

# Utilização de traços de execução para migração de aplicações para a nuvem

Eduardo Roloff, Otávio Carvalho, Alexandre Carissimi, Philippe Navaux

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{eroloff, omcarvalho, asc, navaux}@inf.ufrgs.br

**Abstract.** *Cloud computing has become a real alternative as a platform for scientific applications. The attractiveness of the cloud is in its pay-per-use model and easy access to large amounts of computational resources. However, existing applications must be migrated to the cloud platform before executing them. One of the challenges is to determine which applications perform satisfactorily in the cloud and which do not. In this context, our work proposes a methodology to analyze applications based on information in their execution traces. This information helps to decide which applications are potential candidates for migration and will perform well when running in cloud computing environments.*

**Resumo.** *A computação em nuvem tornou-se uma alternativa real para execução de aplicações científicas. Devido a suas características de cobrança apenas pela utilização e por prover acesso a grande quantidade de recursos computacionais. Porém as aplicações já existentes devem ser migradas. Um dos desafios é mapear quais aplicações terão desempenho satisfatório executando em nuvem e quais não.*

*Neste contexto, nosso trabalho propõe uma metodologia de análise de aplicações, baseada nas informações contidas nos traços de execução das mesmas. Essas informações auxiliam na decisão de quais aplicações são potenciais candidatas para migração e que apresentarão bom desempenho quando executadas em ambiente de computação em nuvem.*

## 1. Introdução

A necessidade de execução de grandes aplicações científicas é um dos motivadores para a busca de sistemas computacionais mais poderosos. Diversos modelos computacionais foram desenvolvidos ao longo do tempo e um dos fatores críticos para sua adoção em larga escala e também o seu posterior aprimoramento é se tais modelos são alternativas viáveis para a execução de aplicações científicas. Modelos computacionais como agregados de computadores e grades computacionais são dois exemplos recentes de novos modelos desenvolvidos e amplamente utilizados pela comunidade científica. Mas ambos os modelos possuem algumas características, como a necessidade de aquisição e manutenção de equipamentos, além de disponibilidade de espaço físico adequado para a instalação das máquinas e dispor de pessoal capacitado para operação. Tais características dificultam o uso desses modelos computacionais

A computação em nuvem é um modelo computacional recentemente desenvolvido e que vem atraindo muita atenção por parte da comunidade científica. Seu modelo foi

concebido com a combinação entre os modelos de processamento paralelo e distribuído, utilizando-se também conceito de serviços e grades computacionais. Podemos considerar que a computação em nuvem é um avanço dos modelos de agregados e grades que são utilizados atualmente para a execução de aplicações científicas.

O modelo de computação em nuvem parece ser a evolução natural para a execução de aplicações científicas. Uma importante questão é que, como os ambientes são virtualizados, existe o problema de perda de desempenho devido a essa virtualização [Huber et al. 2011] principalmente em determinados tipos de aplicações, como por exemplo, aplicações com grande comunicação entre processos ou *threads*. Outro fator que deve ser considerado é o esforço para a migração das aplicações para a nuvem, pois normalmente elas são aplicações grandes com muitas linhas de código e diversas configurações específicas. Portanto não é viável realizar a migração de uma aplicação antes de se realizar um estudo de como a mesma irá se comportar no novo ambiente.

Uma das maneiras de se realizar esse estudo de comportamento da aplicação é através da análise de traços de execução. Esse trabalho apresenta uma metodologia para a geração destes traços como auxílio na identificação de potenciais aplicações que podem ser migradas para a nuvem sem grande perda de desempenho. É feita a aplicação dos procedimentos em algumas aplicações de teste, executados em uma nuvem IaaS e os resultados são analisados afim de identificarmos padrões de comunicação, auxiliando assim a decidir qual aplicação poderá ser movida para a nuvem.

O restante desse trabalho está organizado da seguinte maneira. Na Seção 2 são apresentados os trabalhos relacionados a metodologias e análises de migração de aplicações para a nuvem. A Seção 3 apresenta os conceitos da plataforma JiT Cloud que é a plataforma onde nossos testes foram executados. Nossa metodologia de análise, bem como os procedimentos necessários para sua utilização, ambiente de execução e aplicações utilizadas são apresentados na Seção 4. A Seção 5 apresenta os resultados de nossos experimentos. Ao final, na Seção 6, são discutidos os resultados obtidos e comparados com um caso de migração real, também são apresentados os trabalhos futuros que serão realizados por nosso grupo.

