

Análise de Desempenho de um Simulador de Reservatórios de Petróleo em um Ambiente de Computação em Nuvem

Maicon Melo Alves¹, Lúcia Maria de Assumpção Drummond¹

¹Instituto de Computação - Universidade Federal Fluminense (UFF)
Niterói - RJ - Brasil

{mmelo, lucia}@ic.uff.br

Abstract. *Cloud Computing is a distributed computing paradigm which customers access computing resources from the Internet. Recent researches investigate the use of clouds to perform scientific computing that requires a lot of computing power. In this paper we present a performance analysis of a real petroleum reservoir simulator in a cloud environment provided by both Amazon EC2 and Microsoft Azure platforms. This performance analysis takes into account metrics related to operational system and the results of some specific benchmarks. The results show that virtualization overhead and resource sharing can drastically decrease performance of such applications.*

Resumo. *A computação em nuvem (cloud computing) é considerada como um novo paradigma de computação distribuída em que os clientes podem acessar recursos diretamente da Internet. Estudos recentes avaliam o uso dos recursos providos pela nuvem para executar aplicações científicas que demandam alto poder computacional. Neste sentido, este trabalho apresenta uma análise de desempenho de um simulador de reservatórios de petróleo a fim de avaliar o comportamento de tal aplicação científica em um ambiente de computação em nuvem provido pelas plataformas Amazon EC2 e Microsoft Azure. Essa análise leva em conta a observação de métricas do sistema operacional e a execução de benchmarks específicos. Os resultados mostram que o overhead de virtualização e o compartilhamento de recursos causam um decréscimo significativo de desempenho em tais aplicações.*

1. Introdução

Os recentes avanços em virtualização permitiram o surgimento de um novo paradigma de computação distribuída em larga escala conhecido como computação em nuvem (*cloud computing*). Neste novo paradigma, os clientes podem acessar recursos computacionais como máquinas virtuais, armazenamento e plataformas de desenvolvimento a partir da Internet mediante um acordo de nível de serviço. Um ambiente de computação em nuvem apresenta algumas vantagens se comparado com as formas tradicionais de prover infraestrutura computacional como, por exemplo, elasticidade e redução de custos. Ao utilizar um ambiente de cloud é possível obter uma redução de custos operacionais e de investimento, já que não é preciso levar em conta os gastos relacionados a licenças de software e obsolescência de hardware, além de não ser preciso contratar e manter pessoal especializado [Hoffa et al. 2008]. Uma outra vantagem é a elasticidade de recursos provida pela nuvem em que é possível alocar e liberar rapidamente uma grande quantidade

de recursos, mas sem levar em conta questões limitantes como espaço físico e capacidade energética [Subramanian et al. 2011]. Portanto, todas essas vantagens têm aumentado o interesse em realizar pesquisas a respeito de computação em nuvem como uma opção as tradicionais formas de prover infraestrutura computacional.

Nos últimos anos, a infraestrutura computacional provida pela nuvem tem sido frequentemente usada para abrigar aplicações web [Ramakrishnan et al. 2011]. No entanto, estudos recentes têm investigado o uso dos recursos providos pela nuvem para executar computação científica que requer alto poder computacional [Subramanian et al. 2011] [Vecchiola et al. 2009] [Ramakrishnan et al. 2011] [Ostermann et al. 2010] [Jackson et al. 2010] [Heilmann et al. 2013] [He et al. 2010] [Knight et al. 2012]. De fato, a computação em nuvem tende a ser uma interessante alternativa aos tradicionais supercomputadores, *clusters* de servidores e grades computacionais [Vecchiola et al. 2009] [Ostermann et al. 2010]. A computação em nuvem é considerada mais barata e muito mais escalável do que os supercomputadores e clusters, visto que os recursos computacionais na nuvem podem ser facilmente redimensionados de acordo com os requisitos das aplicações e dos recursos financeiros disponíveis [Ostermann et al. 2010]. Já as grades científicas apresentam alguns entraves técnicos que podem sensivelmente dificultar a implantação de aplicações, pois tais grades são compartilhadas entre grupos de usuários que trabalham em diferentes projetos científicos. Portanto, os softwares presentes no ambiente de uma grade como compiladores, depuradores e bibliotecas, podem não ser compatíveis com o ambiente necessário para executar alguma outra aplicação ou experimento. Ao contrário das grades, a infraestrutura presente na nuvem pode ser perfeitamente configurada e ajustada de acordo com os requisitos necessários para a execução de uma determinada aplicação científica [Vecchiola et al. 2009].

Apesar de todas essas vantagens, há uma série de desafios que precisam ser superados para que a nuvem possa executar aplicações científicas de forma satisfatória. Um dos principais problemas está relacionado a variabilidade de desempenho causada pelo compartilhamento de recursos físicos entre diversas máquinas virtuais [Jackson et al. 2010]. Outra questão é o próprio *overhead* introduzido pelo software da virtualização (*hypervisor*) que pode certamente afetar o desempenho das aplicações [Subramanian et al. 2011]. Além disso, existem recursos requeridos por algumas aplicações que podem não estar presentes em um ambiente de nuvem como, por exemplo, uma rede de comunicação de baixa latência e alta largura de banda, e o acesso a recursos de I/O de alto desempenho [Jackson et al. 2010]. Consequentemente, a ausência destes quesitos específicos pode diminuir drasticamente o desempenho das aplicações científicas ao serem executadas na nuvem. Por esta razão, o uso de computação em nuvem para executar aplicações científicas tem se mostrado como uma promissora área de pesquisa.

