o desempenho de aplicações como as propostas pelo comitê do *Graph 500*.

Segundo a palavra dos autores [5], "as arquiteturas modernas exibem capacidades crescentes para o acesso a memória estruturado (como nas GPUs), memórias caché para explorar operações com alta localidade temporal ... Porém, não é seguro que estes avanços tenham grande impacto em problemas que trabalham com grande quantidade de dados, como os mostrados pelos *kernels* do Graph 500". Desta forma, podemos notar como a lista *Graph 500* quer chamar a atenção da comunidade de pesquisadores de HPC e das grandes empresas para que seja investido esforço e

dinheiro em resolver este novo tipo de problemas. Na seguinte seção falamos sobre o *benchmark* que é usado para fazer a classificação na lista *Graph 500*.

3 O Benchmark

Nesta seção descrevemos os objetivos do *benchmark* do *Graph 500* e as características dele. Segundo os autores [5], o *benchmark* usado na lista *Graph 500* deve exibir as seguintes propriedades:

1. **Ser um *kernel* com amplo alcance de aplicações:** Isto quer dizer que o *benchmark* deve refletir uma classe de algoritmos de grande impacto em diferentes aplicações.

2. **Ser mapeado a problemas reais:** Os resultados obtidos no *benchmark* devem ser mapeados a resultados em problemas reais, e não ser só um resultado puramente teórico.

3. **Exibir conjuntos de dados reais:** O conjunto de dados deve exibir padrões encontrados nos dados na vida real.

Agora, descrevemos o *benchmark* usado na lista do *Graph 500*. Atualmente, o único *benchmark* do Graph 500 é chamado *"Search"* (busca) e o propósito deste é testar a rapidez do supercomputador fazendo uma busca em largura (concorrente) em um grafo gerado aleatoriamente. Segundo o comitê de direção do *Graph 500* [1], a intenção deste *benchmark* é a de implementar "uma aplicação compacta que tenha múltiplas técnicas de análise (*kernels*) acessando uma mesma estrutura de dados que representa um grafo não-orientado com pesos nas arestas". Uma instância deste *benchmark* é definida por dois parâmetros. Eles são a escala e o fator de arestas. A **escala** é o logaritmo em base 2 do número de vértices, ou seja, se N é o número de vértices do grafo e e é a escala, então $e = \log_2(N)$. O **fator de arestas** é a proporção M/N, onde M e N são o número de arestas e vértices do grafo, respectivamente. Na Figura 3, mostramos como o comitê do *Graph 500* classifica uma instância do *benchmark* dependendo do valore da escala e do fator de arestas.

Classe	Escala	Factor de arestas	Tamanho (Aprox. TB)
Brinquedo	26	16	0.0172
Mini	29	16	0.1374
Pequeno	32	16	1.0995
Médio	36	16	17.5922
Grande	39	16	140.7375
Enorme	42	16	1125.8999

Figura 3: *Classificação das instâncias do Graph 500.*

Se seguir, listamos as etapas que compõem o *benchmark "Search"*.

1. Geração do conjunto de arestas do grafo.
2. Construção do grafo a partir do conjunto de arestas (*kernel 1*).
3. Gerar 64 pares de vértices distintos de forma aleatória.
4. Para cada par de vértices (u, v) fazemos o seguinte
   (a) Fazer uma busca em largura a partir de u. (*kernel 2*).
   (b) Validar os dados devolvidos pela BFS.
5. Calculamos as estatísticas do desempenho da aplicação.

Agora, descrevemos cada uma dessas etapas. Em primeiro lugar, para gerar o conjunto de arestas do grafo usamos um gerador de Kronecker (similar ao R-MAT [2]). Nesta etapa, não tomamos em consideração o tempo necessário para gerar o conjunto de arestas. Dada a lista de tuplas que representam as arestas do grafo, na seguinte etapa criamos a estrutura de dados que representa o grafo. É importante mencionar que é permitido usar qualquer representação do grafo (matriz, lista de adjacência, etc.). Porém, não é permitido mudar essa representação nas etapas posteriores. Além disso, o tempo que leva a geração do grafo é medido e considerado no resultado final. Por outro lado, para gerar os 64 pares de vértices, devemos considerar só vértices que tem grau pelo menos um, sem considerar laços. Depois, para cada par de vértices, digamos u e v, fazemos uma busca em largura (BFS) a partir de u, de forma concorrente. Notamos que, nesta etapa não é permitido fazer em paralelo buscas de pares distintos de vértices. Como saída do algoritmo, devolvemos um vetor que representa a árvore de caminhos mínimos a partir de u. O

tempo de cada BFS é medido e considerado no resultado final. Finalmente, para cada BFS validamos se o vetor devolvido representa uma árvore de caminhos mínimos do grafo criado no Passo 2. O tempo gasto para fazer essa validação não é considerado no resultado final.