## 2. Trabalhos relacionados

O trabalho proposto por Leymann et al [Leymann et al. 2011] apresenta uma metodologia e um conjunto de ferramentas para mover aplicações legadas para a sua execução em ambiente de nuvem. Essa metodologia é focada nos componentes e na estrutura da aplicação, decompondo a mesma em diversos módulos que posteriormente serão instalados na nuvem. Este trabalho não leva em conta o comportamento durante a execução da aplicação, em nosso estudo, o comportamento da aplicação é levado em conta para auxiliar na decisão de migração para a nuvem ou não.

O estudo feito por Louridas [Louridas 2010] abrange diversas características de computação em nuvem e explora os recursos disponíveis entre alguns dos maiores provedores de serviço. Inclusive é feito pelo autor um exemplo de desenvolvimento de aplicação utilizando a plataforma [force.com](http://www.salesforce.com)<sup>1</sup>. Seu trabalho possui um foco maior nas características dos provedores e não faz uma análise detalhada em qual tipo de aplicação será

---

<sup>1</sup><http://www.salesforce.com>

mais adequada em cada cenário. Nosso foco é em auxiliar o usuário a decidir a viabilidade de mover sua aplicação para a nuvem levando em conta as características de execução e comunicação da própria aplicação.

A metodologia desenvolvida por Mastelic et al [Mastelic et al. 2012] faz uma análise da migração de aplicações científicas para o ambiente de computação em nuvem. O grupo apresenta uma metodologia para comparação entre o ambiente de execução tradicional, como clusters, e a nuvem. São analisados aspectos como o desempenho da aplicação, o uso otimizado dos recursos, o consumo de energia e complexidade da configuração da aplicação em ambos os ambientes. O foco do estudo é a comparação da migração e da execução de uma aplicação para a nuvem. Porém não foi feito uma análise da aplicação antes da execução da metodologia. Nosso trabalho busca prover uma metodologia que dá suporte em nível de aplicação para verificar se a mesma tem potencial para migração para a nuvem. Caso a aplicação apresente um comportamento satisfatório, pode ser aplicada uma outra metodologia, como esta proposta, para medir o custo de migração bem como o custo de execução da mesma em ambientes de nuvem, tanto públicos como privados.

O trabalho realizado por Tran et al [Tran et al. 2011] realiza um estudo sobre a migração de aplicações para o ambiente de nuvem. Porém o foco do grupo de trabalho são os processos envolvidos para a construção de uma nova aplicação. Não são levados em conta nem o comportamento de uma possível aplicação pré-existente, nem todo o modelo de programação que uma aplicação já em uso possui. Este trabalho é válido do ponto de vista da criação de uma nova aplicação, mas no caso de uma migração de uma aplicação já existente, nossa abordagem é mais eficiente na identificação de uma aplicação com potencial para migração para a computação em nuvem.

### 3. JiT Cloud

Provedores de computação em nuvem, tanto públicos como privados, necessitam adquirir e configurar equipamentos além de arcar com os custos de eletricidade para funcionamento e refrigeração dos mesmos. Em termos de nuvens públicas esses custos estão contemplados no plano de negócio da empresa provedora. Em se tratando de nuvens privadas ou comunitárias, esses custos são de responsabilidade direta dos seus respectivos donos. Ou seja, os custos para se montar e operar uma nuvem comunitária ou privada são praticamente os mesmos de se operar sistemas tradicionais como aglomerados ou grades.

Considerando-se esses aspectos foi concebido o conceito de nuvens *Just in Time* (JiT) que combina os conceitos de computação em nuvem com a filosofia *Just in Time* da Toyota. Esse conceito foi apresentado no trabalho de Costa et al [Costa et al. 2010]. A idéia base é que se devem usar recursos disponíveis que estejam com baixa taxa de utilização, criando-se assim uma nuvem que pode ser acessada livremente pelos usuários que fazem parte do contexto. Os recursos podem ser disponibilizados e retirados da nuvem de acordo com a sua taxa de ocupação do objetivo principal deles, criando-se assim uma nuvem colaborativa. Além do mais podem ser utilizados recursos heterogêneos para a montagem dessa nuvem, tais como dispositivos móveis. A combinação de diversos recursos computacionais oriundos de locais diferentes e conectados através de um gerenciador comum forma uma JiT Cloud, conforme pode ser visto na Figura 1.

## 4. Ambiente de validação e metodologia

Para auxílio na tomada de decisão sobre qual aplicação poderá ser migrada para o ambiente de nuvem sem grande perda de desempenho, nós utilizamos uma metodologia que leva em conta a análise dos traços de execução da mesma, onde podem ser identificados gargalos na execução. Também apresentamos o ambiente no qual nossos testes foram executados. Por fim, apresentamos as características dos *benchmarks* que foram utilizados em nossa avaliação.