Uma das áreas que possuem aplicações científicas que demandam grande poder computacional é a indústria de petróleo. Dentre os vários processos dessa indústria, a simulação de reservatórios é um dos que mais necessitam de poder computacional para ser executado. Basicamente, uma simulação de reservatório de petróleo realiza inferências a respeito do comportamento de um reservatório real a partir de um modelo matemático que representa aquele reservatório [Peaceman 2000]. Em outras palavras, os cientistas executam simulações acerca das condições do reservatório (fluidez, pressão, saturação) a

fim de conhecer seu estado atual e, posteriormente, tomar decisões que permitam obter uma máxima recuperação de hidrocarbonetos (óleo e gás). Para isso, a simulação avalia as condições do reservatório frente a diversas variáveis e por um longo período de tempo simulado (em torno de dois mil anos ou mais). Consequentemente, a execução de uma simulação pode levar muitas horas ou até dias para ser concluída. Diante da importância desta aplicação e dos requisitos por grandes recursos computacionais, as companhias de petróleo têm investido muitos recursos financeiros em supercomputadores ou clusters de servidores para executar este tipo de aplicação. Portanto, essas aplicações se caracterizam como candidatas promissoras à execução em um ambiente provido pela nuvem.

Diante do exposto, este trabalho apresenta uma análise de desempenho de um simulador de reservatórios de petróleo denominado MUFITS (Multiphase Filtration Transport Simulator) [Afanasyev 2013] a fim de estudar o comportamento deste tipo de aplicação científica real da área de petróleo na nuvem. Para entender este comportamento, o MUFITS foi executado tanto em uma máquina física quanto em máquinas virtuais providas pela Amazon EC2 e pela Microsoft Azure que são consideradas duas das mais representativas provedoras de serviços de computação em nuvem atualmente. O MUFITS foi avaliado de acordo com o tempo de execução e também quanto aos parâmetros do sistema operacional como uso de CPU, e vazão de memória e disco. Também foram executados alguns *benchmarks* de CPU e memória a fim de permitir uma análise mais concisa a respeito do desempenho do MUFITS na nuvem. Além dessa análise de desempenho, é realizada uma breve avaliação a respeito dos custos envolvidos para executar este tipo de aplicação na nuvem. Vale ressaltar que o escopo deste trabalho compreende uma execução intra-nó para avaliar mais apropriadamente a influência de parâmetros como CPU, memória e disco no desempenho dessas aplicações na nuvem.

Especificamente, este trabalho apresenta as seguintes contribuições: (i) é utilizada uma aplicação científica real e específica da indústria de petróleo; (ii) a análise de desempenho desta aplicação é executada em duas diferentes plataformas de *cloud* (Azure e EC2); (iii) os resultados obtidos na nuvem são comparados com um ambiente físico tradicional; (iv) além do tempo de execução da aplicação, são analisadas outras métricas como uso de CPU, e vazão de memória e disco. Quanto à organização, o trabalho está subdividido da seguinte forma. Na seção 2 são destacados os trabalhos relacionados. Na seção 3 é apresentado o cenário, os métodos adotados nos experimentos e também os resultados obtidos. Por fim, a seção 4 apresenta as conclusões finais e os trabalhos futuros.

2. Trabalhos Relacionados

Muito esforço tem sido empregado para descrever e caracterizar o desempenho de aplicações científicas ao serem executadas na nuvem. Com o propósito de entender o desempenho dos recursos providos pela nuvem frente a execução deste tipo de aplicação, alguns trabalhos como [Vöckler et al. 2011], [Ostermann et al. 2010] e [He et al. 2010] executaram testes de *benchmark* a fim de avaliar o desempenho individual dos subsistemas de memória, processador, disco e rede. Embora o uso de testes de *benchmark* seja uma forma adequada de avaliar o desempenho de um sistema, a execução de uma aplicação real permite avaliar mais apropriadamente o comportamento de um sistema computacional como um todo. Por essa razão, este trabalho avalia o desempenho da nuvem sob a perspectiva de uma aplicação real da área de petróleo. Ademais, esses trabalhos também não realizaram uma comparação do desempenho obtido na nuvem com o desempenho

que seria alcançado em um ambiente físico tradicional. Diferentemente desses estudos, este trabalho realiza uma comparação dos resultados obtidos na nuvem com os resultados atingidos em ambiente físico tradicional a fim de caracterizar a influência de questões inerentes à computação em nuvem como o compartilhamento de recursos físicos entre máquinas virtuais e o *overhead* introduzido pela virtualização [Jackson et al. 2010].