Agora, falamos sobre como os resultados deste *benchmark* servem para classificar os supercomputadores. A medida de desempenho usada para comparar supercomputadores diferentes é chamada de TEPS ("Traversed Edges Per Second"). Esta medida é uma proporção entre o número de arestas do grafo e o tempo de execução de uma busca em largura. Em outras palavras, se M é o número de arestas do grafo e T é o tempo de execução do *kernel 2*, então

$$\text{TEPS} = \frac{M}{T}.$$

Agora, notamos o seguinte sobre o *benchmark*. Se bem a documentação dada na página é ampla e dá exemplos de implementação, desde que foi lançada esta lista (2010) até hoje o único teste presente no *benchmark* é o "Search". Isto é algo que, de certa forma, contradiz os objetivos mencionados ao início desta seção, já que existem muitas outras classes de problemas (caminhos mínimos, clustering, etc.) de grande importância prática que deveriam ser considerados para serem testados pelo *benchmark*. Tal vez esta seja uma das razões pelas quais a lista do *Graph 500* não tenha ganhado uma aceitação similar à obtida pela lista do *Top 500*, como falamos na próxima seção.

4 A lista Graph 500

Nesta seção falamos sobre a lista do *Graph 500*, e sua evolução no tempo. A primeira lista foi publicada em novembro de 2010, na conferência de supercomputadores da ACM/IEEE. Desde então, uma lista nova é publicada duas vezes por ano, nos meses de junho e novembro. O comitê de direção encarregado de produzir esta lista está composto por pesquisadores de diversas instituições publicas (Sandia Lab., U. de Stanford, U. de Georgia Tech, etc.) e privadas (Google, NVIDIA, AMD, etc.). Na Figura 4, mostramos os dez primeiros lugares da última lista publicada (novembro 2014).

Pos.	Supercomp.	Instituição	Num. nós	Num. cores	Escala	GTEPS
1	Sequoia	LLNL	98304	1572864	41	23751
2	K computer	RIKEN AICS	82944	663552	40	19585.2
3	Mira	ANL	49152	786432	40	14982
4	JUQUEEN	FZJ	16384	262144	38	5848
5	Fermi	CINECA	8192	131072	37	2567
6	Tianhe-2	Changsha, China	8192	196608	36	2061.48
7	Turing	CNRS	4096	65536	36	1427
7	Blue Joule	STFC	4096	65536	36	1427
7	DIRAC	U. of Edinburgh	4096	65536	36	1427
7	Zumbrota	EDF R&D	4096	65536	36	1427

Figura 4: *Lista Graph 500 (novembro 2014).*

Um aspecto que consideramos relevante é comparar a evolução desta lista ao longo do tempo. Para isso, tomamos como base esses dez primeiros lugares e comparamos suas posições ao longo do tempo. Na Figura 5, mostramos um gráfico com esta comparação. Nesse gráfico, por cada supercomputador mostramos uma curva que mostra sua posição na lista desde o ano 2012.

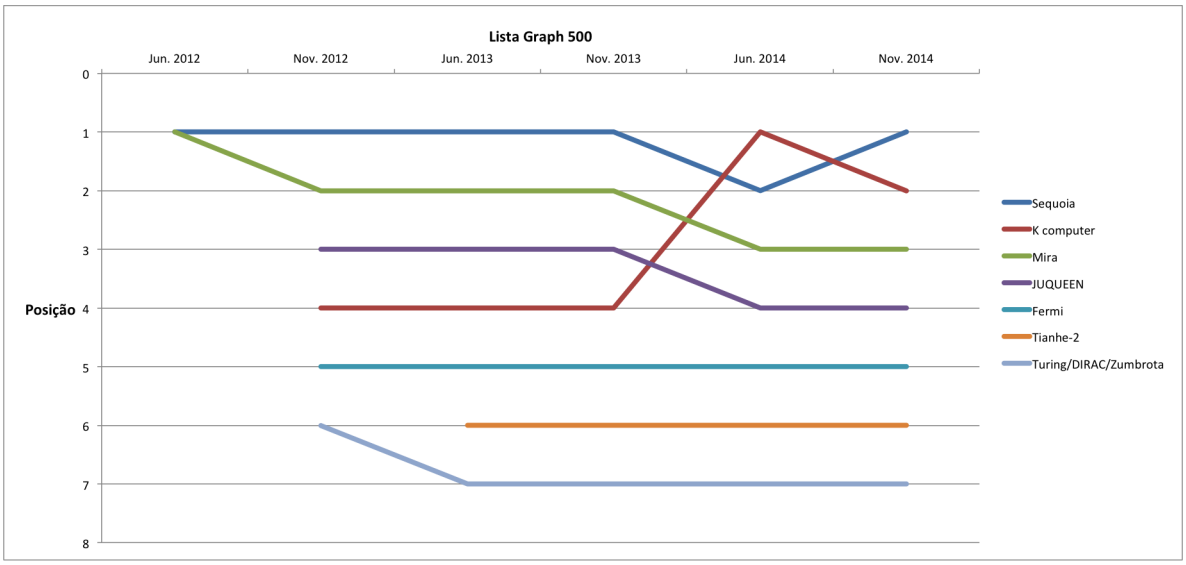


Figura 5: *Evolução dos 10 primeiros lugares nos últimos 3 anos.*

Como podemos notar, o supercomputador Sequoia do laboratório nacional de Lawrence Livermore é quem tem dominado a lista quase desde o início. Além disso, em geral, notamos que cada supercomputador considerado tem mantido sua posição

