

Avaliação dos Serviços de Armazenamento da *Amazon Web Services* para Gravação e Recuperação de *Checkpoints*

Luan Teylo¹, Rafaela Brum¹, Luciana Arantes², Pierre Sens², Lúcia Drummond¹

¹Instituto de Computação – Universidade Federal Fluminense (UFF)

²Laboratoire d’Informatique de Paris 6 – Sorbonne Université

{luanteylo, rafaelabrum}@id.uff.br

{luciana.arantes, pierre.sens}@lip6.fr, lucia@ic.uff.br

Abstract. *Cloud providers offer several resources for processing and storing data. Some of those resources are prone to failures or revocations, and the adoption of fault tolerance techniques are necessary to minimize the impact and cost of those failures on the users’ application and budget. One of the most widely adopted solutions is the checkpoint and recovery techniques, that record the current state of the application and use it to restart the application from its last state if any fault happens. Thus, those approaches require a safe storage location. Fortunately, when adopted cloud environments, the user has different storage options given by the provider himself in the form of cloud services. In this work, we want to evaluate three of those services: Amazon Simple Storage Service (S3), Amazon Elastic Block Store (EBS) and Amazon Elastic File System (EFS). The objective is to characterize and evaluate the performance of those services in relation to the checkpoint and recovery process in the context of faults induced by the revocation of virtual machines.*

Resumo. *Os provedores de nuvem oferecem vários serviços para processamento e armazenamento de dados. Alguns desses recursos são propensos a falhas ou revogações. Neste caso, a adoção de técnicas de tolerância a falhas é essencial para minimizar o impacto e o custo que estas falhas podem exercer na execução das aplicações e no orçamento dos usuários. Uma das soluções amplamente adotadas são as técnicas de checkpoint and recovery (CR), que gravam o estado atual da aplicação para que, caso ocorra falhas, esta seja reiniciada a partir do último estado gravado. Assim, essas abordagens exigem um local de armazenamento seguro e disponível ao longo de toda a execução. Felizmente, os serviços de nuvens também podem ser utilizados para este fim, uma vez que diferentes opções de armazenamento também são oferecidos. Neste trabalho, queremos avaliar três desses serviços: o Amazon Simple Storage Service (S3), o Amazon Elastic Block Store (EBS) e o Amazon Elastic File System (EFS). O objetivo é caracterizar e avaliar o desempenho desses serviços em relação a gravação e recuperação de checkpoints no contexto de falhas induzidas pela revogação de máquinas virtuais.*

1. Introdução

As plataformas de computação em nuvem se consolidaram como um meio economicamente acessível de aquisição de recursos computacionais. Hoje, esses ambientes oferecem serviços que englobam desde processamento e armazenamento, até inteligência

computacional. Uma das principais vantagens do uso de nuvem é a alocação e liberação quase instantânea de recursos, através do modelo *pay-per-use* em que o usuário é cobrado de acordo com o tempo de uso dos recursos contratados.

Uma das principais formas de contratação de recursos nos ambientes de nuvens é a aquisição direta de máquinas virtuais (MVs), pelo modelo de Infraestrutura como um Serviço (IaaS, do inglês *Infrastructure as a Service*). No modelo IaaS, o usuário tem controle total sobre as etapas de instalação, configuração e execução das MVs. Além disso, as MVs são oferecidas em diferentes modelos de contratação, nos quais a disponibilidade e as garantias de Qualidade de serviço (QoS, do inglês *Quality of Service*) variam de acordo com os preços cobrados. No Amazon EC2 (Elastic Cloud Compute), por exemplo, os usuários podem optar pelos seguintes modelos de contratação (também chamados de mercado): i) *on-demand*, no qual as MVs são oferecidas a preços fixos e com uma garantia alta de disponibilidade e resiliência; ii) *reserved*, no qual MVs com alta disponibilidade e resiliência são oferecidas com até 75% de desconto (em relação ao mercado *on-demand*), porém com exigência de contratação de no mínimo um ano; e iii) *spot*, no qual os recursos ociosos do provedor são oferecidos com até 90% de desconto em comparação com o mercado *on-demand*, sem compromisso de longo prazo e sem garantias de QoS [AWS 2020a]. Ao optar por MVs *spots*, os usuários têm acesso ao mesmo poder computacional oferecido no mercado *on-demand*, mas por uma fração do seu preço. No entanto, as MVs do mercado *spot* são sujeitas a revogação do provedor, i.e., o provedor pode terminar a MV a qualquer momento caso necessite dos recursos. Como a decisão de revogação é feita exclusivamente pelo provedor, neste cenário, a adoção de técnicas de tolerância a falhas por parte dos usuários é essencial.