Outros trabalhos como [Hoffa et al. 2008] e [Subramanian et al. 2011] avaliam o desempenho da nuvem sob a perspectiva de uma aplicação real, além de comparar os resultados obtidos na nuvem com o desempenho alcançado em um ambiente tradicional. No entanto, estes trabalhos consideram apenas uma métrica para avaliar o desempenho: o tempo de execução da aplicação. Embora o tempo de execução seja a métrica primária para aferir o desempenho de uma aplicação em diferentes sistemas, esta medida não é suficiente o bastante para entender o porquê de um sistema ter alcançado um melhor resultado em detrimento dos demais. Portanto, este trabalho emprega métricas adicionais que estejam relacionadas a parâmetros de sistema como uso de CPU, e vazão de memória e disco para avaliar os resultados obtidos de uma maneira mais abrangente.

Mais especificamente, alguns trabalhos como [Ramakrishnan et al. 2011], [Vecchiola et al. 2009] e [Jackson et al. 2010] realizaram uma análise de desempenho baseada em um conjunto real de aplicações científicas, compararam os resultados obtidos na nuvem com a execução em um ambiente tradicional e levaram em conta outras métricas além do tempo de execução das aplicações. No entanto, nenhum dos trabalhos citados avaliou uma aplicação específica da indústria petrolífera. Como mencionado anteriormente, a indústria de petróleo é uma das áreas que mais demanda alto poder computacional para executar suas aplicações. Portanto, este trabalho avalia uma aplicação específica desta área por acreditar que podem existir características peculiares deste tipo de aplicação que diferem dos outros conjuntos de aplicações científicas reais.

Poucos trabalhos avaliam o uso da nuvem para executar aplicações específicas da área de petróleo. Em [Heilmann et al. 2013] é proposta uma solução para realizar uma análise de dados e um imageamento sísmico em tempo real usando uma infraestrutura provida pela nuvem. O trabalho citado apenas apresenta a solução criada, mas não realiza qualquer tipo de análise de desempenho a respeito da execução deste processo na nuvem. Já em [Subramanian et al. 2011], é efetuada uma análise da execução de uma inversão sísmica 3D na nuvem. No entanto, apesar de avaliar uma aplicação específica da indústria petrolífera e comparar os resultados da nuvem com a execução em um ambiente tradicional, o trabalho em questão não avalia outras métricas além do simples tempo de execução da aplicação. Ao contrário destes trabalhos citados, este estudo conduz uma análise de desempenho de um simulador de reservatórios real, levando em conta não somente o tempo de execução, mas também as demais métricas de sistema.

3. Experimentos Computacionais

Os experimentos foram conduzidos em duas etapas distintas. Na primeira etapa foram realizados ensaios de acordo com a execução do simulador de reservatórios MUFITS. Já na segunda etapa foram executados alguns testes adicionais de *benchmark* a fim de complementar o entendimento a respeito do desempenho do MUFITS obtido na nuvem. Em ambas as etapas, o processo de coleta de amostras foi repetido cinco vezes a fim de garantir um intervalo de confiança de 99% que fosse suficientemente pequeno. As próximas

subseções descrevem os ambientes computacionais usados em todos os experimentos, além das metodologias e dos resultados dos experimentos com o simulador MUFITS e com os testes de *benchmark*.

3.1. Ambientes Computacionais Usados nos Experimentos

Os experimentos foram executados em quatro ambientes computacionais distintos que consistiram de uma máquina física e de três máquinas virtuais alocadas na nuvem. Essas máquinas virtuais foram providas pelas plataformas de *cloud* da Amazon (denominada EC2) e da Microsoft (denominada Azure). A Tabela 1 apresenta a configuração de hardware desse conjunto de máquinas utilizado nos experimentos.

Tabela 1. Configuração de hardware das máquinas usadas nos experimentos

| Máquina | Tipo | Processador | Memória | Disco |
|------------|--------|----------------------------|---------|--------------|
| HP DL360G7 | Física | Intel Xeon X5650 2.66GHz | 32GB | 300GB SATA |
| C3.2XLARGE | EC2 | Intel Xeon E5-2680 2.80GHz | 15GB | 2 x 80GB SSD |
| M3.2XLARGE | EC2 | Intel Xeon E5-2670 2.50GHz | 30GB | 2 x 80GB SSD |
| A4 | Azure | AMD Opteron 4171HE 2.1GHz | 14GB | 127GB |

Conforme relacionado na Tabela 1, o conjunto de máquinas usado nos experimentos foi composto por um servidor físico HP de médio porte, por duas máquinas virtuais providas pela Amazon EC2 e por uma máquina virtual provida pela Microsoft Azure. Quanto às máquinas virtuais do EC2, a instância M3.2XLARGE é de propósito geral, enquanto que a instância C3.2XLARGE é específica para tarefas de alta intensidade computacional. Já a instância A4 da Azure é de propósito geral e pertence a classe de instâncias *default* que oferece melhor desempenho do que as instâncias da classe *basic*.

Ao avaliar as informações descritas na Tabela 1, percebe-se que as máquinas são constituídas de configurações de hardware díspares, mas, apesar desta heterogeneidade, todas as máquinas virtuais possuem a mesma quantidade de núcleos de processamento (oito) a fim de permitir uma justa comparação de desempenho. Todavia, é preciso enfatizar que a máquina física HP possui doze núcleos de processamento e, por este motivo, o seu sistema operacional foi configurado para trabalhar apenas com oito núcleos. Essa medida permitiu equalizar a quantidade de *cores* desse servidor com o número de núcleos das demais máquinas utilizadas nos experimentos.

