

Uma Comparação de Múltiplas Políticas de Migração de Contêineres Suportadas pela Ferramenta CRIU

Leonel Feitosa¹, Vандirleya Barbosa¹, Arthur Sabino¹
Luiz Nelson Lima¹, Iure Fé¹, Bruno Silva² e Francisco Airtон Silva¹

¹PASID Lab, Universidade Federal de Piauí (UFPI)

²Microsoft Research

{leonelfeitosa, vандirleya.barbosa}@ufpi.edu.br

{arthursabino, luizznelson, iure.fe, faps}@ufpi.edu.br

brunosilva@microsoft.com

Resumo. A migração de contêineres entre servidores vem sendo utilizada para lidar com questões de desempenho e disponibilidade. Ferramentas como o Checkpoint Restoration In Userspace (CRIU) são utilizadas na migração de contêineres. Contudo, a seleção de uma política de migração adequada pode ser desafiadora. Nesse contexto, os modelos formais como rede de Petri estocástica (SPN) surgem como representações matemáticas para sistemas reais de migração de contêineres computacionais. Este artigo propõe dois modelos de redes de Petri estocásticas (SPN) com e sem estado absorvente. São avaliadas as métricas Migration Total Time (MTT), Mean Migration Time (MMT), probabilidade de descarte e utilização. O modelo com estado absorvente também permite calcular a função de distribuição de probabilidade acumulada (CDF).

Abstract. Migrating containers between servers is used to improve performance and availability issues. Tools such as Checkpoint Restoration In Userspace (CRIU) are used in container migration. However, selecting an appropriate migration policy can be challenging. In this context, formal models such as the stochastic Petri net (SPN) emerge as mathematical representations for real computational container migration systems. This paper proposes two stochastic Petri net (SPN) models with and without absorbing state. The analyzes Total Migration Time (MTT), Average Migration Time (MMT), probability of disposal and use are evaluated. The model with absorbing state also allows calculating the cumulative probability distribution function.

1. Introdução

A migração é uma funcionalidade essencial em plataformas de computação virtualizadas em grande escala. A migração de máquinas virtuais é comumente usada por gerentes de *data centers* como um facilitador para desativação programada de servidores, consolidação de recursos, recuperação de desastres, escalabilidade vertical, etc. [Kaur et al. 2023]. À medida que muitas aplicações estão migrando de infraestruturas baseadas em VM para infraestruturas baseadas em contêineres por razões de simplicidade,

desempenho e custo, técnicas semelhantes estão se tornando necessárias em ambientes de contêineres como uma ferramenta fundamental de gerenciamento de sistemas.

Para implementar a migração de contêineres em sistemas operacionais Linux existe o *Checkpoint Restoration In Userspace*¹. O CRIU é uma opção de utilitário usado para migração de contêineres e com uma comunidade de desenvolvimento bastante ativa. Este artigo se concentra em contêineres runC de código aberto da *Open Container Initiative*², com o CRIU selecionado como a ferramenta de migração. A ferramenta CRIU pausa o contêiner em execução e salva o estado atual em arquivos chamados de metadados. Os arquivos podem ser restaurados, permitindo que o contêiner retome a execução do estado salvo [Maheshwari et al. 2018]. O CRIU apresenta várias políticas para migração de contêiner, incluindo Cold, PreCopy, PostCopy e HybridCopy [Puliafito et al. 2019].

Cada política de migração de contêiner tem suas próprias regras, etapas de cópia e processamento. A escolha da política de migração de contêiner é complexa, pois diversos fatores podem influenciar a migração. Para minimizar impactos negativos de desempenho na migração, se faz necessário considerar não apenas o tamanho do elemento a ser migrado, mas também a quantidade de elementos migráveis, capacidade de migração paralela, probabilidade de falha e a taxa de chegada. Balancear essas variáveis é uma tarefa desafiadora, e experimentos reais podem exigir esforços significativos de tempo e recursos. Modelos analíticos podem ajudar a mitigar esse problema, fazendo previsões do desempenho da migração com base nos fatores relevantes ao processo de migração.

Existem poucos trabalhos com experimentos que abranjam os fatores que posam, de fato, influenciar migração de contêineres. Foram encontrados alguns trabalhos que avaliaram o desempenho de migração de contêineres executando experimentos reais [Puliafito et al. 2019, Benjaponpitak et al. 2020, Fan et al. 2019, Chou et al. 2019, Govindaraj and Artemenko 2018, Di et al. 2021, Machen et al. 2017, Torre et al. 2019, Karhula et al. 2019]. Porém, nenhum dos trabalhos elencados explorou a predição de desempenho com uso de modelagem analítica para migração de contêiner, mas sim, apenas para migração de máquinas virtuais [Silva et al. 2014]. A composição e modo de implantação das máquinas virtuais diferem bastante de contêineres [Joy 2015].

Este artigo propõe modelos de redes de Petri estocásticas (SPN) para representar e calcular cinco métricas: o migration total time (MTT), o mean migration time (MMT), o cumulative distribution function (CDF), a probabilidade de descarte e utilização. Dessa forma, administradores de *data centers* poderão planejar melhor o processo de migração, mesmo em estágios iniciais de design da arquitetura computacional a ser implantada. O artigo propõe dois modelos SPN que representam e avaliam quatro políticas de migração de contêiner, variando quantidade de elementos migráveis, capacidade de migração paralela, probabilidade de falha e taxa de chegada. O CDF também calculado provê a probabilidade de migração finalizar em determinada janela de tempo. As políticas de migração modeladas são: Cold, PreCopy, PostCopy e Hybrid. Por restrição de espaço do artigo sugerimos a leitura do artigo de [Puliafito et al. 2019] sobre o funcionamento de tais políticas. A estrutura restante deste trabalho segue o seguinte formato. Na Seção 2, serão discutidos os trabalhos relacionados. Os modelos SPN são descritos nas Seções 3 e 4. A conclusão do trabalho é exposta na Seção 5.

