

# Análise de Custo da Nuvem Computacional para a Execução de Algoritmos no Processamento Sísmico

Nicholas T. Okita<sup>1</sup>, Tiago A. Coimbra<sup>1</sup>, Edson Borin<sup>1</sup>

<sup>1</sup>Centro de Estudos de Petróleo (CEPETRO)  
Universidade Estadual de Campinas (UNICAMP)  
Caixa Postal 6052 – 13.083-970 – Campinas – SP – Brasil

nicholas.okita@ggaunicamp.com, tiago@ggaunicamp.com, edson@ic.unicamp.br

**Resumo.** *O objetivo deste trabalho é a investigação do uso da nuvem computacional para a execução de algoritmos no processamento sísmico, visando minimizar o custo financeiro com o uso do serviço em nuvem contra a compra de máquinas físicas. Os resultados obtidos indicam que (i) as máquinas virtuais equipadas com GPU apresentam o menor custo/benefício para a aplicação executada, (ii) existe escalabilidade no desempenho e esta pode ser explorada para obter tempos de execuções menores e, por fim, (iii) o uso da nuvem permite que não seja necessária a compra de novas máquinas físicas.*

## 1. Introdução

Nos dias atuais, tarefas que necessitam de alto poder computacional ganharam destaque devido ao processamento de dados cada vez maiores. Assim, o processamento de grandes volumes de dados em um tempo viável é um dos principais desafios na sísmica de exploração. Para tratar de tal desafio, a nível de *software*, algoritmos e métodos heurísticos mais eficientes foram criados para reduzir o tempo de execução. Entretanto, em nível de *hardware*, o processamento paralelo e em nuvem foram aprimorados a fim de reduzir o tempo de processamento.

Este trabalho visa o processamento em nuvem como uma alternativa para a compra de um novo *hardware*, o qual apresenta problemas como a incerteza de qual será o investimento inicial para que o mesmo não seja obsoleto até o fim do projeto, além da manutenção, instalação e uso dependendo do tamanho do desse *hardware*, por exemplo um *cluster*. Enquanto em nuvem, usando-se do modelo de negócio “*Pay as you go*”, não existe essa preocupação. Pois, paga-se apenas a quantidade de tempo utilizado nas especificações de *hardware* desejadas.

Neste trabalho dois tópicos foram investigados, os quais são: (i) a busca do menor custo/benefício na escolha de máquinas em nuvem para o processamento sísmico; e (ii) a análise do investimento em *hardware* contra o uso do mesmo investimento em máquinas em nuvem.

O texto está organizado da seguinte forma: a seção 2 apresenta os recursos computacionais da nuvem utilizados, a descrição dos testes executados e os métodos de comparação. A seção 3 apresenta e discute os resultados dos experimentos realizados, obtendo informações sobre a otimização de custo de uso da nuvem computacional. Por fim, a seção conclusão apresenta as conclusões derivadas dos experimentos.

## 2. Materiais e métodos

A nuvem computacional Azure, da Microsoft, foi utilizada para realização dos experimentos onde vários conjuntos de máquinas foram criados para comparar o desempenho e o custo do algoritmo do método *common-midpoint* (CMP) [Yilmaz 2001]. O método CMP foi implementado utilizando o modelo *Spits* da plataforma *PY-PITS* [Borin et al. 2016]. O *Spits* é um modelo de programação que facilita a implementação e execução escalável de tarefas paralelas em sistemas computacionais distribuídos.

As infraestruturas utilizadas em nuvem foram de *mini-clusters* em CPUs e GPUs. A aplicação foi executada três vezes em cada *mini-cluster* e a mediana foi utilizada para resumir o resultado. A ferramenta de compilação do código utilizada foi o *gpp* versão 5.4.0 com a *flag* de otimização *O3*. Para o caso específico de GPUs foi utilizado o *OpenCL* versão 1.2. Por fim, para execução do *PY-PITS* foi utilizado *Python* versão 3.