Diversas abordagens de tolerância a falhas foram propostas na literatura para lidar com as revogações do mercado *spots* [Teylo et al. 2019, Yi et al. 2010, Poola et al. 2014]. Uma das principais técnicas adotada é o *checkpoint and recovery* (CR). As abordagens de CR permitem que o estado atual da aplicação seja preservado em um dispositivo de armazenamento, e recuperado caso ocorram revogações ou falhas [Hargrove and Duell 2006]. Portanto, ao usar essa abordagem com MVs *spots*, é essencial garantir que todos os arquivos necessários para a recuperação das aplicações estejam sempre disponíveis e que estes não sejam afetados pelas revogações. No caso da nuvem, várias opções de armazenamento são oferecidas, e podem ser utilizadas para essa finalidade. O Amazon Web Service (AWS), por exemplo, oferece diversos destes serviços [Services 2020c]. Neste trabalho, estamos interessados em opções de armazenamento de uso geral que podem ser usadas para armazenar e recuperar arquivos durante a execução de aplicações em MVs *spot*. Assim, três serviços da AWS foram considerados: o *Amazon Simple Storage Service* (S3), um serviço de armazenamento de objetos que oferece escalabilidade, segurança e desempenho [AWS 2020c]; o *Amazon Elastic Block Store* (EBS), um serviço de armazenamento em bloco projetado para MVs do EC2 cuja carga de trabalho seja alta e constante [AWS 2020b]; e o *Amazon Elastic File System* (EFS), um sistema de arquivos NFS elástico, simples e escalável [Services 2020a].

Neste trabalho são apresentados os esforços iniciais para avaliar e caracterizar cada um destes serviços em relação à gravação e recuperação de *checkpoints*. O objetivo principal é identificar o desempenho e a viabilidade monetária da utilização dessas opções de armazenamento aplicados à abordagens de CR. O restante deste trabalho tem a

seguinte organização. Na seção 3 são discutidas as características e os custos dos serviços avaliados. A seção 4 apresenta a abordagem de CR usada nas avaliações. Os resultados da avaliação são discutidos na Seção 5 e a Seção 6 conclui o artigo e apresenta algumas direções futuras.

2. Trabalhos Relacionados

Diversas técnicas e protocolos de CR são apresentadas na literatura [Elnozahy et al. 2002, Egwuotuoha et al. 2013]. Elnozahy *et al.* [Elnozahy et al. 2002] apresenta um estudo que classifica diversos destes protocolos. Neste trabalho estamos interessados na avaliação de desempenho da gravação e recuperação de *checkpoints* no contexto dos serviços de armazenamentos oferecidos pela AWS. Porém, a maioria dos trabalhos relacionados não apresentam resultados práticos e avaliações destes serviços [Nicolae and Cappello 2011, Zhou et al. 2017, Meroufel and Belalem 2018, Yaothanee and Chanchio 2019]. Além disso, esses trabalhos se concentram na proposta de abordagens de armazenamento de *checkpoints* e não na avaliação dos serviços. Em [Nicolae and Cappello 2011], por exemplo, os autores propõem um repositório centralizado para o armazenamento dos *checkpoints*. Este repositório seria construído a partir das partes não alocadas dos nós de computação da nuvem. Embora direcionado para serviços de nuvens, o foco principal do trabalho são estratégias do lado do provedor e não do usuário, como é o caso deste trabalho.

Em [Zhou et al. 2017], os autores assumem uma topologia *fat-node* de 3 camadas para o provedor e propõem o armazenamento de *checkpoints* em servidores vizinhos. Assim como em [Nicolae and Cappello 2011], os autores assumem um nível de controle disponível apenas para o provedor. Já em [Meroufel and Belalem 2018], uma solução de armazenamento acessível ao usuário é proposta. Os autores apresentam uma estratégia para *workflows*, no qual os *checkpoints* são replicados em outras MVs. No entanto, não são apresentadas avaliações em relação aos tempos de gravação e recuperação da estratégia. Além disso, como as MVs utilizadas como replicas estão sujeitas a falhas, o mecanismo de replicação precisa distribuir as cópias de modo a minimizar a probabilidade de falhas. Essa preocupação não existe na adoção de serviços de armazenamento, uma vez que a segurança dos dados armazenados depende do provedor e não do usuário.

Yaothanee e Chanchio [Yaothanee and Chanchio 2019] propõem que os *checkpoints* sejam salvos primeiro na memória principal para depois serem escalonados para serem transferidos assincronamente para uma memória permanente do tipo SSD ou HDD. Os autores demonstram um ganho considerável de desempenho na abordagem proposta. Porém, os testes foram realizados considerando um ambiente de nuvem privada e abordagens de armazenamento compartilhado não foram avaliadas. Em [Di et al. 2013] os autores caracterizam o tempo de gravação e recuperação dos checkpoints em duas configurações de armazenamento: i) disco local, e ii) disco compartilhado NFS (*Network File System*). Porém, assim como [Yaothanee and Chanchio 2019], os testes são realizados em uma nuvem privada e os serviços de armazenamento oferecidos por um provedor público não são considerados.