¹CRIU <https://criu.org/>

²Open Container Initiative <https://opencontainers.org/>

2. Trabalhos Relacionados

Esta seção aborda trabalhos na literatura que compartilham abordagens semelhantes a este estudo, focando na migração de contêineres computacionais. Foram selecionados trabalhos que tratam da migração de contêineres, com pelo menos, uma das políticas exploradas neste estudo. O trabalho de [Benjaponpitak et al. 2020] desenvolveu o CloudHopper, capaz de realizar a migração ao vivo de contêineres entre múltiplos provedores de nuvem. O trabalho de [Torre et al. 2019] apresenta um teste que analisa a migração de contêineres considerando diversas condições ajustáveis, como o tamanho da imagem, a capacidade de rede e a carga de CPU e RAM. O estudo de [Pecholt et al. 2021] propõe um novo conceito de instância e migração de contêineres OS ao vivo. O sistema assegura integridade e confidencialidade durante implantação, migração e repouso.

O estudo de [Puliafito et al. 2019] relata casos de uso de migração de contêineres na névoa e discute a containerização. Posteriormente, apresenta uma visão abrangente das técnicas de migração de contêineres. O trabalho de [Fan et al. 2019] apresenta um modelo de migração ao vivo de localidade para contêineres Docker, onde consideram fatores como a distância entre os contêineres, a largura de banda disponível e os custos associados. O trabalho de [Chou et al. 2019] apresentou uma abordagem de migração baseada em *checkpoints*, utilizando um *pool* global de memória não volátil compartilhada. O estudo de [Karhula et al. 2019] utiliza o Docker e a ferramenta CRIU para a realização de *checkpoints* e a suspensão de funções de longa duração. O estudo de [Ramanathan et al. 2021] propõe um *framework* para migrar componentes virtuais EPC em contêineres usando migração ao vivo, com análise experimental comparando VMs e contêineres.

A Tabela 1 compara pontos relevantes deste estudo com trabalhos da literatura. O primeiro critério abordado foi o **Método de avaliação**, sendo nosso artigo o único a apresentar modelos SPN para modelar múltiplas políticas de migração. O uso de modelos permite análises diversas sem a necessidade de uma estrutura física preexistente. Além disso, foi validado o modelo com estado absorvente comparando com experimento real. Diversas **Métricas** foram mapeadas na literatura no contexto de migração de contêineres, como *Migration Total Time* (MTT), *Mean Migration Time* (MMT), utilização e tempo de inatividade. Algumas destas métricas também foram exploradas neste estudo. Embora não tenhamos abordado o tempo de inatividade, propomos a taxa de descarte e de migração, métricas que não foram encontradas em outros trabalhos. As **Políticas exploradas** em cada estudo também foram abordadas. Nosso trabalho explora todas as políticas de migração mapeadas na literatura. Ao contrário de outros trabalhos, este é o único que realiza uma **Análise de falhas** utilizando uma variação da probabilidade de falha durante o processo de cópia para avaliar a taxa de migração dos contêineres. Outro diferencial é a **Análise do Cumulative Distribution Function (CDF)**, que possibilita calcular a probabilidade de toda a migração finalizar em uma determinada janela de tempo, referente ao tempo total de migração.

3. Modelo SPN Absorvente

A representação do modelo e a computação dos resultados das simulações foram obtidos com a ferramenta Mercury [Maciel et al. 2017]. A adoção de modelos SPN foi motivada

Tabela 1. Trabalhos relacionados.

Trabalho	Método de Avaliação	Métricas	Políticas Exploradas	Análise de Falhas	Análise do CDF
[Benjaponpitak et al. 2020]	Mensuração	MRT e vazão	PreCopy	×	×
[Torre et al. 2019]	Mensuração	MTT	Cold	×	×
[Pecholt et al. 2021]	Mensuração	MTT e tempo de inatividade	Cold, PreCopy e PostCopy	×	×
[Puliafito et al. 2019]	Mensuração	MTT e tempo de inatividade	Cold, PreCopy, PostCopy e Hybrid	×	×
[Fan et al. 2019]	Mensuração	MTT	PreCopy	×	×
[Chou et al. 2019]	Mensuração	MTT e tempo de inatividade	PreCopy e PostCopy	×	×
[Karhula et al. 2019]	Mensuração	Utilização	Cold	×	×
[Ramanathan et al. 2021]	Mensuração	MTT e tempo de inatividade	Cold e PreCopy	×	×
Este trabalho	Modelagem	MTT, MMT, MR, U, e PD	Cold, PreCopy, PostCopy e Hybrid	✓	✓

pela capacidade de generalização oferecida por tal modelagem. A modelagem permite explorar, a baixo custo, os impactos das variações de arquiteturas e parâmetros numericamente nas métricas de migração de contêineres [Jain 1991].

A Figura 1 mostra o modelo SPN de migração de contêineres com estado absorvente. Um modelo de estado absorvente possui a característica que, em algum momento, os tokens não mais se movimentarem no modelo, conforme será explicado. No nosso caso, isso significa K contêineres chegarem à máquina de destino e não se moverem mais. Os tokens simbolizam os contêineres e as transições representam as fases do processo de migração. O modelo condensa as quatro políticas de migração, no qual a escolha é realizada por meio da alteração do valor da variável *POLÍTICA*. Existe uma quantidade de elementos migráveis correspondentes à marcação K no lugar *Node_A*. Assim, os tokens se encontram inicialmente no *Node_A* e serão movidos para o *Node_B*. A migração somente ocorrerá caso tenha capacidade de migração no lugar *Cap*. O lugar *Node_B* é do tipo absorvente [Maciel et al. 1996]. Quando os tokens chegam ao destino *Node_B*, a migração está finalizada. O cálculo do CDF é feito a partir da probabilidade de todos os tokens presentes no *Node_A* chegarem ao *Node_B*. Com a CDF é possível estimar a probabilidade de terminar a execução antes de um tempo específico [$P(T < t)$] e a probabilidade de terminar a execução em um intervalo de tempo [$P(t1 < T < t2) = P(T < t2) - P(T \leq t1)$] [Bause and Kritzinger 2002].