### 4.1. Procedimentos para visualização

Para a realização da instrumentação do código das aplicações, foi utilizado a suíte Score-P<sup>2</sup> que realiza a instrumentação de programas em MPI durante a compilação. Dessa maneira não é necessário que sejam feitas modificações no código fonte da aplicação em análise. Após a instrumentação do código é feita a execução da aplicação sem qualquer modificação nos procedimentos normais da mesma. Ao final da execução são gerados traços de execução que devem ser analisados através de uma ferramenta de visualização.

Como ferramenta de visualização dos traços de comportamento da aplicação, foi utilizada a ferramenta Pajé [Kergommeaux et al. 2000]. O Score-P gera traços em formato OTF2<sup>3</sup> que são diferentes dos traços no formato entendido pelo Pajé<sup>4</sup> então os mesmos necessitam ser formatados para poderem ser lidos.

Para a conversão do formato OTF2 para o formato de traço do Pajé deve ser utilizado o conjunto de ferramentas Akypuera<sup>5</sup>, especificamente a ferramenta otf22paje<sup>6</sup>.

<sup>2</sup><http://www.vi-hps.org/projects/score-p/>

<sup>3</sup><https://silc.zih.tu-dresden.de/otf2-current/html/>

<sup>4</sup><http://paje.sourceforge.net/download/publication/lang-paje.pdf>

<sup>5</sup><https://github.com/schnorr/akypuera>

<sup>6</sup><https://github.com/schnorr/akypuera/wiki/OTF2WithAkypuera>

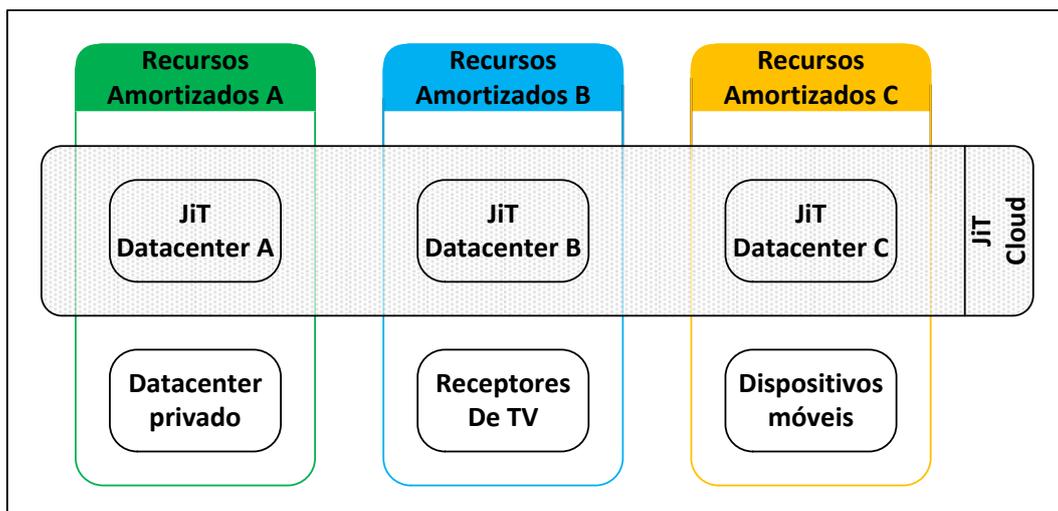
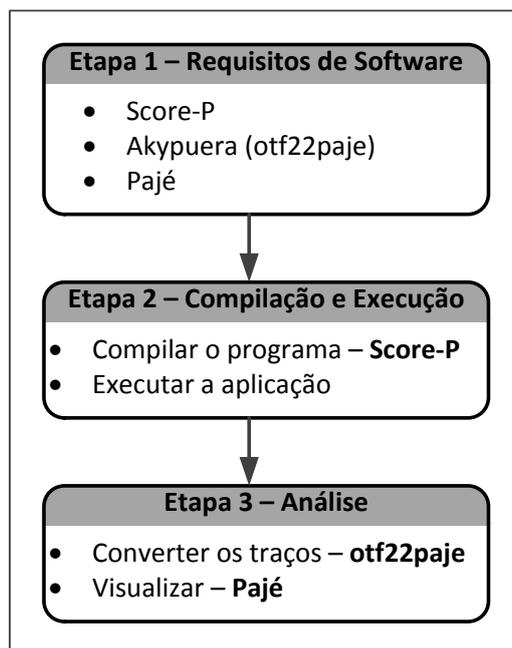


Figura 1. Composição de uma nuvem JiT



**Figura 2. Etapas para visualização.**

Após a conversão concluída, os traços podem ser visualizados com o Pajé. A Figura 2 ilustra os procedimentos necessários para a visualização.