3. Serviços de Armazenamento Avaliados

A AWS oferece diversos serviços de armazenamento [Services 2020c]. Cada serviço é otimizado para diferentes necessidades e demandas de armazenamento. Neste trabalho, três dos serviços mais populares de propósito geral são avaliados: (i) o *Amazon Simple Storage Service* (S3), (ii) o *Amazon Elastic Block Store* (EBS), e (iii) o *Amazon Elastic File System* (EFS). A Tabela 1 apresenta os valores monetários de cada um destes serviços.

Tabela 1. Comparação de preços dos serviços de armazenamento avaliados

Serviço	Preço por GB/Mês (US\$)
S3	0,023
EFS	0,100
EBS	0,025

3.1. Amazon Simple Storage Service

O *Amazon Simple Storage Service* (S3) é provavelmente a opção de armazenamento mais conhecida e utilizada da AWS. De acordo com a AWS, esse serviço pode ser utilizado para armazenar e resgatar qualquer quantidade de dados e foi desenvolvido com um conjunto mínimo de características focadas na simplicidade e robustez do sistemas [AWS 2020c]. O serviço provê armazenamento para objetos de uma ampla gama de tamanhos (de 0 Bytes a 5TB), que são armazenados seguindo uma organização de dois níveis. No nível superior há os chamados *buckets*, que são estruturas similares a pastas e que possuem um nome global único. Cada usuário pode criar até 100 *buckets* associados a uma única conta, sendo que um número ilimitado de objetos podem ser armazenados em um *bucket*. Já no nível inferior, encontram-se os objetos, que são os dados e metadados armazenados pelo usuário.

Dentro de um *bucket*, o usuário pode criar e ler os objetos. No entanto, a alteração de um objeto exige que este seja transferido completamente para um sistema de arquivos local, atualizado e reenviado para o S3. Além disso, o S3 não oferece suporte nativo para sistemas de arquivos, sendo necessária a utilização de ferramentas de terceiros para esta finalidade. Neste trabalho, foi utilizada a ferramenta S3FS-FUSE [Rizun 2011], que faz uma interface com o S3 montando um sistema de arquivos no espaço de usuário compatível com o padrão POSIX [Rizun 2011]. Os custos monetários do S3 variam de acordo com cada região da AWS, e é determinada conforme o tempo de armazenamento dos dados e o tamanho dos objetos. Na região *us-east-1*, em abril de 2020, o preço mensal de cada giga byte de dados armazenados era de US\$0,023. Além disso, a cada 1.000 solicitações do tipo *PUT*, *COPY*, *POST* ou *LIST* são cobrados US\$0,005 adicionais [AWS 2020c].

3.2. Amazon Elastic Block Store

O *Amazon Elastic Block Store* (EBS) é a opção de armazenamento local do EC2. Neste serviço, os usuários podem criar volumes virtuais, que são então associados a um ponto de montagem no sistema de arquivo da MV [Ruiz-Alvarez and Humphrey 2011]. A capacidade dos volumes EBS pode variar de 1 a 16 TBs e é definida pelo usuário no momento da solicitação [AWS 2020b]. Além disso, os volumes são persistentes, independentes e

podem ser mantidos por tempo indefinido. No entanto, cada volume só pode ser associado e montado a uma MV por vez, limitando o acesso ao disco a essa única MV.

O EBS oferece dois tipos de volumes: os SSDs (Solid State Drives), que são indicados para aplicações sensíveis à latência; e os Hard Disk Drives (HDDs), indicados para aplicações que necessitem de altas taxas de escrita. Neste trabalho, foram avaliados os volumes SSDs, uma vez que estes discos apresentam latência inferior a 10 milissegundos e portanto apresentam maior potencial para redução de tempo de gravação dos *checkpoints* [AWS 2020b]. Além disso, as taxas de escrita dos volumes SSDs são de até 250 MB/s. Diferente do S3, no EBS o usuário é cobrado pelo tamanho total do disco e não pelos dados armazenados. Em abril de 2020, o preço mensal por GBs dos volumes da EBS na região *us-east-1* era de US\$0,10. Não há cobranças adicionais para as operações de leitura e gravação nos volumes EBS.

3.3. Amazon Elastic File System

O *Amazon Elastic File System* (EFS) fornece um sistema de arquivos simples e escalável, que pode ser utilizado nos ambientes de nuvens [Services 2020a]. O serviço aceita arquivos de até 52.66 TBs e é limitado para até 35,000 operações de leitura e escrita por segundo. O sistema de arquivos compatível com EFS é o NFS (Network File System) versão 4 (NFSv4.0 ou NFSv4.1). Sendo assim, os volumes EFS podem ser acessados concorrentemente por diversas MVs através de um cliente NFS.