A escolha de qual caminho o token segue se dá por condições de guarda presentes nas seguintes transições: *Dump1*, *PDump2*, *Dump3* e *PDump4*. As transições são ativadas conforme o estado da variável *POLÍTICA*. *POLÍTICA* igual a 1 refere-se à política de migração *Cold*. Se for a variável *POLÍTICA* for igual 2, o token seguirá pela política *PreCopy*. Se *POLÍTICA* for igual a 3, o token seguirá pela política *PostCopy*. Por fim, se a variável *POLÍTICA* for igual a 4, ela irá pelo caminho referente à política *Hybrid*. Todas as transições possuem distribuição de probabilidade exponencial, que foi condizente com o experimento de validação. Além disso, todas as transições possuem semântica *infinite server*, indicando que as transições podem ser disparadas com múltiplos tokens em

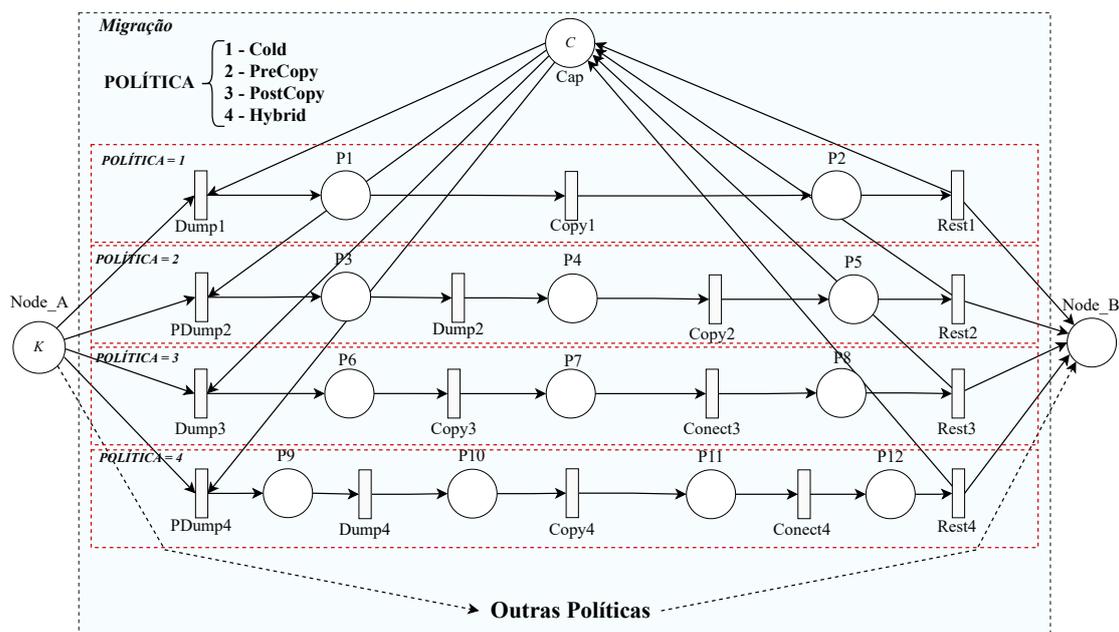


Figura 1. Modelo SPN com estado absorvente.

paralelo [Marsan et al. 1998]. A Tabela 2 apresenta os principais elementos do modelo.

Tabela 2. Descrição dos elementos principais do modelo.

Tipo	Elemento	Descrição
Lugares	Node_A	Host inicial contendo o container a ser migrado
	Node_B	Local para o será migrado o elemento de migração
	Cap	Quantidade de contêineres migráveis paralelamente
Transições Temporizadas	Dump1	Tempo associado a criação do ponto de restauração
	Copy1	Tempo associado a transferência dos metadados do elemento a ser migrado
	Rest1	Tempo associado a restauração do contêiner para a política de migração Cold
	PDump2	Tempo associado a verificação do aplicativo no nó de origem
	Dump2	Tempo associado à coleta das informações das páginas de memória moficadas
	Copy2	Tempo associado à transferência das informações das páginas de memória
	Rest2	Tempo associado a restauração do contêiner para a política de migração PreCopy
	Dump3	Tempo associado a criação do ponto de restauração
	Copy3	Tempo associado à transferência das informações das páginas de memória
	Conect3	Tempo associado ao tuneio de comunicação entre Nós
	Rest3	Tempo associado a restauração do contêiner para a política de migração PostCopy e transferência das páginas com falha
	PDump4	Tempo associado a verificação do aplicativo no nó de origem
Dump4	Tempo associado a criação do ponto de restauração	
Copy4	Tempo associado à transferência das informações das páginas de memória moficadas	
Conect4	Tempo associado a conexão do Node_A ao Node_B	
Rest4	Tempo associado a restauração do contêiner para a política de migração PostCopy e transferência das páginas com falha	
Marcações dos Lugares	K	Número de Contêineres a ser migrado
	C	Capacidade associada aos contêineres de migração

O Tempo Total de Migração (MTT) é o tempo médio para terminar a migração de todos os K elementos a serem migrados. Na teoria das redes de Petri, um estado absorvente é um estado que, após ele ser encontrado, nenhum outro estado pode ser alcançado. O tempo de absorção pode ser computado por meio da análise numérica da Cadeia de Markov de Tempo Contínuo (CTMC) gerada pela SPN [Nelson 2013] ou por meio da

simulação da SPN. No presente trabalho, utilizamos análise transiente, que, a partir de uma SPN gera uma CTMC e faz o cálculo do estado absorvente.