Com a finalidade de entender mais profundamente o desempenho esperado por cada máquina, a Tabela 2 apresenta alguns detalhes dos processadores usados na máquina física e nas instâncias virtuais. A primeira questão a ser observada é que os processadores das máquinas virtuais do EC2 tendem a obter um maior desempenho, visto que apresentam um maior tamanho de *cache*, uma maior quantidade de canais de memória e, por fim, operam em um barramento (*Front Side Bus* - FSB) mais veloz. Os detalhes presentes na Tabela 2 também revelam que o processador utilizado pela máquina virtual da Azure é considerado como sendo inferior aos das demais máquinas, já que opera em uma frequência mais baixa e possui um tamanho de *cache* muito inferior.

Quanto ao software, foi utilizado em todas as máquinas o mesmo sistema operacional e as mesmas versões de ferramentas e bibliotecas a fim de não haver qualquer

influência dessas questões no desempenho alcançado. O sistema operacional utilizado foi o Ubuntu Linux 4.8.2 com *kernel* versão 3.13.0 e com *glibc* versão 2.40.0-2. A implementação do padrão MPI utilizada pelo MUFITS foi o MPICH2 na versão 3.0.4.

Tabela 2. Detalhes dos processadores utilizados pelas máquinas

| Máquina | Processador | Cache | FSB | Canais de memória |
|------------|-----------------------|-------|---------|-------------------|
| HP DL360G7 | Xeon X5650 2.66GHz | 12MB | 6.4GT/s | 3 |
| C3.2XLARGE | Xeon E5-2680 2.80GHz | 20MB | 8GT/s | 4 |
| M3.2XLARGE | Xeon E5-2670 2.50GHz | 20MB | 8GT/s | 4 |
| A4 | Opteron 4171HE 2.1GHz | 6MB | 6.4GT/s | — |

3.2. Experimentos com o Simulador MUFITS

Antes de descrever esta primeira etapa, é interessante fornecer algumas informações primordiais acerca do simulador de reservatórios MUFITS. Basicamente, este aplicativo executa uma leitura inicial de arquivos em que são obtidas informações para (i) compor o modelo representativo do reservatório a ser simulado, (ii) identificar os parâmetros da simulação, (iii) carregar o modelo matemático de simulação e (iv) definir as etapas e o tempo total de simulação. Em geral, uma simulação típica pode demorar várias horas ou dias para ser concluída, mas, por questões práticas e sem perda de generalidade, a simulação para estes experimentos foi redimensionada para ser executada em uma ordem de minutos. Por fim, é interessante mencionar que esse aplicativo é considerado como sendo *cpu-bound*, já que apresenta um maior uso de CPU em detrimento de uma alta carga de acesso a disco.

Nesta primeira etapa, foi analisado o tempo de execução do MUFITS ao rodar em cada uma das máquinas descritas anteriormente. Além do tempo de execução mensurado, foram obtidos dados específicos a respeito dos parâmetros do sistema operacional, isto é, durante a execução do MUFITS foram coletadas informações acerca dos subsistemas de CPU, memória e disco. Estes dados foram obtidos através de uma ferramenta chamada System Activity Report (SAR) que é capaz de interpretar as informações providas pelo Linux e traduzi-las em métricas relacionadas aos recursos utilizados pelo sistema operacional [Popiolek and Mendizabal 2013].

As métricas coletadas com o auxílio do SAR em relação ao subsistema de memória foram taxa média de páginas movidas por segundo da memória para o disco (*page out*) e a taxa média de páginas movidas por segundo do disco para a memória (*page in*). Quanto à CPU, foi obtido o percentual de tempo médio em que este recurso estava (i) ocioso, (ii) executando tarefas de usuário, (iii) aguardando a conclusão de operações de entrada e saída (disco, mais precisamente) ou (iv) executando tarefas do sistema operacional. Por fim, o subsistema de disco foi monitorado quanto a taxa média de blocos lidos e gravados por segundo.

3.2.1. Resultados dos Experimentos Computacionais com o MUFITS

O primeiro ponto a ser considerado na análise de resultados dos experimentos com o MUFITS é o tempo de execução obtido em cada um dos ambientes computacionais descritos.

Portanto, a Figura 1 ilustra, em segundos, o tempo de execução do MUFITS em cada uma das quatro máquinas.

A primeira e evidente constatação que pode ser feita a partir dos dados da Figura 1 é que a execução do MUFITS na máquina HP foi mais veloz do que em qualquer uma das máquinas providas pela nuvem. Esse menor tempo de execução obtido pela máquina física advém do fato de que este equipamento apresentou um melhor desempenho em todas as métricas que foram observadas. Primeiramente, quanto ao uso de CPU, pode-se observar na Figura 2 que o percentual de tempo gasto pelo processador em tarefas de usuário é superior ao valores atingidos por todas as máquinas virtuais. Da mesma forma, o tempo gasto pela CPU da máquina física com tarefas de sistema (também ilustrado na Figura 2) foi o menor dentre todas as máquinas. Esses resultados permitem concluir que o processador foi usado mais eficientemente na máquina física do que nas máquinas virtuais, já que o tempo gasto com uma carga útil de tarefas foi maior na máquina HP. Todavia, é preciso observar que o percentual atingido pela máquina virtual da Azure ficou bem próximo da máquina física. Apesar de ter se equiparado com a máquina física nesta métrica, há outros fatores que serão discutidos adiante que demonstram o porquê da máquina virtual da Azure ter tido desempenho tão inferior as demais.