#### **4.2. Ambiente utilizado**

A execução de nossos experimentos foi realizada na nuvem privada do Grupo de Processamento Paralelo e Distribuído da UFRGS. Essa nuvem é composta por sete servidores físicos, cada um possuindo dois processadores Intel® Xeon® E5310 com *clock* de 1.60 GHz e 8MB de memória cache, e memória principal de 16 GB. A interconexão de rede é através de Giga-Ethernet. Como Sistema Operacional, cada servidor executa o Ubuntu GNU/Linux 12.04.

Como ambiente gerenciador da nuvem, está sendo usado o sistema *Eucalyptus Cloud* <sup>7</sup> na versão 3.1.2. Um dos servidores é usado exclusivamente como nodo de controle, *Cluster Controller* (CC) na denominação do Eucalyptus, e as outras seis máquinas são os nós disponíveis para alocação de recursos dos usuários, *Node Controller* (NC). Essa nuvem está conectada a uma implementação de JiT Cloud proposta pela Universidade Federal de Campina Grande (UFCG) <sup>8</sup> que possui o nó centralizador desse projeto de gerenciador de nuvens.

Para a execução dos experimentos foram usados quatro dos seis nós disponíveis. Em cada um dos nós foi provisionada uma máquina virtual com 4 *cores* e 14 GB de memória. Como Sistema Operacional das máquinas virtuais foi escolhida a distribuição GNU/Linux Fedora 17, pois a versão de Eucalyptus usada disponibiliza esta imagem em

<sup>7</sup><http://www.eucalyptus.com>

<sup>8</sup><http://redmine.lsd.ufcg.edu.br/projects/jitclouds>

**Tabela 1. Visão geral dos *benchmarks* do NAS usados em nossa avaliação.**

Nome	Descrição	Foco	Linguagem
BT	Bloco Tridiagonal	Ponto flutuante	Fortran
CG	Gradiente Conjugado	Comunicação Irregular	Fortran
DT	Tráfego de dados	Comunicação	C
EP	Desordenamento paralelo	Ponto flutuante	Fortran
FT	Transformação de Fourier	Equações diferenciais	C
IS	Ordenamento de inteiros	Números inteiros	C
LU	Cálculo triangular	Comunicação regular	Fortran
MG	Multigrid	Comunicação regular	Fortran
SP	Pentadiagonal escalar	Ponto flutuante	Fortran

sua configuração padrão. Como biblioteca MPI foi usado o MPICH2 na versão 3.0.2. A instrumentação da aplicação foi feita com a utilização da ferramenta Score-P na versão 1.0.2. A aplicação utilizada foi o conjunto de benchmarks NPB (NAS) na versão 3.3. Como ferramenta de visualização dos traços gerado foi usado o Pajé em sua versão 1.98.

### 4.3. Benchmarks usados na avaliação

Para realizar a avaliação dos traços de execução, nós utilizamos a versão MPI do *Numerical Aerodynamic Simulation Parallel Benchmarks* (NPB)[Bailey et al. 1995], em sua versão 3.3.1. Esse conjunto de *benchmarks* é formado por diversas aplicações que executam métodos numéricos usados em simulações aerodinâmicas para computação científica. São nove diferentes testes, com comportamentos diferentes. A Tabela 1 fornece uma visão geral das aplicações.

Os testes BT, SP e LU possuem seu paralelismo expressado com a divisão do domínio do problema, enquanto compartilham dados nas bordas de cada subdomínio, mostrando assim acesso linear aos dados. O teste CG apresenta características de acesso randômico aos dados. A aplicação MG trabalha com dados contínuos, mas varia o padrão de acesso a esses dados em cada etapa da execução, cobrindo desde acessos totalmente desalinhados até acessos totalmente lineares. Essa aplicação pode ser considerada como intermediária entre as que apresentam acesso regular (BT, LU e SP) e as de acesso irregular (CG). O teste DT faz comunicação entre processos no esquema produtor consumidor usando uma matriz de comunicação. O EP é completamente paralelo e possui pouco acesso a dados, seu desempenho pode ser utilizado como referência para o pico de desempenho de um sistema computacional. IS utiliza apenas cálculo com números inteiros sem o uso de ponto flutuantes, possui nível médio de comunicação entre os processos. O FT é uma pseudo aplicação que resolve equações diferenciais usando a transformação de Fourier.