O serviço apresenta duas classes de cobrança, determinada automaticamente conforme a frequência de acesso aos arquivos armazenados. Na região *us-east-1*, em abril de 2020 os custos mensais pelos arquivos com maior e menor frequência de acesso era de US\$0,30 e US\$0,025 mensais por GB, respectivamente.

4. Abordagem de CR Utilizado nas Avaliações

Para realizar as avaliações foi desenvolvido um controlador responsável por coordenar a execução da aplicação, requisitar os recursos e gerenciar os procedimentos de CR. Este controlador foi construído utilizando o boto3, um SDK disponibilizado pela Amazon que permite o gerenciamento dos serviços da AWS, como o EC2 e o S3, através de aplicações escritas em Python [Services 2020b]. A Figura 1 ilustra as etapas realizadas pelo controlador para executar uma tarefa no mercado *spot*. Como pode ser visto, além do controlador, estas etapas envolvem o usuário, que submete a tarefa e aguarda os resultados da execução, e o provedor de nuvem, que disponibiliza os recursos computacionais.

Primeiro, o controlador recebe a solicitação do usuário que é encaminhada na forma de uma tupla (a, it, S) , na qual a representa a tarefa a ser executada, it é o tipo¹ da MV utilizada e S determina qual o serviço de armazenamento será adotado. Na sequência, o controlador solicita uma MV no mercado *spot*, chamada de MV primária, e aguarda a resposta. Caso a requisição seja atendida pelo provedor, uma mensagem de confirmação é encaminhada para o controlador. Caso contrário, o provedor envia uma mensagem de negação e, neste caso, a execução é abortada.

¹O tipo da MV determina a quantidade de recursos da máquina, como a quantidade de CPUs e de memória. Os tipos de MVs oferecidos pela AWS podem ser consultados em <https://aws.amazon.com/ec2/instance-types/>

No caso representado na Figura 1, a solicitação é atendida pelo provedor e a MV é alocada. Em seguida, o controlador aguarda a inicialização completa da máquina e realiza uma conexão via protocolo SSH, que é utilizada para: i) enviar os arquivos de execução e entrada; ii) executar os comandos de configuração da MV; e iii) iniciar a execução da tarefa. Vale ressaltar que é na fase de configuração da MV que o sistema de arquivos, criado de acordo com o serviço S , é montado na máquina.

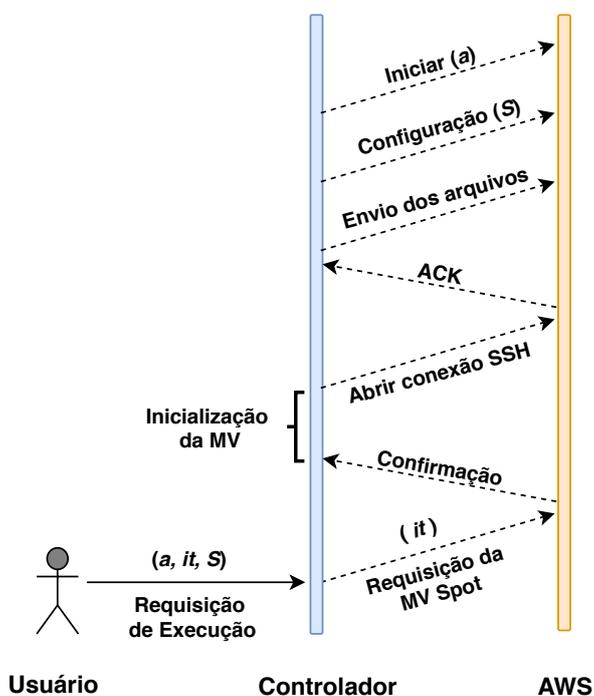


Figura 1. Etapas realizadas pelo controlador para executar uma tarefa em uma MV no mercado spot

Ao longo da execução o controlador monitora a MV primária e coordena a gravação dos *checkpoints*. Neste trabalho, a gravação dos *checkpoints* é feita utilizando a ferramenta *Checkpoint/Restore In Userspace* (CRIU) [EMELYANOV 2011]. O CRIU é uma ferramenta de código aberto amplamente utilizada que pode registrar o estado de aplicações individuais. Sendo assim, neste trabalho, apenas o estado das aplicações são gravadas durante o procedimento de *checkpoint* e não o estado da MV. Além disso, diferentemente de outras ferramentas, o CRIU é implementado no espaço de usuário e não precisa de privilégios especiais de superusuário. Esta ferramenta está disponível nas principais distribuições Linux.