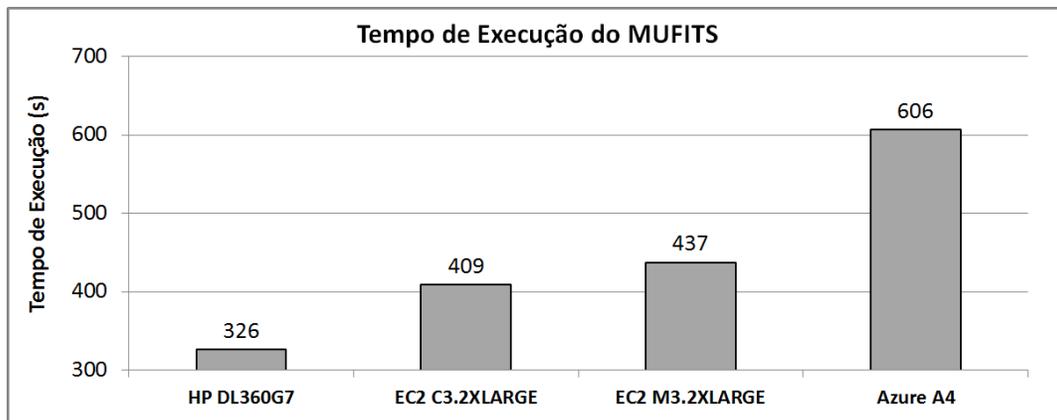


Figura 1. Tempo de Execução do MUFITS

Da mesma forma que o tempo de CPU, as métricas observadas quanto ao processo de paginação indicam que a vazão de dados transmitidos da memória para o disco (e vice-versa) foi muito maior na máquina física conforme ilustra a Figura 3a. Percentualmente, o desempenho da máquina física para páginas movidas para o disco (*page out*) e movidas do disco (*page in*) foram, respectivamente, 24% e 26% superior ao segundo melhor resultado (obtido pela C3.2XLARGE). Este fato demonstra que o sistema operacional presente na máquina HP conseguiu realizar um acesso mais eficiente à memória principal e ao disco, já que foi possível transferir, em uma mesma quantidade de tempo, uma maior quantidade de dados entre esses dois recursos.

Por fim, apesar do MUFITS não ser uma aplicação *io-bound* como explicitado anteriormente, os resultados referentes a blocos lidos e gravados no disco (principalmente lidos) também apontam para um melhor desempenho da máquina física conforme ilustrado na Figura 3b. Há uma exceção interessante no desempenho de dados gravados no disco: o desempenho da Azure foi ligeiramente superior ao da máquina física. Uma das

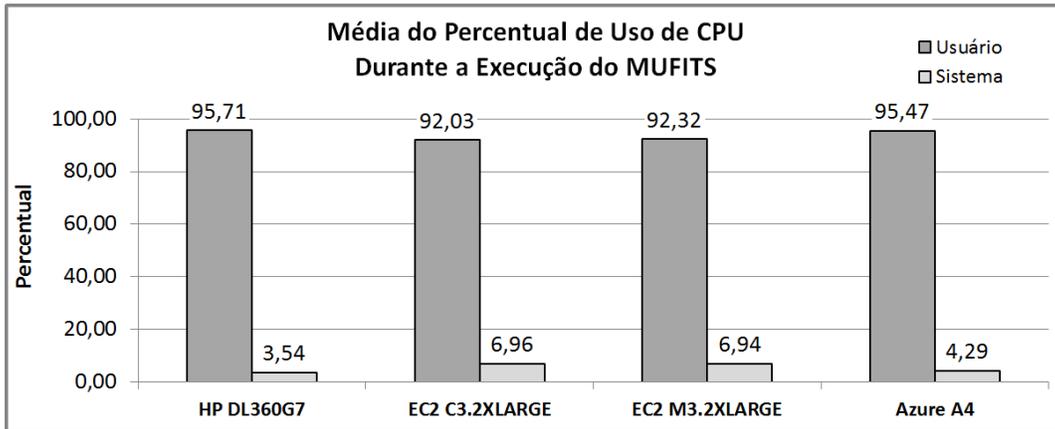


Figura 2. Média do percentual de uso de CPU durante a execução do MUFITS.

possíveis explicações para esse comportamento é o modo de *cache* de escrita habilitado nas instâncias da Azure. Neste modo, o sistema operacional recebe a confirmação de uma solicitação de gravação de dados no disco, apesar deste processo ainda não ter sido realizado efetivamente. Como resultado, a vazão de escrita no disco é aumentada, sendo que há um grande risco de perda de dados caso ocorra desligamento abrupto do equipamento físico em que a máquina virtual está hospedada.