## 5. Resultados

Foram gerados traços de execução usando todas as aplicações que fazem parte do conjunto de *benchmarks* do NAS. Primeiramente foi feita uma execução para realizar a calibragem do Score-P e após isso, foi feita uma execução de cada aplicação com 16 processos executando-se 4 em cada máquina virtual. Após obtermos os traços do Score-P,

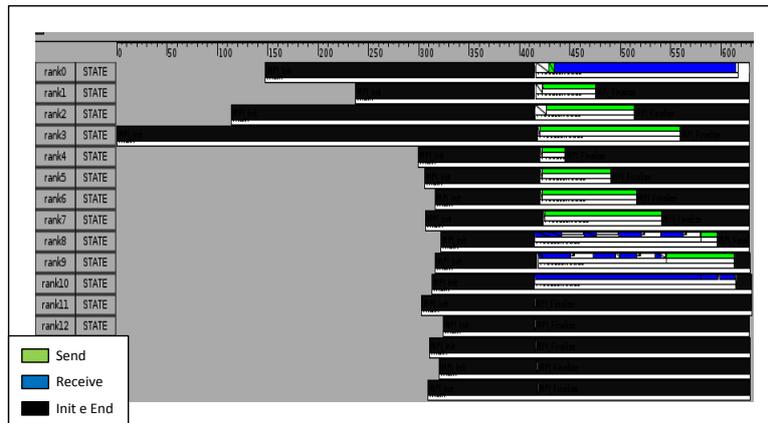


Figura 3. Traço de execução do DT

foi usado o conversor do Akypuera para converter para o formato do Pajé. Geramos duas massas distintas, uma somente com os traços relativos ao MPI, que foi utilizada somente como auxílio na identificação das comunicações entre processos e também barreiras da aplicação, e outra com todos os dados relativos a execução da aplicação.

Como legenda de cores, temos as operações de inicialização e finalização do MPI plotadas na cor preta. Operações de envio de dados estão plotadas em verde e de recepção de dados estão em azul. Barreiras de sincronização e operações de *wait* estão representadas na cor vermelha. Reduções estão plotadas em amarelo e comunicações do tipo todos para todos de marrom. As áreas brancas do gráfico são a representação da computação propriamente dita.

Podemos dividir os padrões de comunicação encontrados em três grupos (1) aplicações focadas somente em comunicação, (2) aplicações com interpolação entre computação e comunicação e (3) aplicações com foco em computação.