Quando uma nova MV é iniciada, o controlador deve garantir que o sistema de arquivos escolhido seja devidamente instalado e esteja disponível durante toda a execução. Como dito anteriormente, a configuração do sistema de arquivos é realizada na fase de configuração da MV por meio do protocolo SSH e cada serviço avaliado demanda etapas distintas de configuração. No caso do S3, o usuário deve criar e disponibilizar previamente um *bucket*, que será utilizado ao longo da execução. Neste trabalho, a criação do *bucket* só é realizada uma única vez e todas as tarefas executadas utilizam o mesmo *bucket*. Após a criação do *bucket*, o controlador cria o sistema de arquivos compartilhados utilizando a ferramenta S3FS-FUSE.

No caso dos volumes EBS, sempre que uma MV primária é iniciada, o controlador cria um novo volume que é então associado à MV. Uma vez associado a uma MV, o volume é reconhecido como um dispositivo padrão de armazenamento e pode então ser montado pelo sistema operacional da máquina. Já no caso do EFS, o controlador executa a montagem do sistema através de um cliente NFS. Assim como no S3, o usuário deve criar previamente um volume EFS que será então utilizado em todas as execuções.

Sempre que há a ocorrência de revogações das MVs *spot*, o controlador inicia o processo de recuperação da aplicação. Neste caso, o controlador requisita uma nova MV, chamada MV *backup*, no mercado *spot*. Porém, caso o provedor não atenda à essa requisição, a MV é então solicitada no mercado *on-demand*. Uma vez que a MV *backup* esteja disponível e pronta para acesso, o controlador monta o sistema de arquivos (de acordo com o serviço *S*) e reinicia a aplicação a partir do último *checkpoint* gravado. Vale ressaltar que, assim como a MV primária, a MV *backup* também será do tipo *it*. Além disso, caso a MV seja do mercado *spot*, o controlador continua coordenando a gravação dos *checkpoints* e caso ocorra uma nova revogação, o procedimento de recuperação é novamente executado.

5. Avaliações e Resultados

5.1. Caracterização do tempo de gravação dos *Checkpoints*

Para caracterizar o tempo de gravação dos *checkpoints* nos serviços de armazenamento avaliados, foi utilizada uma MV do tipo *c3.2xlarge* que é composta por 8 CPUs virtuais e têm capacidade de memória de 16 GBs. Os testes foram executados utilizando uma aplicação sintética proposta em [Alves and Drummond 2017]. Essa aplicação não realiza trocas de mensagens, executando apenas uma série de operações locais sobre vetores. Além disso, a aplicação permite que o consumo de memória e o tempo de execução sejam definidos de acordo com alguns parâmetros de entrada.

Para avaliar o impacto do tamanho da memória da aplicação no tempo de gravação dos *checkpoints*, foram criadas aplicações com tempo médio de execução de 40 minutos e com consumo de memória variando de 139,57 MB a 7.751,68 MB. As gravações de *checkpoints* foram realizadas em períodos fixos de 5 minutos, totalizando uma média de 8 *checkpoints* por aplicação. O gráfico da Figura 2 apresenta os resultados dessa avaliação. Como pode ser visto, entre os serviços avaliados, o EBS apresentou os menores tempos de gravação. Para a aplicação com 139,57 MB o tempo médio foi de 0,65 segundos, enquanto para a aplicação com um pouco mais de 7 GBs o tempo médio foi de 55,81 segundos. Na sequência aparecem os serviços EFS, com tempos de 2,11 a 78,72 segundos, e o S3 com tempos variando de 19,86 a 215,68 segundos.

Considerando o caso em que a aplicação utilizou pouco mais que 7 GBs de memória, o EBS apresentou uma melhora de desempenho de 74,12% em relação ao S3 e de 29,11% em relação ao EFS. Como pode ser visto no gráfico, essa diferença diminui a medida que a memória da aplicação se aproxima dos 100 MBs. No entanto, embora o EBS apresente os menores tempos, vale ressaltar que ao utilizar o EBS o acesso aos arquivos é limitado a uma única MV, o que impede que aplicações compartilhando um mesmo volume EBS para armazenar os *checkpoints* sejam reiniciadas em diferentes MVs. Além disso, quando comparado ao S3, o EBS apresenta um custo monetário de armazenamento 77% mais