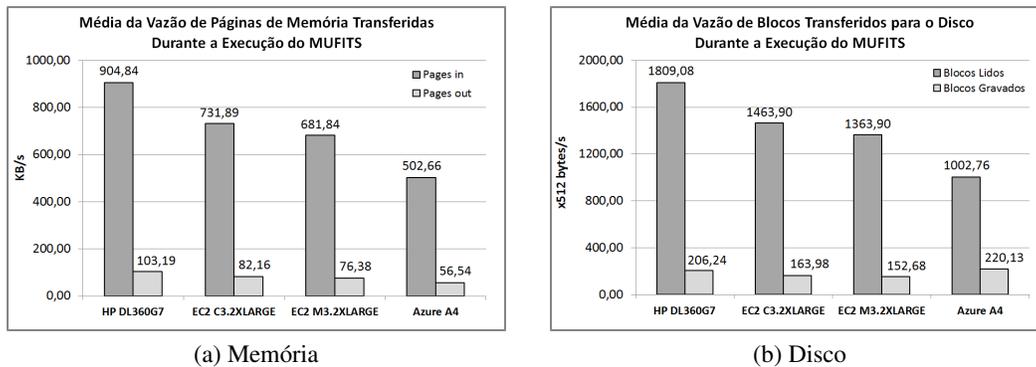


Figura 3. Desempenho de disco e memória durante a execução do MUFITS.

Enfim, apesar de ter um processador considerado inferior aos das máquinas virtuais do EC2, a máquina HP conseguiu alcançar um desempenho superior quanto ao tempo de execução do MUFITS. Acredita-se que esse comportamento esteja relacionado tanto ao *overhead* introduzido pelo software (*hypervisor*) de virtualização quanto ao compartilhamento de recursos (disco, processador, memória) entre diversas máquinas virtuais que estão sendo executadas simultaneamente em um mesmo equipamento físico [Jackson et al. 2010].

Outro fato que pode ser observado nos resultados de tempo de execução do MUFITS (conforme Figura 1), é o desempenho semelhante entre as máquinas virtuais do EC2, apesar de uma ser apenas de propósito geral e a outra específica para computação. Essa proximidade no desempenho alcançado justifica-se pela pequena diferença apresentada nos resultados que foram obtidos por ambas as máquinas virtuais em todos os quesitos

avaliados. Considerando que ambas as máquinas pertencem ao mesmo ambiente provido pela Amazon e que possuem quase todos os mesmos componentes de hardware, resta observar que a principal justificativa para essa leve disparidade no desempenho reside na diferença entre os processadores utilizados pelas instâncias. De acordo com a Tabela 2, ambos os processadores das máquinas EC2 são muito semelhantes, exceto pela frequência de operação que é maior no caso do processador utilizado pela C3.2XLARGE. Portanto, pode-se dizer que o melhor desempenho alcançado pela máquina C3.2XLARGE está ligado a um maior poder computacional oferecido pelo processador Xeon E5-2680 em detrimento ao Xeon E5-2670.

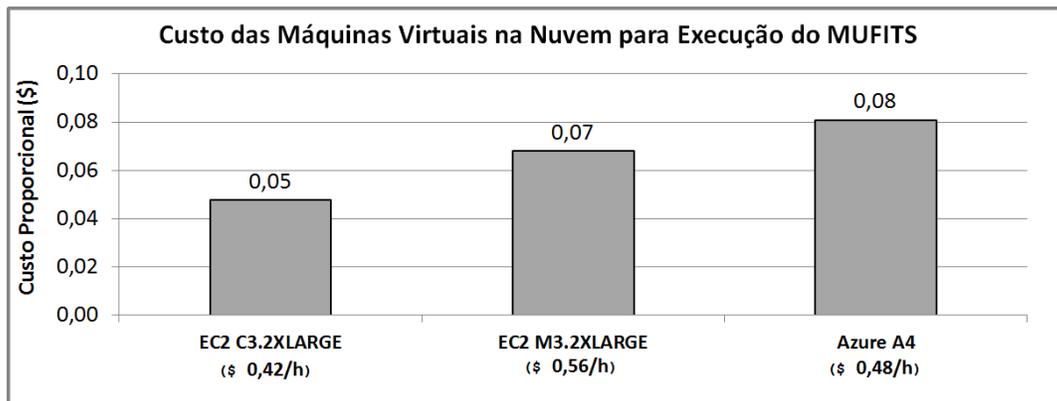


Figura 4. Custo de utilização da execução do MUFITS na nuvem.

Por último, os resultados do tempo de execução do MUFITS também revelam que a máquina virtual da Azure obteve o pior desempenho dentre todas as outras. Os resultados de acesso a disco (leitura) e paginação também apontam nesta mesma direção, já que a A4 apresentou sempre o pior desempenho. Realizando uma comparação direta com as máquinas EC2, pode-se constatar que outro fator além das características da nuvem (virtualização e compartilhamento de recursos) contribuiu para um desempenho tão ruim, pois, do contrário, tanto as máquinas da Amazon quanto a A4 teriam atingido um desempenho semelhante. Portanto, como esperado, uma das principais razões para esse comportamento é o baixo poder computacional oferecido pelo processador Opteron 4171HE em detrimento dos processadores Xeon das máquinas EC2.