O comportamento da CTMC pode ser descrito pela equação de Kolmogorov dado o vetor de probabilidade inicial $\pi(0)$ (ver Equação 1). A Equação 2 fornece o tempo total esperado que o CTMC gasta no estado i durante o intervalo $[0, t)$. O tempo gasto antes da absorção pode ser calculado restrito aos estados do conjunto de estados não absorventes (N) por $\lim_{t \rightarrow \infty} L_N(t)$. Assim, $L(t)$ satisfaz a Equação 3 onde $\pi_N(0)$ é o vetor $\pi(0)$ restrito aos estados do conjunto N . Q_N é a matriz geradora infinitesimal restrita aos estados não absorventes [Pinheiro et al. 2018]. Finalmente, o MTT pode ser descrito como a Equação 4 [Bolch et al. 2006].

$$\frac{d\pi(t)}{dt} = \pi(t)Q \quad (1)$$

$$L_i(t) = \int_0^t \pi_i(x)dx \quad (2)$$

$$L_N(\infty)Q_N = -\pi_N(0) \quad (3)$$

$$MTT = \sum_{i \in N} L_i(\infty) \quad (4)$$

A Tabela 3 mostra os parâmetros de entrada utilizados para alimentar o modelo advindos de um experimento real realizado como validação e descrito na próxima seção. Tais valores referem-se a médias de 50 execuções de uma migração.

Tabela 3. Parâmetros de configuração de tempo usados no estudo de caso.

POLÍTICA	Name	Time(s)
Cold	Dump1	0,84352
	Copy1	4,55536
	Rest1	0,95682
PreCopy	PDump2	0,77930
	Dump2	0,75376
	Copy2	9,35288
PostCopy	Rest2	0,90658
	Dump3	4,98894
	Copy3	0,54768
Hybrid	Conect3	3,70338
	Rest3	0,63900
	PDump4	0,69168
Hybrid	Dump4	5,48088
	Copy4	4,06736
	Conect4	4,11074
	Rest4	0,71172

3.1. Validação

Para utilizar um modelo analítico (e.g., PN, CTMC, redes de filas, etc.) é essencial validar tal modelo para verificar sua corretude. A validação se trata de comparar os resultados gerados pelo modelo com os resultados advindos de uma simulação ou mensuração real [Jain 1990]. Esta subseção apresenta a validação do modelo absorvente proposto.

Começamos por descrever o protótipo sintético e depois apresentamos a metodologia e os resultados.

Para validar o modelo SPN aborvente proposto, desenvolvemos um protótipo e conduzimos experimentos em um cenário real. O objetivo foi comparar o Tempo Total de Migração (MTT) calculado pelo modelo com o MTT observado nos experimentos. Durante os experimentos, testamos as quatro políticas de migração: Cold, PreCopy, Post-Copy e Hybrid [Puliafito et al. 2019]. Os dados coletados foram usados para a análise deste experimento.

Nosso ambiente de testes é composto por duas máquinas virtuais, ambas hospedadas no mesmo Hypervisor. Cada máquina virtual possui 5000 MB de RAM e 2 núcleos de processamento, e opera com Ubuntu 22.04 e Kernel Linux. Elas utilizam o CRIU 3.17.1 para as funcionalidades de *checkpoint* e *restore*, o OpenSSL 3.0.2 para transferência de arquivos e o runC 1.1.9 como *runtime* de contêiner. A aplicação em uso é projetada para ser intensiva em termos de consumo de memória RAM e processamento. Utilizamos uma aplicação em Python que calcula uma sequência de Fibonacci e soma cada número ao seu antecessor na sequência. Para garantir um alto uso da memória RAM, a aplicação gera uma imagem correspondente a cada número calculado na sequência de Fibonacci, armazenada diretamente na memória RAM, em vez de ser salva em um disco rígido. Isso resulta em um constante preenchimento da memória RAM com novas imagens. A adoção da sequência de Fibonacci foi inspirada pelo trabalho de [Dayo 2021] que fez algo semelhante com contêineres Docker.

Para registrar os tempos de execução, instrumentamos o início e o fim de cada etapa de migração. Foram utilizadas as instruções da documentação oficial do CRIU para garantir a execução adequada das políticas de migração exploradas. Os tempos de serviço obtidos com este experimento foram utilizados para alimentar as transições estendidas do modelo SPN. A partir desses valores, todas as transições foram preenchidas com os tempos de suas respectivas etapas. Para tornar o resultado mais confiável, repetimos a execução do experimento 50 vezes para cada política de migração.

A Tabela 4 apresenta o resultado da validação. O modelo SPN pode ser solucionado de duas formas, por análise estacionária ou simulação estacionária. O modelo gerou os resultados por meio de simulação, condicionados a uma margem de erro de 2%. Aplicamos o teste T de duas amostras para comparar o MTT gerado pelo modelo com o MTT obtido nos experimentos. Todas as amostras apresentaram distribuição normal. Para verificar a significância do Teste T, observamos o valor p. O valor p é superior a 0,05 em todos os casos. Portanto, não podemos refutar a hipótese nula de igualdade em todos os casos, com 95% de confiança. Os resultados gerados pelo modelo são estatisticamente equivalentes ao experimento, logo o modelo reflete o ambiente real.

Tabela 4. Comparação dos resultados da mensuração com o modelo proposto.

Política	MTT(s)		Intervalo de Confiança		Desvio Padrão		P-Value
	Exp	Mod	Exp	Mod	Exp	Mod	
Cold	6,355	6,308	[6,405 - 6,305]	[6,019 - 6,598]	2,161	4,659	0,949
PreCopy	11,792	11,847	[11,742 - 11,842]	[11,269 - 12,425]	2,733	9,317	0,968
PostCopy	9,879	9,749	[9,829 - 9,929]	[9,369 - 10,128]	0,902	6,121	0,883
Hybrid	15,062	15,269	[9,829 - 9,929]	[14,743 - 15,795]	1,260	8,471	0,865

3.2. Estudo de Caso

Com o modelo agora validado podemos fazer estudos de caso para mostrar como o modelo pode ser útil. O estudo de caso serve tanto para obter novas descobertas sobre o funcionamento do modelo e consequentemente do sistema real. Nos estudos de caso, utilizamos a mesma quantidade de políticas de migração presentes no experimento de validação. Os parâmetros foram alimentados baseados em tempos colhidos no processo de validação, que estão presentes na Tabela 3. Os resultados deste primeiro estudo de caso são apresentados na Figura 2, o qual foca na métrica MTT.