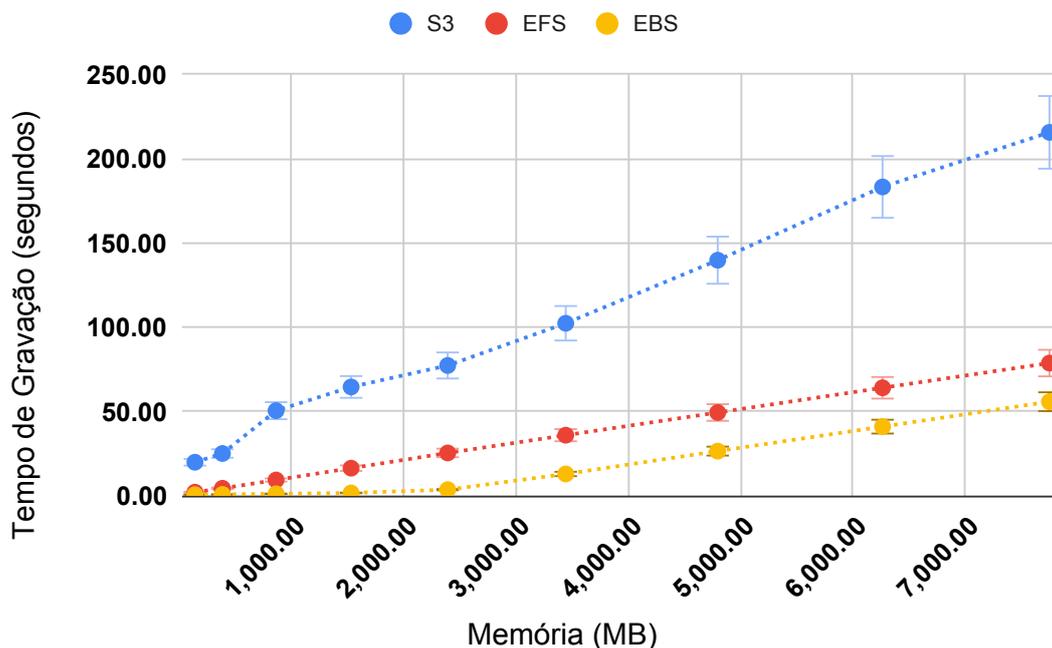


Figura 2. Tempo de gravação dos *checkpoints* no S3, EBS e EFS em relação ao tamanho da memória utilizada pela aplicação

caro. Portanto, existe uma relação clara entre o custo do armazenamento e o desempenho oferecido.

Ainda em relação aos custos monetários do EBS, o armazenamento neste serviço é 66% mais barato quando comparado aos preços praticados no EFS de acesso frequente. Porém, em comparação ao EFS de acesso de baixa frequência o volume do EBS é 75% mais caro. Como os arquivos de *checkpoints* provavelmente não se enquadrarão na classe de cobrança de baixa frequência, uma vez que os arquivos são escritos, possivelmente utilizados e descartados em intervalos inferiores a 7 dias, o EFS só é viável do ponto de vista financeiro caso o desempenho e a necessidade de um sistema de arquivos compartilhados sejam indispensáveis para a aplicação executada.

O baixo desempenho do S3 em relação aos demais serviços avaliados, observado na Figura 2, pode ser explicado em parte pela falta de suporte nativo para sistemas de arquivos, e a necessidade da criação de um sistema de arquivos utilizando uma ferramenta de terceiro. Sendo assim, o desempenho apresentado é fortemente relacionado a ferramenta utilizada. Além disso, podemos observar uma maior variabilidade do tempo de gravação no S3 com o aumento da memória da aplicação. Essa variabilidade não é observada nos demais serviços. Vale ressaltar ainda que, como o S3 não permite alterações diretas de arquivos já gravados, todos os arquivos que sofrem alterações durante o processo de *checkpoint*, são copiados localmente para o sistema de arquivo local da MV, alterados e reenviados para o serviço sempre que uma nova alteração é realizada. Além de causar uma sobrecarga adicional de transferência, isso faz com que a criação de *checkpoints* incrementais, no qual um novo *checkpoint* é gravado a partir dos arquivos de *checkpoints* anteriores, seja proibitivo se a gravação for feita diretamente no S3.

5.2. Caracterização do procedimento de recuperação

Para avaliar o impacto do procedimento de recuperação e a sobrecarga de alocação e configuração das MVs, uma aplicação com consumo de memória de 7 GBs e tempo de execução de 25 minutos foi utilizada em um cenário com uma única revogação. Assim como no teste anterior, foram utilizadas MVs do tipo *c3.2xlarge* e a revogação da MV primária foi feita de forma controlada: 15 minutos após a sua solicitação. Além disso, foram realizados um total de 4 gravações de *checkpoints* (2 na MV primária e 2 na MV *backup*) com intervalo fixo de 2 minutos. Os resultados deste teste são apresentados na Figura 3.