Além da análise de desempenho, também foi considerado o custo envolvido para rodar o MUFITS na nuvem. Portanto, a Figura 4 apresenta o custo proporcional de execução do MUFITS em cada máquina virtual e, entre parênteses, o custo unitário de utilização por hora de cada uma das instâncias. Vale ressaltar que o custo unitário refere-se a máquinas alugadas nos EUA. Os resultados permitem comprovar que a máquina C3.2XLARGE foi a que apresentou um melhor custo-benefício, pois, além de ter um custo unitário menor, foi a máquina virtual que conseguiu concluir a execução do MUFITS no menor tempo possível. Outra questão que pode ser observada neste comparativo de custos é que, apesar da máquina M3.2XLARGE possuir um custo unitário maior do que a A4, foi mais vantajoso usar a máquina da Amazon para executar o MUFITS do que a instância da Microsoft. Claramente, isso se deve ao fato de que o tempo de execução alcançado pela M3.2XLARGE foi bastante inferior ao da A4.

3.3. Execução dos Testes de *Benchmark*

Concluídos os experimentos relacionados ao MUFITS, o conjunto de máquinas foi submetido a testes de *benchmark* específicos para avaliar o desempenho dos recursos de memória e processador a fim de complementar as explicações apresentadas na etapa anterior. Não foram realizados testes de *benchmark* para disco, pois, como dito anteriormente, o MUFITS é um aplicativo que faz uso intenso de processador (*cpu-bound*) e não realiza grande quantidade de acesso a disco.

A ferramenta empregada para realizar os experimentos de *benchmark* foi a Phoenix Test Suite (PTS) [Deshane et al. 2008] que é considerada como uma das mais abrangentes plataformas de *benchmark* disponíveis para o sistema operacional Linux. O PTS é composto por uma série de testes que permite avaliar especificamente um dado recurso ou subsistema (memória, processador, disco e placa gráfica), ou até mesmo o sistema como um todo.

Para avaliar o desempenho do subsistema de processamento e memória foram escolhidos, respectivamente, os testes NAS Parallel Benchmark (NPB) [He et al. 2010] e Cachebench [Knight et al. 2012]. O NPB foi criado pela divisão de Supercomputação Avançada da NASA (NAS) para avaliar o desempenho de supercomputadores ao executar algoritmos relacionados à computação dinâmica de fluidos e tem sido frequentemente usado para avaliar o desempenho computacional dos serviços oferecidos pela nuvem. Este teste é composto por cinco *microkernels* e por três aplicações compactas. Optou-se neste trabalho pela aplicação compacta BT (*Block Tri-diagonal Solver*), já que o uso de *microkernels* para testar o desempenho de sistemas tem sido desencorajado [Hennessy and Patterson 2012]. Escolhida a aplicação ou *microkernel* a ser executado, a operação do NPB consiste em criar vários processos de acordo com o número de núcleos de processamento disponíveis e, ao concluir sua execução, informar a quantidade de milhões de operações por segundo (MOP/s) que fora alcançada.

Já o Cachebench foi desenvolvido para avaliar a hierarquia de memória de um sistema e mensurar, em megabytes por segundo (MB/s), a máxima vazão alcançada. Este teste oferece as opções de realizar operações de leitura, escrita ou uma tríade composta por operações consecutivas de leitura, modificação e escrita. Ao escolher uma das opções anteriores, o Cachebench é executado durante um período pré-fixado de tempo a fim de verificar a quantidade de dados que foram movimentados (de acordo com a opção escolhida) na memória principal.

3.3.1. Resultados dos Testes de *Benchmark*

Os resultados obtidos ao se executar os *benchmarks* descritos anteriormente estão apresentados na Figura 5. Os resultados do teste NPB (Figura 5a) refletiram o desempenho obtido por cada uma das máquinas ao executar o MUFITS. Em outras palavras, a máquina HP, que atingiu o menor tempo de execução do MUFITS, foi a que conseguiu alcançar a maior quantidade de operações por segundo no teste NPB, e assim por diante. Da mesma maneira, os resultados de escrita em memória principal também seguiram a mesma linha de desempenho de execução do MUFITS. No entanto, houve uma menor vazão nas operações de leitura realizadas pela máquina HP se comparado com ambas as máquinas da Amazon. Este comportamento se deve ao tamanho de *cache* dos processadores, a qual

é substancialmente maior nas máquinas virtuais da Amazon (20MB) do que na máquina física HP (12MB).

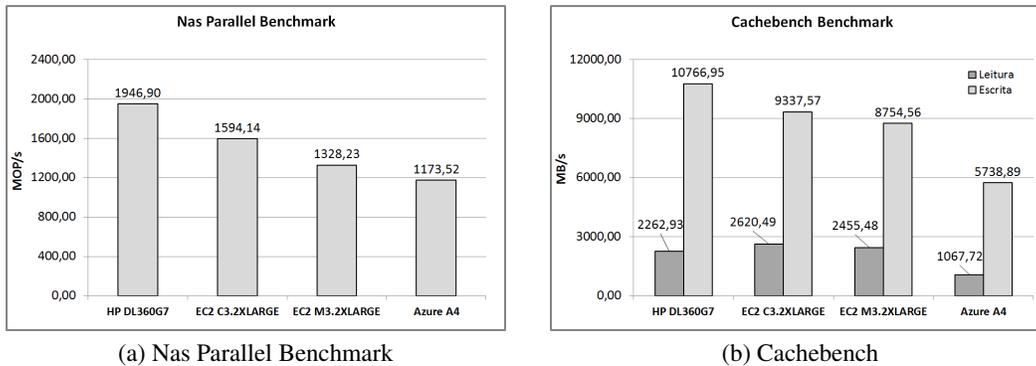


Figura 5. Testes de *benchmark*.