**Tabela 1. Grupos de máquinas**

Grupo	Máquinas	Núcleo de processamento	Memória RAM	Custo por hora
CPU0	4 x A9	64 vCPUs (Xeon E5-2670)	448GB	R\$ 25,8960
CPU1	1 x D64	64 vCPUs (Xeon E5-2673 v3)	256GB	R\$ 10,1990
CPU2	2 x D64	128 vCPUs (Xeon E5-2673 v3)	512GB	R\$ 20,3980
CPU3	4 x D64	256 vCPUs (Xeon E5-2673 v3)	1024GB	R\$ 40,7960
GPU0	2 x NC6	2 Nvidia K80	112GB	R\$ 05,0996
GPU1	1 x NC6S	1 Nvidia P100	112GB	R\$ 06,8724
GPU2	2 x NC6S	2 Nvidia P100	224GB	R\$ 13,7448
GPU3	1 x NC24S	4 Nvidia P100	448GB	R\$ 27,4896
HET0	4 x D16 + 2 x NC6	64vCPUs (Xeon E5-2673 v3) 2 Nvidia K80	368GB	R\$ 15,2988
HET1	1 x NC6S	6 vCPUs (Xeon E5-2690) 1 Nvidia P100	112GB	R\$ 06,8724
HET2	1 x NC6S	4 vCPUs (Xeon E5-2690) 1 Nvidia P100	112GB	R\$ 06,8724

A Tabela 1 descreve os sistemas utilizados em nossos experimentos. O campo grupo descreve o dispositivo computacional utilizado para a execução do método. Aqui, CPU# representa somente CPUs, GPU# somente GPUs e HET# representa ambos CPUs e GPUs. O campo máquinas descreve as máquinas virtuais utilizadas. O campo núcleo de processamento indica quantos núcleos virtuais (vCPUs) foram usados identificando o processador de cada vCPU e os aceleradores. Por fim, o campo custo por hora mostra qual o custo das máquinas. Este valor é informado pelo provedor de serviço da nuvem, e.g., no grupo HET2 foi usada uma máquina NC6S com 4 vCPUs e uma Nvidia P100 a um custo de R\$ 6,8724 por hora de uso.

Os testes executados envolviam computar os parâmetros do método CMP para um dado de cerca de 950 MB, através do algoritmo heurístico evolutivo *differential evolution* (DE) [Storn and Price 1995] com cento e uma sementes e cento e uma gerações de evolução. Esses parâmetros combinados com outros específicos do processamento do

dado, os quais a discussão está fora do escopo deste trabalho, garantiram uso de 100% das CPUs e das GPUs quase constante no decorrer da execução.

Por fim, duas métricas foram desenvolvidas para a avaliação dos resultados, sabendo que o maior tempo do programa é gasto computando interpolações. Essas métricas são: (i) interpolação por unidade monetária, que mede o potencial do algoritmo na máquina em relação ao custo monetário; e (ii) interpolações por unidade de tempo, que mede a velocidade que o programa computa interpolações.

### 3. Resultados e discussões

Na Tabela 2, pode-se comparar os resultados dos testes para cada grupo, contendo o tempo de execução em segundos, o custo total em Reais para a execução, a métrica de desempenho interpolações por segundo e a métrica financeira interpolações por Real. Por exemplo, o grupo GPU3 levou 158,24 segundos a um custo total de R\$ 1,21. Neste contexto, foram realizadas  $1,34E+11$  interpolações por segundo e  $1,75E+13$  interpolações por Real.

**Tabela 2. Resultados**

Grupo	Tempo	Custo	Interpolações/segundo	Interpolações/R\$
CPU0	3194,96 s	R\$ 22,98	6,55E+09	9,11E+11
CPU1	4750,91 s	R\$ 13,46	4,41E+09	1,56E+12
CPU2	2365,08 s	R\$ 13,40	8,85E+09	1,56E+12
CPU3	1160,26 s	R\$ 13,15	1,80E+10	1,59E+12
GPU0	0748,08 s	R\$ 01,06	2,83E+10	2,00E+13
GPU1	0483,99 s	R\$ 00,92	4,38E+10	2,29E+13
GPU2	0247,41 s	R\$ 00,94	8,56E+10	2,24E+13
GPU3	0158,24 s	R\$ 01,21	1,34E+11	1,75E+13
HET0	0737,71 s	R\$ 03,14	2,87E+10	6,57E+12
HET1	0753,07 s	R\$ 01,44	2,81E+10	1,47E+13
HET2	0490,62 s	R\$ 00,94	4,32E+10	2,26E+13

Os resultados mostram que apesar das máquinas A9 utilizadas no grupo CPU0 serem indicadas pelo provedor para a execução de código de alto desempenho, para a aplicação executada, o custo benefício das máquinas D64 - dos grupos CPU1, CPU2 e CPU3 - mostrou-se superior. Ao explorar a escalabilidade do código, permitiu-se obter resultados mais rápidos e mais baratos aumentando o número de instâncias D64.