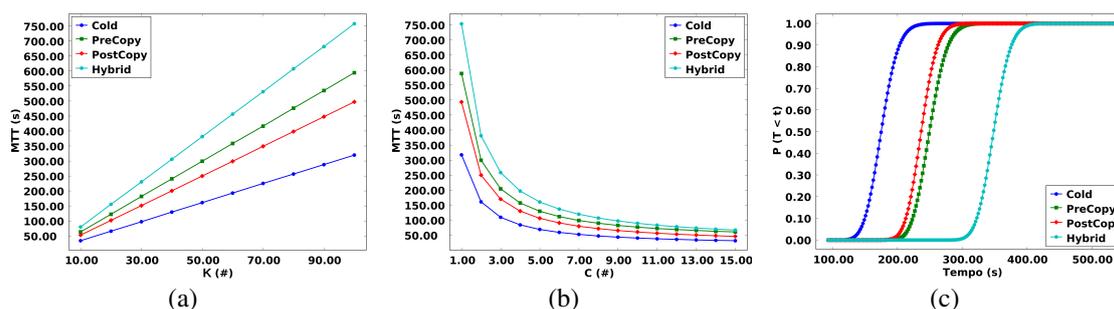


Figura 2. (a) - MTT em função do número de elementos a serem migrados, (b) - MTT em função da capacidade de migrações paralelas, (c) - CDF considerando o tempo total de migração.

A Figura 2(a) apresenta o resultado para o MTT em função da variação da quantidade de elementos a serem migrados. Foram testadas as seguintes quantidades de elementos migráveis: $K = [10, 20, 30, \dots, 100]$. O tempo de migração aumenta proporcionalmente com quantidade de elementos migrados, por haver um limite paralelo de migrações igual a 2. A política Hybrid teve o maior de todos MTT, chegando a ≈ 750 s. A política Cold teve um MTT menor em relação às outras políticas apresentadas. Comparando-se as quatro políticas, vemos que com menos elementos a serem migrados, os MTTs são semelhantes, e à medida que a quantidade aumenta, a política Hybrid que possui o, pior desempenho. Por exemplo, comparando-se o intervalo de $K=[50-100]$, temos que o Cold possui um aumento de $MTT=158$ s enquanto o Hybrid possui um aumento de $MTT=247$ s. Portanto, para mais elementos a serem migrados a escolha da política se torna cada vez mais criteriosa, pois isso impacta expressivamente no valor final de MTT.

A Figura 2(b) apresenta o MTT em função do aumento da capacidade de migrações paralelas (C). O C foi variado com os seguintes valores: $[1, 2, 3, \dots, 15]$. A simulação foi feita considerando 50 elementos a serem migrados ($K=50$). Durante o começo da análise, as quatro políticas de migração começam com um tempo consideravelmente alto, onde a Hybrid teve o pior desempenho. Desde o início da variação de C até o final, o tempo MTT segue diminuindo até se estabilizar. A estabilização acontece nas quatro políticas para $C \geq 13$. Isso implica haver uma baixa alteração a partir desse ponto, pois o MTT tem uma variação muito baixa em relação a um $C \geq 13$. Apesar de as linhas se aproximarem com o aumento do C, observa-se que elas não se tocam (pelo menos não até este valor de $C=15$). Portanto, o impacto da variação de C foi maior do que o impacto da variação de K. A principal contribuição deste estudo de caso é mostrar que o avaliador pode ter uma grande economia de recursos conhecendo o limiar de C que fará com que não melhore mais o desempenho mesmo o aumentando.

A Figura 2(c) apresenta a CDF, que permite determinar a probabilidade de toda a migração finalizar em determinada janela de tempo. Essa janela de tempo refere-se ao MTT. A simulação (do tipo transiente) foi feita considerando 50 elementos a serem migrados ($K=50$) e com capacidade paralela de migração igual a 2 ($C=2$). A janela total de MTT variou entre 0 e 459s. No geral, a probabilidade de finalização da migração é maior para Cold, que é maior do que PostCopy, que é maior do que a PreCopy, por fim, maior que a Hybrid. A partir daqui, consideramos a notação $P[\text{política}]$ para denotar a probabilidade de uma política específica. $P[\text{Cold}] = P[\text{PreCopy}] = P[\text{PostCopy}] = P[\text{Hybrid}] = 0$ para os MTTs iguais a 130s, respectivamente. Um ponto interessante é analisar o $\text{MTT} = 250\text{s}$, onde temos que $P[\text{Cold}] = 100\%$, $P[\text{PreCopy}] = 52\%$, $P[\text{PostCopy}] = 76\%$ e $P[\text{Hybrid}] = 0\%$. Consideremos, para este exemplo, que $\text{MTT} = 250\text{ms}$ é um requisito importante de acordo com o Nível de Serviço a ser atendido. Se o analista não quiser assumir riscos, obviamente irá adotar a política Cold, porém, se este requisito não for tão crítico, ele pode usar o PostCopy assumindo o risco que há apenas 76% de chance de a migração terminar nesse requerido tempo. Finalmente, vale ressaltar que, no tempo $\text{MTT}=459\text{s}$, a probabilidade de término da migração é de 100% para todas as políticas. Novamente, se o requisito demanda que se tenha um MTT igual a 459s, qualquer uma das quatro políticas irá atendê-lo.

4. Modelo SPN Não Absorvente

A Figura 3 representa o modelo SPN sem estado absorvente que permite calcular mais métricas de interesse. Um modelo sem estado absorvente é aquele que recebe novas requisições de forma constante e de forma ininterrupta obedecendo um tempo entre chegadas AD. O Nó de origem é representado por uma transição de cor cinza de distribuição determinística, ou seja, a cada intervalo de tempo (arrival delay - AD) é iniciada a migração de um novo contêiner. O comportamento do modelo segue o mesmo fluxo do modelo com estado absorvente. Um ponto que deve ser observado e que inclusive difere do modelo absorvente é, que nos lugares P2, P6, P8 e P14, o token pode seguir por dois caminhos, sendo que no qual se o token for pelas transições imediatas TI1, TI2, TI3 ou TI4, isso indica que o processo de migração do contêiner falhou ao ser copiado do Nó_A para o Nó_B. Porém, se o token seguir por TI5, TI6, TI7 ou TI8, isso indica que o processo de migração prosseguiu sem falha na etapa de cópia.