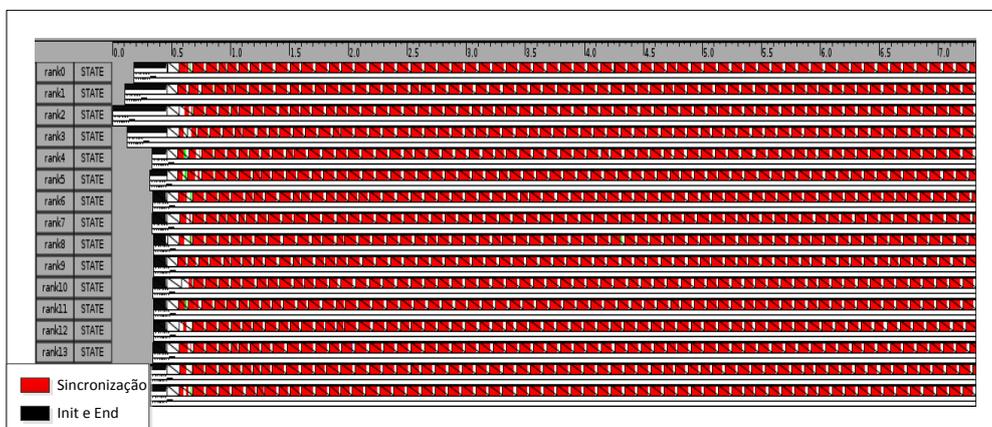


Figura 4. Traço de execução do SP

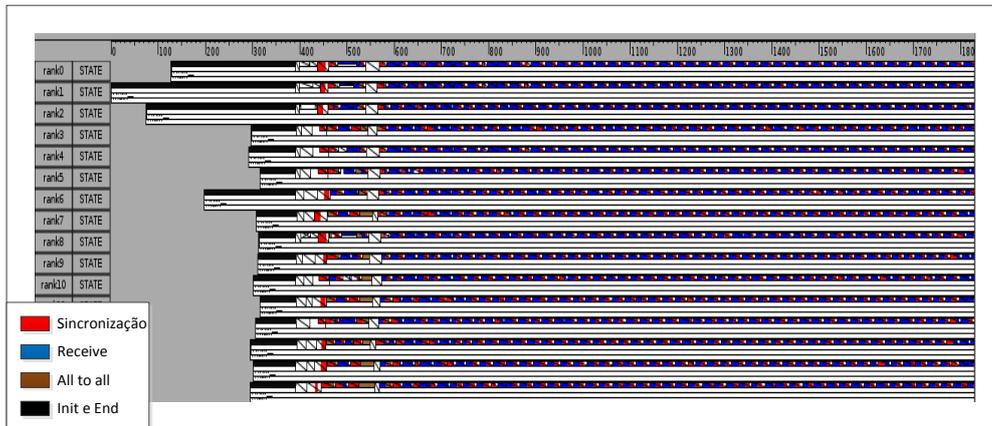


Figura 5. Traço de execução do LU

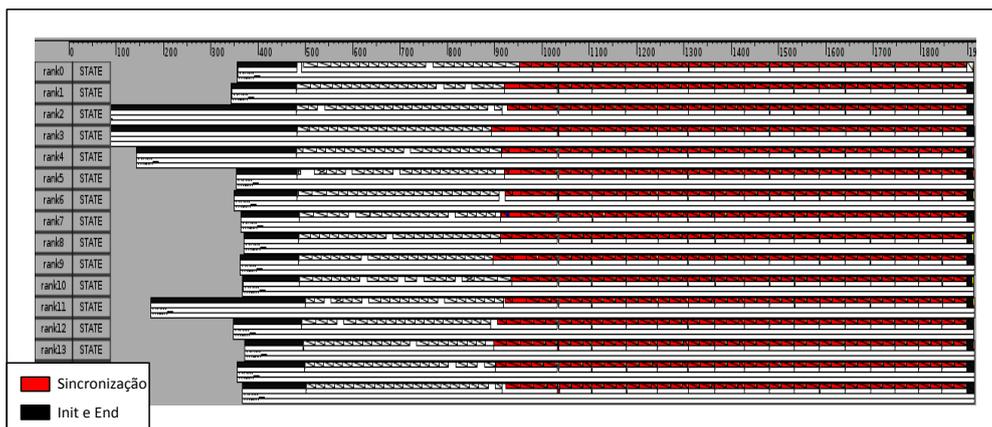


Figura 6. Traço de execução do CG

No primeiro grupo, temos somente a aplicação DT, tendo seu traço de execução representado na Figura 3. Pode-se observar que a aplicação tem um curto tempo de execução, levando mais tempo na inicialização e finalização do MPI do que na carga útil em si. Ou seja, aplicações focadas em fluxo de dados, como gateways, cujo desempenho depende diretamente da velocidade da rede não terão bom desempenho em ambientes de nuvem.

O segundo grupo pode ser dividido em duas categorias, as aplicações que realizam muitas operações bloqueantes, como envio de dados bloqueantes e barreiras; e as aplicações que fazem muita comunicação entre todos os processos.

As aplicações que realizam operações bloqueantes são caracterizadas por ter predominância da cor vermelha ou azul em seus traços. Esse grupo é composto por quatro aplicações SP representado na Figura 4, LU representado na Figura 5, CG representado na Figura 6 e BT. Nessas figuras, podemos perceber que as três aplicações realizam comunicação e operações bloqueantes durante todo o seu tempo de execução, sendo que quanto