Ao usar máquinas CPU com máquinas GPU - grupo HET0 - existe uma perda considerável na métrica interpolações por real, pois as CPUs aumentaram o preço em uma escala maior que aumentam o desempenho. No grupo HET1, não vale a pena o uso de todos os núcleos da CPU para tentar ajudar a GPU, pois a GPU fica subutilizada pela demora maior ao receber dados, enquanto ao usar apenas 4 núcleos - no grupo HET2 -, não se percebe diferença para o uso somente da GPU.

As GPUs apresentam os melhores resultados tanto em tempo quanto em custo benefício, vide que o grupo GPU1 mostra o melhor resultado em Interpolações/R\$. Comparando os grupos GPU1, GPU2 e GPU3, o aumento no número de GPUs mostra uma melhora no tempo, porém existe também uma perda de custo benefício. Ape-

sar desta aplicação não escalar com o preço, o aumento de desempenho ainda é considerável. Esta escalabilidade de CPUs e GPUs se dá devido à característica de algoritmo embarçosamente paralelo.

Para adquirirmos a máquina NC24S, seria necessário o investimento de cerca de 315 mil Reais, baseando-se nos preços da *Dell*. Supondo um projeto de 48 meses, vamos agora verificar quanto teria que ser o uso da máquina para compensar o investimento em *hardware*, desconsiderando o investimento em TI para a manutenção e também o custo de Internet, além disso supõe-se que, usando a nuvem, o valor inicial da máquina está em um investimento fixo de 0,5% ao mês, enquanto ao adquirir o *hardware*, ele será vendido no final do projeto com uma depreciação entre 20% a 36% ao ano.

Dado um consumo elétrico de 3500 W para a máquina e o sistema de refrigeração com um preço de energia elétrica de R\$0,6/kWh contra o gasto de R\$ 27,4896 por hora da máquina NC24S em nuvem, conclui-se que seria necessário usar a máquina 70% do tempo para compensar o investimento em *hardware*, caso contrário, a nuvem é mais economicamente viável.

#### 4. Conclusão

Neste trabalho investiga-se o uso da nuvem computacional Azure, da Microsoft, para a execução de algoritmos no processamento sísmico. Sendo que nessa nuvem o desempenho do algoritmo do método CMP é analisado para cada tipo de *hardware* utilizado. Os resultados mostram uma redução do custo/benefício ao utilizar máquinas do grupo CPU ou do grupo HET contra o uso de máquinas GPU, concluindo assim que máquinas do grupo GPU são de fato as melhores escolhas para o tipo de aplicação. Além disso, observa-se que em aumentar o número de núcleos de CPU ou GPUs o desempenho melhora proporcionalmente, ou seja, o tempo de execução diminui ao passo que o aumenta o número de núcleos de processamento, dessa forma, mostra-se que existe escalabilidade no programa. Por fim, caso a nuvem seja usada menos que 70% no período o seu uso é mais vantajoso que a compra de novas máquinas físicas.

#### 5. Agradecimentos

Os autores agradecem à FAPESP/CEPID #2013/08293-7 e à Petrobras pela ajuda financeira, ao laboratório *High Performance Geophysics* (HPG) pelo suporte computacional e à Microsoft pelo acesso à Azure.

#### Referências

- Borin, E., Benedicto, C., Rodrigues, I. L., Pisani, F., Tygel, M., and Breternitz, M. (2016). PY-PITS: A scalable python runtime system for the computation of partially idempotent tasks. In *2016 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW)*. IEEE.
- Storn, R. and Price, K. (1995). Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. international computer science institute, berkeley. Technical report, CA, 1995, Tech. Rep. TR-95-012.
- Yilmaz, O. (2001). *Seismic Data Analysis*, volume 10 of *Investigations in Geophysics*. Society Of Exploration Geophysicists, Tulsa, USA, 2 ed edition.