O tempo médio de resposta (*MRT*) pode ser obtido a partir da Lei de Little [Jain 1990]. Esta lei requer um sistema estável, ou seja, que possua uma taxa de requisições menor que a taxa de processamento dos servidores. A lei indica que o MRT é dado pela multiplicação da quantidade de requisições dentro do sistema pelo tempo entre chegadas (AD). No modelo, o tempo entre chegadas reside na transição de nome AD, localizada na extremidade esquerda do modelo. Neste artigo, as requisições são os contêineres e chamamos o MRT de *Mean Migration Time* (MMT). Assim, neste artigo, o número de elementos no sistema é a soma de tokens em todos os lugares por onde passa o contêiner. Por ter um único lugar de capacidade (Cap), para nosso modelo, esta soma pode ser facilmente obtida por $C - \text{Esp}\{Cap\}$, onde C é a quantidade total máxima de processos agentes de migração e $\text{Esp}\{Cap\}$ representa o número de tokens no lugar Cap naquele momento. $\text{Esp}\{Nomedolocal\}$ representa a esperança estatística de existir tokens em “nomedolocal”, onde $\text{Esp}\{Nomedolocal\} = (\sum_{i=1}^n P(m(Local) = i) \times i)$, sendo n o maior número de tokens que o *Local* pode conter. Em outras palavras,

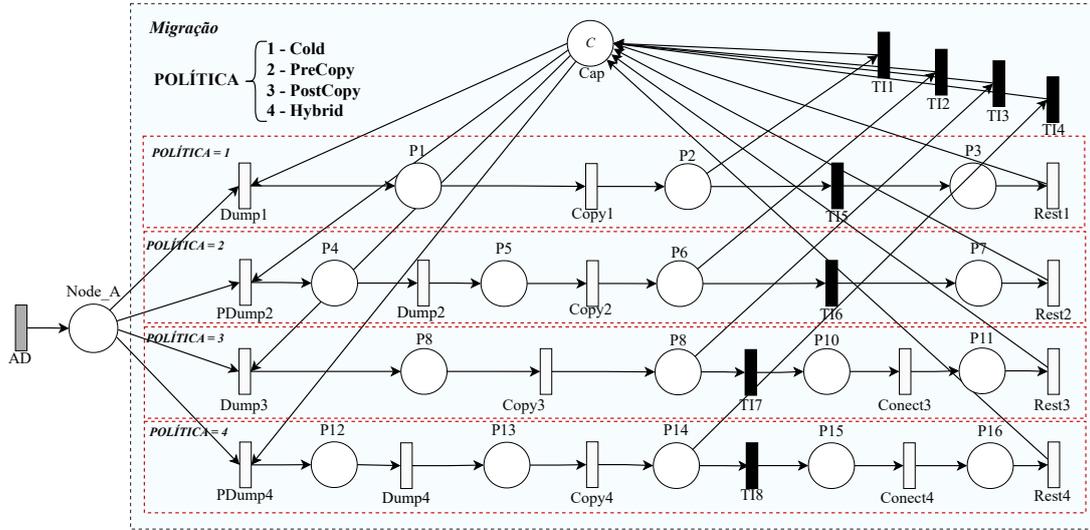


Figura 3. Modelo não absorvente

$Esp\{Nomedolocal\}$ indica o valor esperado de tokens naquele Local em determinado momento ou estacionariamente. Portanto, a equação correspondente à Lei de Little para MMT utilizada no nosso modelo é expressa na Equação 5.

$$MMT = (C - Esp\{Cap\}) \times AD \quad (5)$$

A Equação 6 define a probabilidade de haver perdas de requisições (DP_PROB). Para calcular o descarte, é necessário não restar mais nenhuma capacidade de enfileiramento na entrada do sistema. $P(Local = n)$ calcula a probabilidade de existirem n tokens em “Local”. Portanto, DP_PROB considera não ter mais capacidade disponível ($\#Cap = 0$) e ter tokens aptos a entrar no processo de migração ($\#Node_A > 0$).

$$DP_PROB = P\{(\#Cap = 0) \text{ AND } (\#Node_A > 0)\} \quad (6)$$

A utilização é a divisão do número esperado de tokens em um local (por onde passam os tokens executados) pela respectiva capacidade total. A utilização média do processamento da etapa de migração é dada pela Equação 7, lembrando-se que $C - Esp\{Cap\}$ retorna para este modelo o número de elementos dentro do sistema.

$$U = \frac{C - Esp\{Cap\}}{C} \quad (7)$$

A taxa de migração (MR) é dada pela soma das vazões de cada política. Assim, quando uma política for habilitada, a vazão das demais é nula. O cálculo da vazão de um par lugar+transição é dado pela divisão do número de tokens naquele lugar ($Esp\{lugar\}$) pelo tempo daquela transição ($T\{transicao\}$). Assim MR é dada pela equação 8.

$$MR = \frac{Esp\{P3\}}{T\{Rest1\}} + \frac{Esp\{P7\}}{T\{Rest2\}} + \frac{Esp\{P11\}}{T\{Rest3\}} + \frac{Esp\{P16\}}{T\{Rest6\}} \quad (8)$$

4.1. Estudo de Caso

A Figura 4 mostra o resultado da variação da taxa de chegada para calcular O MMT, o DP e U com o modelo não absorvente. A Figura 4(a) apresenta o resultado do Tempo Médio de Migração (MMT) em função da variação da Taxa de Chegada (AR). Foram testados os seguintes valores: $AR = [0,005, 0,043, 0,082, \dots, 0,350]$ e capacidade paralela de migração igual a 2 ($C=2$). Para a condução desse caso de uso, a Probabilidade de Falha está igual a 0,0001.