Figura 7. Traço de execução do FT

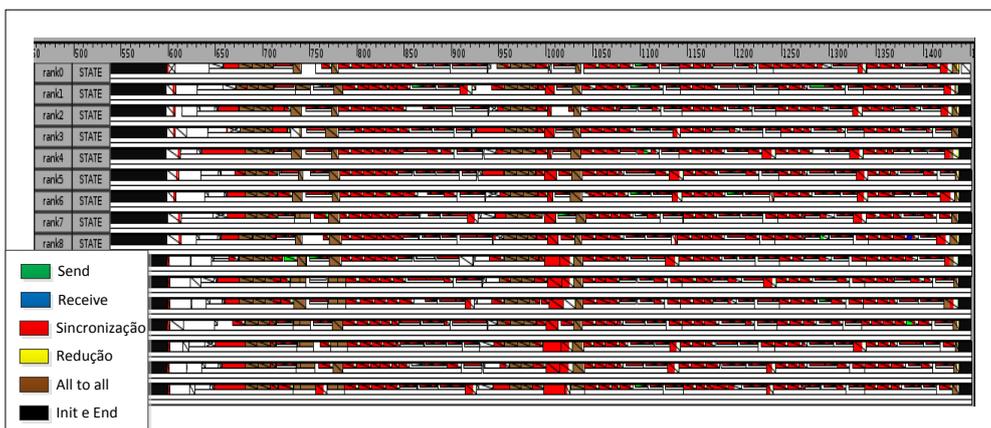


Figura 8. Traço de execução do MG

maior for a área colorida, maior é o tempo dispendido nessas operações. Podemos notar que o SP e o LU realizam comunicação durante todo o período de cálculo, já o CG possui cerca de um terço de sua execução como computação independente entre os processos para só então começar a comunicação.

Aplicações com comportando como o CG e o LU são candidatas a migração para o ambiente de nuvem, pois apesar de possuírem elevado grau de comunicação, a área da figura que representa a computação é maior do que a comunicação, o que representa um maior tempo investido em computação. Aplicações com comportante semelhante ao SP estão no limite da migração, pois praticamente passam metade do tempo de execução comunicação ou bloqueadas esperando dados. Nessas aplicações convém uma análise mais profunda dos traços antes de se optar pela migração.

A segunda categoria é composta pelas aplicações FT, Figura 7, e MG, Figura 8, essas aplicações realizam muitas reduções e comunicações do tipo todos para todos durante a sua execução. Esse tipo de operação é extremamente oneroso em termos de rede

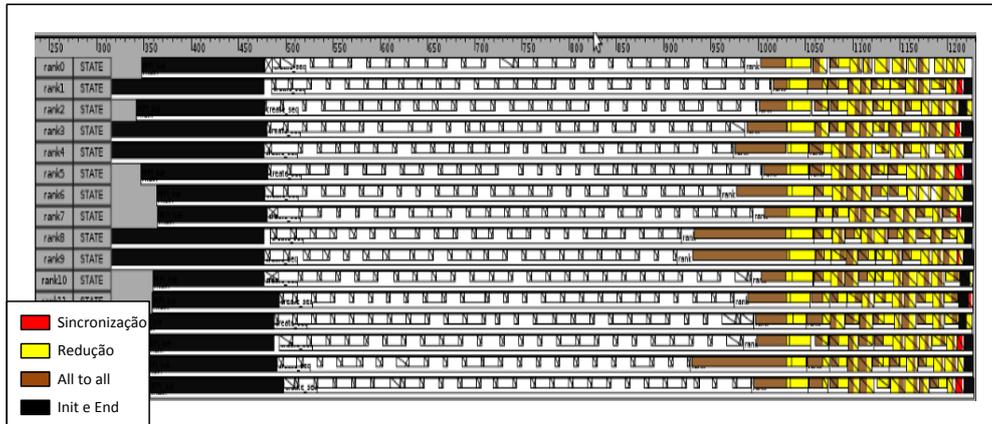


Figura 9. Traço de execução do IS

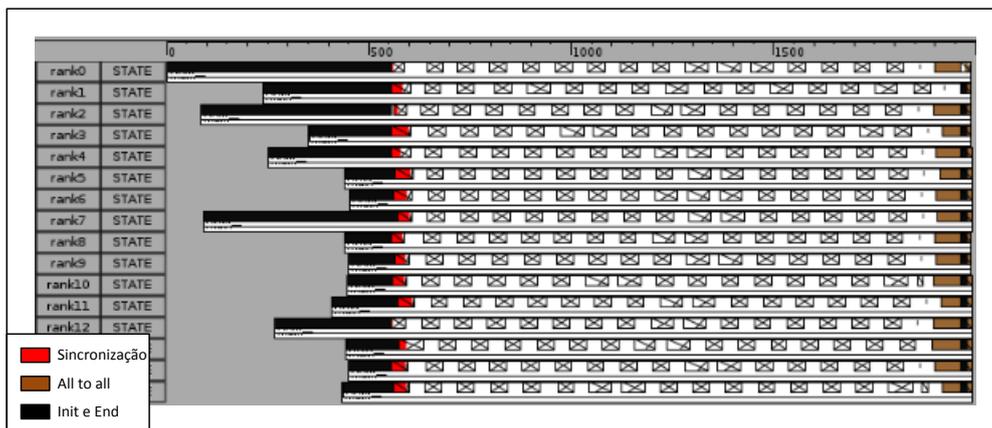


Figura 10. Traço de execução do EP

e não é recomendada a migração desse tipo de aplicação.