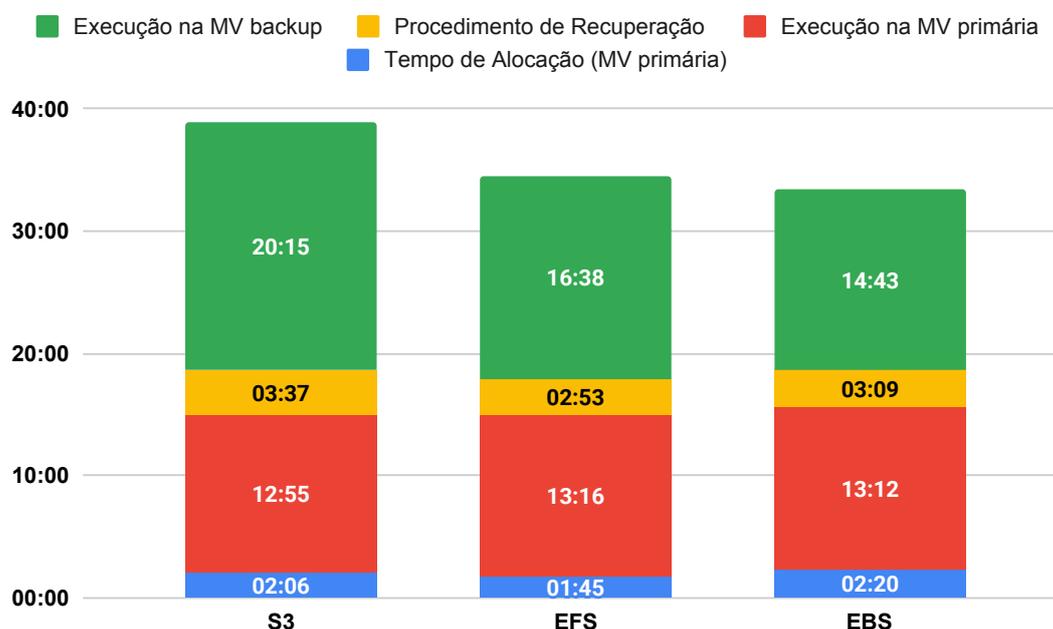


Figura 3. Tempos de alocação das MVs e execução da tarefa no cenário com revogação da MV spot

Os tempos apresentados na Figura 3 englobam todo o processo de execução, iniciando no momento em que o controlador recebe a solicitação dos usuários, e terminando com a finalização da MV *backup*. A primeira faixa de tempo, representada em azul, é referente ao tempo de alocação da MV primária e é composto pelo tempo de resposta do provedor, o tempo de inicialização da MV e de configuração do serviço de armazenamento escolhido. Como pode ser visto na figura, o tempo médio de alocação da MV primária sofre pouca variação em relação aos serviços de armazenamento avaliados. Com o EBS representando um acréscimo de pouco mais de 10 segundos em relação ao S3 e de 35 segundos em relação ao EFS. Esse acréscimo de tempo é esperado uma vez que, diferente dos demais serviços, no EBS o controlador executa uma etapa adicional durante a configuração do sistema de arquivos: a criação do volume.

Na sequência, em vermelho, é apresentado o tempo de execução da tarefa na MV primária. Assim, como no caso anterior, essa faixa de tempo também não apresenta diferenças significativas entre os serviços de armazenamento. Isso ocorre pois, nesta avaliação, a revogação da MV primária é realizada 15 minutos após a sua solicitação,

independente do serviço de armazenamento utilizado. Uma vez que a MV primária é revogada, é iniciado o processo de recuperação da aplicação. Esse processo, inclui o tempo de detecção da revogação, a solicitação da MV *backup* e o processo de reinicialização da tarefa. Como pode ser visto, o tempo médio do procedimento de recuperação também apresentou pouca variação entre os serviços.

A partir dos resultados apresentada na Figura 3, vemos que diferenças significativas nos tempos avaliados ocorrem apenas a partir da segunda metade da execução, durante a execução da MV *backup*. Essa variação de tempo é explicada pelas diferentes sobrecargas impostas pela gravação dos *checkpoints* (Seção 5.1). Sendo assim, podemos concluir que para a abordagem de CR avaliada, o impacto da escolha dos serviços de armazenamento se da, quase que exclusivamente, no desempenho da gravação dos *checkpoints* e não no processo de recuperação.

6. Conclusão

Neste trabalho foram avaliados os serviços de armazenamento EBS, EFS e S3 oferecidos pela AWS. As avaliações foram realizadas considerando a gravação e recuperação de *checkpoints*, no cenário em que a ocorrência de falhas se dá devido a revogação de MVs do mercado *spot*. Os resultados demonstraram que os serviço de armazenamento local, EBS, apresenta o melhor desempenho em relação ao tempo de gravação. Porém, esse serviço apresenta um custo monetário superior ao S3 (77% mais caro) e inferior ao EFS (66% mais barato), e não permite o compartilhamento dos arquivos de *checkpoint* entre diferentes MVs. Já o S3, apresenta o menor preço entre os serviços avaliados, porém teve o pior desempenho da avaliação.