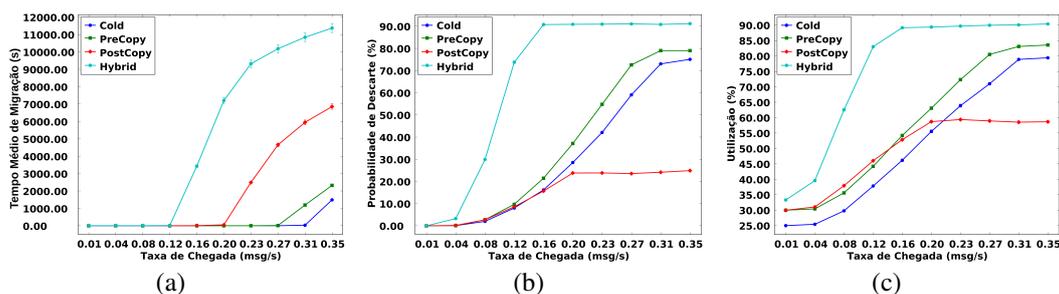


Figura 4. (a) - Tempo Médio de Migração, (b) - Probabilidade de Descarte, (c) - Utilização do Sistema.

Durante o começo do experimento, onde o $AR = 0,005$, todas as políticas possuem um valor de MMT próximo, para política Cold e PreCopy $\approx 6,9s$, seguido por PostCopy com $\approx 9,8s$ e Hybrid com $\approx 15,0s$. O MMT segue com esses valores até $AR = 0,158$. Nesse ponto, a política Hybrida tem um aumento significativo do MMT chegando a $\approx 3434,0s$. As outras políticas seguem sem grandes alterações no MMT até $AR = 0,235$, quando a política PostCopy chega a um MMT de $\approx 2496,0s$. No final do experimento, quando o $AR = 0,35$, a política Cold possui um MMT menor em relação às demais, chegando a $\approx 1496,0s$, comparado com a política Hybrida a que teve um MMT maior chegando a $\approx 11379,0s$ uma diferença de $9883,0s$. Portanto, caso o AR seja inferior a $AR = 0,158$ a escolha da política não é relevante em relação ao MMT, porém quando, o AR é superior a $0,35$, vale considerar a escolha da política Cold para o processo de migração.

A Figura 4(b) apresenta o resultado da Probabilidade de Descarte (PD) em função da variação da Taxa de Chegada (AR). Durante o início do experimento, onde $AR = 0,005$, todas as políticas começam com um valor de PD próximo a zero. O PD permanece próximo a esses valores até $AR = 0,081$, quando a política Hybrid tem um aumento significativo da probabilidade de descarte, chegando a aproximadamente $29,9\%$. As outras políticas seguem sem grandes alterações do PD até $AR = 0,158$, quando todas as políticas têm um aumento significativo da probabilidade de descarte, chegando a $\approx 18\%$. Nesse ponto, a política Hybrid já está com 90% de PD. Um ponto interessante é que quando o AR é igual e superior a $0,196$, a política PostCopy tem a probabilidade de descarte estabilizada em $\approx 23\%$, seguindo até o final do experimento quando o $AR = 0,35$. No final do experimento, a política PostCopy possui um PD menor em relação às demais políticas, chegando a 24% , comparado com a política Hybrid que teve o maior probabilidade de descarte, chegando a 91% , uma diferença de 64% em relação à política PostCopy. Portanto, caso o AR seja inferior a $AR = 0,005$, a escolha da política não é relevante, pois todas as políticas têm a probabilidade de descarte igual a zero. No entanto, quando o AR é superior a $0,120$, vale a pena considerar a escolha das demais políticas. Caso o AR seja

igual a 0,350, a política de migração com a menor probabilidade de descarte é a PostCopy.

A Figura 4(c) apresenta o resultado da utilização do sistema em função do AR. No início do experimento, onde $AR = 0,005$, todas as políticas começam com a utilização superior a 25%. A utilização da política Hybrid mostra um aumento rápido na utilização e atinge rapidamente 90% em $AR = 0,12$. As políticas Cold e PreCopy tiveram um aumento mais gradual na utilização durante o experimento. A PostCopy mostra uma estabilização em torno de $\approx 59\%$ de utilização para AR superior a 0,196. No final do experimento, onde o $AR = 0,35$, a política PostCopy tem a menor utilização, chegando a 58,7%, enquanto a política Hybrid tem a maior utilização, chegando a 90,4%. Portanto, caso a Taxa de Chegada seja igual a 0,158, o avaliador pode considerar a escolha das políticas Cold, PreCopy e PostCopy, pois a utilização das três políticas é $\approx 50\%$. Caso o avaliador considere um $AR = 0,35$, a política PostCopy demonstrou uma utilização de 58% dos recursos, 32% a menos que a política Hybrid.

A Figura 5 apresenta o resultado da Taxa de Migração (MR) em função da variação da probabilidade de falha durante o processo de cópia, incluindo cenários extremos com 0.9 de probabilidade de falha. Foram testados os seguintes valores: Probabilidade = [0.1, 0.2, 0.3, ..., 0.9] e capacidade paralela de migração igual a 2 ($C=2$).

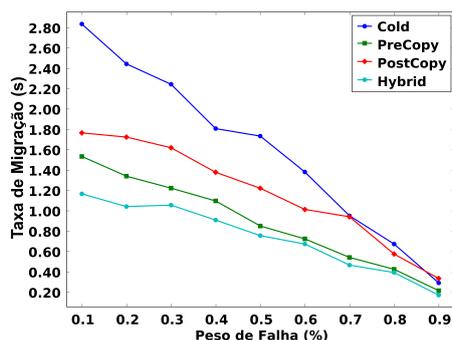


Figura 5. Taxa de Migração.