Por fim temos as aplicações que realizam basicamente a computação durante a grande maioria de sua execução e ao final realizam comunicações todos para todos ou reduções, somente para fins de consolidação dos resultados. Esse grupo é composto pelas aplicações IS e EP, representados nas Figuras 9 e 10 respectivamente.

Esse tipo de aplicação é a ideal de ser migrada para o ambiente de nuvem, pois não possui dependência de rede para execução de sua computação e terá os benefícios de computação em nuvem, principalmente o acesso a um conjunto grande de recursos. Essas aplicação se beneficiam quando são aplicações que executam o cálculo durante muito tempo e fazem a redução ao final, quanto maior o tempo de execução sem comunicação, maior será o benefício.

A análise dos padrões de comunicação e computação acima e as devidas conclusões sobre quais aplicações são boas candidatas para migração para a nuvem foram

baseadas nos trabalhos de Church e Goscinski [Church and Goscinski 2011] e no trabalho de Wang e Ng [Wang and Ng 2010] e também em parte dos resultados de performance de nosso trabalho anterior que envolveu a migração desses benchmarks para ambientes de nuvem públicos [Roloff et al. 2012].

## 6. Conclusões e trabalhos futuros

A computação em nuvem é um modelo interessante para a execução de grandes aplicações científicas, pois provê acesso a recursos virtualmente ilimitados sem qualquer custo de aquisição ou manutenção física. Mas é necessário um esforço de migração das aplicações para execução nesse modelo. Este trabalho propôs uma metodologia com a utilização de traços de execução para auxílio na decisão de qual tipo de aplicação pode ser migrada para a nuvem com sucesso e qual não.

Podemos concluir que a análise de traços de execução é uma ferramenta que possibilita identificar o comportamento de aplicações sem qualquer mudança em nível de código fonte ou de ambiente de execução. Com a análise dos traços gerados, pode-se perceber os padrões de execução da aplicação e com base nisso, e nos resultados apresentados nesse trabalho, pode-se identificar potenciais aplicações a serem migradas.

Como trabalhos futuros, pretendemos realizar mais experimentos de análise de traços e, com isso, gerarmos métricas para migração de aplicações. Também será interessante gerar uma ferramenta que faça a análise automática dos traços de execução e, com base nas métricas definidas, gere um relatório de compatibilidade.

## Agradecimentos

Este trabalho foi parcialmente financiado pela RNP e Microsoft.

## Referências

- Bailey, D., Harris, T., Saphir, W., Van Der Wijngaart, R., Woo, A., and Yarrow, M. (1995). The NAS parallel benchmarks 2.0. NASA Ames Research Center.
- Church, P. and Goscinski, A. (2011). IaaS clouds vs. clusters for hpc : a performance study. In *International Conference on Cloud Computing, GRIDS, and Virtualization*, pages 39–45.
- Costa, R., Brasileiro, F., Souza Filho, G. L., and Sousa, D. M. (2010). Just in time clouds: Enabling highly-elastic public clouds over low scale amortized resources. Tech. Rep. TR-3.
- Huber, N., von Quast, M., Hauck, M., and Kounev, S. (2011). Evaluating and modeling virtualization performance overhead for cloud environments. In Leymann, F., Ivanov, I., van Sinderen, M., and Shishkov, B., editors, *CLOSER*, pages 563–573. SciTePress.
- Kergommeaux, J. C., Stein, B. O., and Bernard, P. (2000). Pajé, an interactive visualization tool for tuning multi-threaded parallel applications. *Parallel Computing*, pages 1253–1274.
- Leymann, F., Fehling, C., Mietzner, R., Nowak, A., and Dustadar, S. (2011). Moving applications to the cloud: An approach based on application model enrichment. *International Journal of Cooperative Information Systems*, 20(03):307–356.

- Louridas, P. (2010). Up in the air: Moving your applications to the cloud. *Software, IEEE*, 27(4):6–11.
- Mastelic, T., Lucanin, D., Ipp, A., and Brandic, I. (2012). Methodology for trade-off analysis when moving scientific applications to cloud. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 281–286.
- Roloff, E., Diener, M., Carissimi, A., and Navaux, P. (2012). High performance computing in the cloud: Deployment, performance and cost efficiency. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 371–378.
- Tran, V., Keung, J., Liu, A., and Fekete, A. (2011). Application migration to cloud: a taxonomy of critical factors. In *Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing, SECCLOUD '11*, pages 22–28, New York, NY, USA. ACM.
- Wang, G. and Ng, T. (2010). The impact of virtualization on network performance of amazon ec2 data center. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9.