Esses resultados de *benchmark* confirmam a análise feita apenas sob a perspectiva de execução do MUFITS, ou seja, o compartilhamento de recursos e o *overhead* do software de virtualização causam um decréscimo significativo na capacidade computacional das máquinas virtuais. Essa condição é capaz de diminuir sensivelmente o desempenho de aplicações *cpu-bound* como o MUFITS.

4. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma análise de desempenho de um simulador de reservatórios de petróleo ao ser executado nos ambientes de computação em nuvem providos pela Amazon EC2 e Microsoft Azure. Os resultados demonstraram que a execução deste tipo de aplicação na nuvem ainda não apresenta um desempenho satisfatório. Esse comportamento deriva da concorrência de recursos originada pela execução de diversas máquinas virtuais no mesmo equipamento físico, além do próprio *overhead* introduzido pelo software de virtualização. Por conta dessas questões, a execução de computação científica no ambiente de *cloud* ainda não é encorajada, já que, para essas aplicações, é imprescindível obter os resultados no menor tempo possível. Uma das possibilidades de melhorar este desempenho é a avaliação do uso de nuvens privadas, onde seria possível exercer um maior controle sobre os recursos. Apesar de ainda não oferecer um desempenho satisfatório, é preciso avaliar a relação entre o custo e o tempo de execução obtido na nuvem, pois, para algumas aplicações, pode valer a pena esperar um pouco mais para obter o resultado ao passo que é possível alcançar uma significativa economia financeira.

Os próximos passos nesta direção de pesquisa incluem: (i) executar outras aplicações da área de petróleo a fim de avaliar outros quesitos como I/O e rede; (ii) realizar análise financeira mais abrangente levando em conta o custo total de uma infraestrutura tradicional e de um ambiente similar alocado na nuvem; (iii) avaliar um número maior de diferentes instâncias; (iv) considerar outros provedores de serviços de nuvem.

Referências

Afanasyev, A. A. (2013). Application of the reservoir simulator mufits for 3d modelling of co2 storage in geological formations. *Energy Procedia*, 40:365–374.

- Deshane, T., Shepherd, Z., Matthews, J., Ben-Yehuda, M., Shah, A., and Rao, B. (2008). Quantitative comparison of xen and kvm. *Xen Summit, Boston, MA, USA*, pages 1–2.
- He, Q., Zhou, S., Kobler, B., Duffy, D., and McGlynn, T. (2010). Case study for running hpc applications in public clouds. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 395–401. ACM.
- Heilmann, Z., Deidda, G., Satta, G., and Bonomi, E. (2013). Real-time imaging and data analysis for shallow seismic data using a cloud-computing portal. *Near Surface Geophysics*, 11(4):407–421.
- Hennessy, J. L. and Patterson, D. A. (2012). *Computer architecture: a quantitative approach*. Elsevier.
- Hoffa, C., Mehta, G., Freeman, T., Deelman, E., Keahey, K., Berriman, B., and Good, J. (2008). On the use of cloud computing for scientific workflows. In *eScience, 2008. eScience'08. IEEE Fourth International Conference on*, pages 640–645. IEEE.
- Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H. J., and Wright, N. J. (2010). Performance analysis of high performance computing applications on the amazon web services cloud. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 159–168. IEEE.
- Knight, D., Shams, K., Chang, G., and Soderstrom, T. (2012). Evaluating the efficacy of the cloud for cluster computation. In *Aerospace Conference, 2012 IEEE*, pages 1–10. IEEE.
- Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., and Epema, D. (2010). A performance analysis of ec2 cloud computing services for scientific computing. In *Cloud Computing*, pages 115–131. Springer.
- Peaceman, D. W. (2000). *Fundamentals of numerical reservoir simulation*. Elsevier.
- Popiolek, P. F. and Mendizabal, O. M. (2013). Monitoring and analysis of performance impact in virtualized environments. *Journal of Applied Computing Research*, 2(2):75–82.
- Ramakrishnan, L., Zbiegel, P. T., Campbell, S., Bradshaw, R., Canon, R. S., Coghlan, S., Sakrejda, I., Desai, N., Declerck, T., and Liu, A. (2011). Magellan: experiences from a science cloud. In *Proceedings of the 2nd international workshop on Scientific cloud computing*, pages 49–58. ACM.
- Subramanian, V., Ma, H., Wang, L., Lee, E.-J., and Chen, P. (2011). Rapid 3d seismic source inversion using windows azure and amazon ec2. In *Services (SERVICES), 2011 IEEE World Congress on*, pages 602–606. IEEE.
- Vecchiola, C., Pandey, S., and Buyya, R. (2009). High-performance cloud computing: A view of scientific applications. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 4–16. IEEE.
- Vöckler, J.-S., Juve, G., Deelman, E., Rynge, M., and Berriman, B. (2011). Experiences using cloud computing for a scientific workflow application. In *Proceedings of the 2nd international workshop on Scientific cloud computing*, pages 15–24. ACM.