Durante o começo do experimento, onde a Probabilidade = 0,1, a política Cold possui a maior taxa de migração, chegando a $\approx 2,80s$, seguido por PostCopy com $\approx 1,80s$, PreCopy com $\approx 1,60s$ e Hybrid com $\approx 1,20s$. A taxa diminui conforme o Probabilidade de Falha aumenta. Podemos observar, quando a Probabilidade de Falha chega a 0,5 que as políticas Hybrid e PreCopy estão com a Taxa de Migração bem próximas, tendo apenas uma diferença de 0,10s. Quando a Probabilidade de Falha chega a 0,7, a Taxa de migração chega a ser a mesma para as políticas de migração Cold e PostCopy, com uma Taxa de 0,95s. Por fim, quando a Probabilidade de Falha chega a 0,9, a Taxa de Migração das quatro políticas estão bem próximas, chegando a $\approx 0,20s$. Portanto, caso a Probabilidade de Falha seja baixa, a escolha da política influencia diretamente na Taxa de Migração, porém, se a Probabilidade de Falha for alto, a escolha da política não se torna um fator relevante, pois a Taxa de Migração será semelhante nas quatro políticas do estudo de caso.

5. Conclusão

Este artigo apresentou um modelo SPN para comparar políticas de migração de contêiner. O modelo calcula as métricas MTT, MMT, MR, PD e U. O modelo explora

a variação da quantidade de elementos a serem migrados, capacidade de migração paralela do sistema, probabilidade de falha e taxa de chegada. O modelo também permite observar o CDF. O modelo proposto é essencial para analistas de sistemas preverem o tempo de migração em vários cenários, especialmente quando a infraestrutura para testes reais não está disponível. Neste trabalho, afirmamos que a política de Cold tem o melhor desempenho considerando o tempo total de migração e tempo médio de migração. Para a questão de o sistema possuir uma taxa de falha alta, a escolha da política de migração não é relevante, pois a taxa de migração é semelhante para as políticas analisadas. Trabalhos futuros incluem calcular gasto energético demandado na migração, topologia da rede, bem como o custo monetário da infraestrutura a ser utilizada no processo.

Referências

- Bause, F. and Kritzinger, P. S. (2002). *Stochastic petri nets*, volume 1. Vieweg Wiesbaden.
- Benjaponpitak, T., Karakate, M., and Sripanidkulchai, K. (2020). Enabling live migration of containerized applications across clouds. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2529–2538. IEEE.
- Bolch, G., Greiner, S., De Meer, H., and Trivedi, K. S. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.
- Chou, C. C., Chen, Y., Milojevic, D., Reddy, N., and Gratz, P. (2019). Optimizing post-copy live migration with system-level checkpoint using fabric-attached memory. In *2019 IEEE/ACM Workshop on Memory Centric High Performance Computing (MCHPC)*, pages 16–24. IEEE.
- Dayo, A. O. (2021). A multi-containerized application using docker containers and kubernetes clusters. *Int J Comput Appl*, 183(44):55–60.
- Di, Z., Shao, E., and Tan, G. (2021). High-performance migration tool for live container in a workflow. *International Journal of Parallel Programming*, 49:658–670.
- Fan, W., Han, Z., Li, P., Zhou, J., Fan, J., and Wang, R. (2019). A live migration algorithm for containers based on resource locality. *Journal of Signal Processing Systems*, 91:1077–1089.
- Govindaraj, K. and Artemenko, A. (2018). Container live migration for latency critical industrial applications on edge computing. In *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, volume 1, pages 83–90. IEEE.
- Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.
- Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*, volume 1. Wiley New York.
- Joy, A. M. (2015). Performance comparison between linux containers and virtual machines. In *2015 international conference on advances in computer engineering and applications*, pages 342–346. IEEE.

- Karhula, P., Janak, J., and Schulzrinne, H. (2019). Checkpointing and migration of iot edge functions. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, pages 60–65.
- Kaur, A., Kumar, S., Gupta, D., Hamid, Y., Hamdi, M., Ksibi, A., Elmannai, H., and Saini, S. (2023). Algorithmic approach to virtual machine migration in cloud computing with updated sesa algorithm. *Sensors*, 23(13):6117.
- Machen, A., Wang, S., Leung, K. K., Ko, B. J., and Salonidis, T. (2017). Live service migration in mobile edge clouds. *IEEE Wireless Communications*, 25(1):140–147.
- Maciel, P., Matos, R., Silva, B., Figueiredo, J., Oliveira, D., Fé, I., Maciel, R., and Dantas, J. (2017). Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In *2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC)*, pages 50–57. IEEE.
- Maciel, P. R., Lins, R. D., and Cunha, P. R. (1996). *Introdução às redes de Petri e aplicações*. UNICAMP-Instituto de Computacao Sao Paulo, Brazil.
- Maheshwari, S., Choudhury, S., Seskar, I., and Raychaudhuri, D. (2018). Traffic-aware dynamic container migration for real-time support in mobile edge clouds. In *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE.
- Marsan, M. A., Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. (1998). Modeling with generalized stochastic petri nets. *ACM SIGMETRICS performance evaluation review*, 26(2):2.
- Nelson, R. (2013). *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. Springer Science & Business Media.
- Pecholt, J., Huber, M., and Wessel, S. (2021). Live migration of operating system containers in encrypted virtual machines. In *Proceedings of the 2021 on Cloud Computing Security Workshop*, pages 125–137.
- Pinheiro, T., Silva, F. A., Fe, I., Kosta, S., and Maciel, P. (2018). Performance and data traffic analysis of mobile cloud environments. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4100–4105. IEEE.
- Puliafito, C., Vallati, C., Mingozzi, E., Merlino, G., Longo, F., and Puliafito, A. (2019). Container migration in the fog: A performance evaluation. *Sensors*, 19(7):1488.
- Ramanathan, S., Kondepu, K., Razo, M., Tacca, M., Valcarengi, L., and Fumagalli, A. (2021). Live migration of virtual machine and container based mobile core network components: A comprehensive study. *IEEE Access*, 9:105082–105100.
- Silvaa, B., Maciela, P. R. M., Zimmermannb, A., and Brillhantea, J. (2014). Survivability evaluation of disaster tolerant cloud computing systems. In *Proc. Probabilistic Safety Assessment & Management conference*, page 12.
- Torre, R., Urbano, E., Salah, H., Nguyen, G. T., and Fitzek, F. H. (2019). Towards a better understanding of live migration performance with docker containers. In *European Wireless 2019; 25th European Wireless Conference*, pages 1–6. VDE.