ao longo do tempo. A única mudança significativa foi obtida pelo supercomputador *K-computer* do instituto RIKEN (Japão), que passou de quarto para o os primeiros lugares nos últimos anos.

Por outro lado, consideramos relevante comparar os resultados da lista *Graph 500* com os resultados da lista *Top 500* no mesmo período. Considerando as listas de novembro 2014, na Figura 6, mostramos as posições obtidas por estes supercomputadores em ambas listas.

Top 500	Graph 500	Supercomputador
3	1	Sequoia (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)
4	2	K computer (Fujitsu - Custom supercomputer)
5	3	Mira (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)
8	4	JUQUEEN (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)
23	5	Fermi (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)
1	6	Tianhe-2 (MilkyWay-2) (National University of Defense Technology - MPP)
42	7	Turing (IBM - BlueGene/Q, Power BQC 16C 1.60GHz)
30	7	Blue Joule (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)
43	7	DIRAC (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)
73	7	Zumbrota (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)

Figura 6: *Comparação com a lista Top 500.*

Podemos ver que o supercomputador Tianhe-2 (China) que ocupa a posição 1 no *Top 500*, ocupa a posição 6 na lista *Graph 500*. Mais notório ainda são as posições obtidas pelos supercomputadores Turing, Blue Joule, DIRAC e Zumbrota. Por um lado, eles estão na posição 7 na lista *Graph 500* e por outro, eles estão muito afastados dos primeiros lugares na lista *Top 500*. Isto pode ser uma evidência que a lista *Top 500* está focada em certo tipo de problemas computacionais e reforça os resultados obtidos por Richard Murphy e outros pesquisadores, que mostram a diferença entre problemas derivados de simulações físicas, e problemas emergentes

que estão focados em grafos e no processamento de grande quantidade de dados.

Por outro lado, é importante mencionar que supercomputadores como o Titan (China) ou Piz Daint (CSCS - Suiza), que ocupam as posições 1 e 6, respectivamente, na lista *Top 500*, não foram considerados na lista *Graph 500*. E mais ainda, embora a lista seja chamada *Graph 500*, na última lista (novembro 2014) só foram considerados 183 supercomputadores. Isto pode dever-se a que, os representantes de um supercomputador que deseja ser considerado na lista, devem enviar os resultados do *benchmark* "Search" ao comitê de direção do *Graph 500*. Logo, isso pode evidenciar que a lista *Graph 500* ainda não tem o prestigio atingido pela lista *Top 500*.

5 Conclusões

Finalmente, após o estudo da informação apresentada nesta monografia, concluímos o seguinte,

- A lista *Graph 500* mostra que existem problemas emergentes, diferentes de simulações físicas e meteorológicas, que precisam de grande poder de computo. Neste sentido, é importante incentivar pesquisa e desenvolvimento de outras arquiteturas de computadores para tentar resolver este tipo de problemas.

- A lista *Graph 500* ainda é nova (comparada com o *Top 500*), e precisa ganhar maior prestigio dentro da comunidade de HPC. Para isso, precisa aumentar o tipo de aplicações que são testadas pelo seu *benchmark*. Como foi visto neste trabalho, existem outros problemas de interesse prático que deveriam ser considerados. Desta forma, poderia obter maior credibilidade dos resultados obtidos.

Referências

[1] *Graph 500 site*, `http://www.graph500.org/`, Accessed: 16-06-2015.

[2] D. Chakrabarti, Y. Zhan, and C. Faloutsos, *R-mat: A recursive model for graph mining*, SIAM International Conference on Data Mining, 2004.

[3] R.C. Murphy, *On the effects of memory latency and bandwidth on supercomputer application performance*, 2013 IEEE International Symposium on Workload Characterization (IISWC) **0** (2007), 35–43.

[4] R.C. Murphy and P.M. Kogge, *On the memory access patterns of supercomputer applications: Benchmark selection and its implications*, Computers, IEEE Transactions on **56** (2007), no. 7, 937–945.

[5] R.C. Murphy, K.B. Wheeler, W. Barret, B, and J.A. Ang, *Introducing the Graph 500*, Cray User's Group (CUG), May 2010.