Em trabalhos futuros serão avaliados o impacto de abordagens com gravação assíncrona de duas etapas, no desempenho destes serviços. Também serão avaliadas outras ferramentas de interface com o S3. Além disso, os resultados obtidos neste trabalho serão utilizados em um *framework* de escalonamento de aplicações, que utiliza técnicas de tolerância a falhas, incluindo CR, para a execução de aplicações em MVs *spots*.

Agradecimentos

Este trabalho foi financiado pelo *Programa Institucional de Internacionalização* (PrInt) da CAPES (processo 88887.310261/2018-00) e contou com apoio, por meio de bolsas de estudos, das agências CNPq e FAPERJ.

Referências

- Alves, M. M. and Drummond, L. M. (2017). A multivariate and quantitative model for predicting cross-application interference in virtual environments. *Journal of Systems and Software*, 128:150 – 163.
- AWS (2020a). Amazon EC2 pricing. <https://aws.amazon.com/ec2/pricing/>. Accessed 15 March 2020.
- AWS (2020b). Amazon Elastic Block Store. <https://aws.amazon.com/ebs/>. Accessed 15 March 2020.
- AWS (2020c). Amazon S3. <https://aws.amazon.com/s3/>. Accessed 15 March 2020.

- Di, S., Robert, Y., Vivien, F., Kondo, D., Wang, C.-L., and Cappello, F. (2013). Optimization of cloud task processing with checkpoint-restart mechanism. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, New York, NY, USA. Association for Computing Machinery.
- Egwutuoha, I. P., Levy, D., Selic, B., and Chen, S. (2013). A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *The Journal of Supercomputing*, 65(3):1302–1326.
- Elnozahy, E. N., Alvisi, L., Wang, Y.-M., and Johnson, D. B. (2002). A survey of rollback-recovery protocols in message-passing systems. *ACM Computing Surveys (CSUR)*, 34(3):375–408.
- EMELYANOV, P. (2011). Criu: Checkpoint/restore in userspace, july 2011. URL: <https://criu.org>.
- Hargrove, P. H. and Duell, J. C. (2006). Berkeley lab checkpoint/restart (BLCR) for linux clusters. *Journal of Physics: Conference Series*, 46:494–499.
- Meroufel, B. and Belalem, G. (2018). Optimization of checkpointing/recovery strategy in cloud computing with adaptive storage management. *Concurrency and Computation: Practice and Experience*, 30(24):e4906. e4906 CPE-17-0585.R2.
- Nicolae, B. and Cappello, F. (2011). Blobcr: Efficient checkpoint-restart for hpc applications on iaas clouds using virtual disk image snapshots. In *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, Seattle, WA, USA. IEEE.
- Poola, D., Ramamohanarao, K., and Buyya, R. (2014). Fault-tolerant workflow scheduling using spot instances on clouds. *Procedia Computer Science*, 29:523 – 533. 2014 International Conference on Computational Science.
- Rizun, R. (2011). S3fs: Fuse-based file system backed by amazon s3. <https://github.com/s3fs-fuse/s3fs-fuse/>.
- Ruiz-Alvarez, A. and Humphrey, M. (2011). An automated approach to cloud storage service selection. In *Proceedings of the 2nd International Workshop on Scientific Cloud Computing*, ScienceCloud '11, page 39–48, New York, NY, USA. Association for Computing Machinery.
- Services, A. W. (2020a). Amazon Elastic File System. <https://aws.amazon.com/efs/>. Accessed 07 April 2020.
- Services, A. W. (2020b). Boto 3 Documentation. <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>. Accessed 01 April 2020.
- Services, A. W. (2020c). Cloud Storage with AWS. https://aws.amazon.com/products/storage/?nc1=h_ls. Accessed 02 April 2020.
- Teylo, L., Arantes, L., Sens, P., and de A. Drummond, L. M. (2019). A hibernation aware dynamic scheduler for cloud environments. In *Proceedings of the 48th International Conference on Parallel Processing: Workshops, ICPP 2019*, New York, NY, USA. Association for Computing Machinery.

- Yaothanee, J. and Chanchio, K. (2019). An in-memory checkpoint-restart mechanism for a cluster of virtual machines. In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 131–136, Chonburi, Thailand, Thailand. IEEE.
- Yi, S., Kondo, D., and Andrzejak, A. (2010). Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 236–243, Miami, FL, USA. IEEE, IEEE.
- Zhou, A., Sun, Q., and Li, J. (2017). Enhancing reliability via checkpointing in cloud computing systems. *China Communications*, 14(7):1–10.