

Multi-Cloud

Alexandre Canalle¹, Ariel Frozza¹

¹Instituto de Informática – Universidade Católica do Paraná (PUC-PR)
Caixa Postal 15.064 – 91.501-970 – Curitiba – PR – Brazil

arielfrozza@gmail.com, roxsnd@gmail.com

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Abstract. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1. Introdução

Muitas organizações estão migrando para a Nuvem e procuram fazer esta migração de forma segura, confiável e sem “vendor lock in”. A maioria dos provedores (inclusive os líderes de mercado) de Nuvem usam/disponibilizam Software e APIs proprietárias e também não são aderentes à padrões de Cloud API como a Tosca [39] ou OASIS CAMP [41]. Além disso, todos os provedores de Cloud experimentam, eventualmente, períodos de indisponibilidade que podem causar impactos a negócios [].

Performance variável também é um problema por que a maioria dos provedores “overprovision” sua infraestrutura virtualizada e isto resulta em degradação de performance e qualidade de serviço [42].

Este artigo vai focar em frameworks e patterns que oferecem Softwares e APIs não proprietárias com as funcionalidades compatíveis/equivalentes com os softwares e serviços ofertados pelos principais provedores de Cloud

Este estudo intenciona responder às seguintes questões:

Como desenhamos “Deployment Patterns” em ambiente multi-cloud que ofereçam:

1) Tolerancia à falhas (faul-tolerance) envolvendo uso de mais de um provedor de Cloud.

2) Neutralidade de fornecedor (vendor neutrality), ability/possibility to run on major Cloud providers IaaS using open-source solutions to avoid “lock-in”.

3) Performance como habilidade para alterar/ajustar cargas para outros provedores de Cloud em caso de degradação de performance ou qualidade de serviço em um ou mais provedores.

4) Segurança

5) Eficiência de custos como habilidade de aproveitar os melhores preços de recursos computacionais e serviços em um ambiente multi-cloud.

Implementação Open-source da plataforma com padrão neutro (vendor-neutral) será desenvolvida/apresentada para validar padrões e guias de adoção. Usaremos open-source software para demonstrar como os padrões podem ser implementados. Fazemos este estudo mais útil/completo/específico que padrões genéricos como os descritos em <http://www.cloudpatterns.org>.

Custos de operação em um determinado provedor Cloud variam assim como os custos de trocar de provedor podem ficar muito altos caso tudo (o padrão em questão) tenha que ser refeito/revisto inteiramente. Usando padrões "vendor neutral" o objetivo seria ser capaz de mudar de provedor de cloud de acordo com a conveniência ou executar on-premises sem ter que alterar o software stack.

Outro benefício importante deste estudo é a ênfase em padrões "fault tolerant" que implica na garantia que sistemas atinjam uptime e resiliência à stress e falhas adequadas.

Por fim, este estudo (as soluções definidas/propostas) garante que os padrões funcionam também em cloud privadas, pois:

1) Custos de longo prazo para cloud pública podem ficar mais altos que on-premises devido ao OPEX da cloud pública eventualmente ultrapassa o CAPEX da cloud privada quando servidores e outros equipamentos são comprados.

2) Por causa de segurança ou motivos regulatórios, pode ser necessário manter parte do workload on-premises.

3) Performance on-premises pode ser melhor porque você controla a taxa de over-subscrição e cargas submetidas a uma mesma instância de virtualização.

Neste artigo vamos identificar e categorizar os principais desafios para atingir o objetivo principal. Em seguida, vamos identificar os atributos principais de qualidade desejados para o deploy dos padrões em Cloud. Depois, proporemos soluções de padrão de qualidade para cada padrão apresentado.

Todos os padrões serão unificados em um framework Multi-Cloud coeso.

Por fim, validaremos os padrões-chaves com experimentos para a "feasibility" dos casos de "fault-tolerance" e "self-healing" usando padrões open-source de implementação.

O resultado deste estudo será um catálogo de Padrões de Cloud "Fault Tolerant", Seguro e de alta performance com as soluções mapeadas com software open-source com exemplos do deploy (e alguns templates/pseudo-código) que podem ser usados para implementar um stack completo de infraestrutura que será compatível com a especificação OASIS CAMP (Cloud Application Management for Platforms) e mais tarde, com OASIS TOSCA (Topology and Orchestration Specification for Cloud Applications).

Em seguida, todos os padrões serão validados usando software Open Source. A

maioria das ofertas em cloud não seria possível sem uso de software open source (e.g. hypervisors Xen e KVM). Como selecionamos os melhores projetos open-source? We need to select the projects with a large and active community, quantos "committers" estão ativos, quantos contribuidores são de diferentes organizações ou este é um projeto open-source de uma única empresa (algumas empresas desenvolvem open-source software mas não aceitam contribuições externas - Read Only Opensource). Idealmente, melhor ainda se a solução open-source tiver alguma empresa mantenedora/responsável por oferecer serviços de consultoria e bug-fixes (caso a empresa não tenha todo o expertise in-house).

Multi-Cloud deployment for fault-tolerance, performance, security and cost efficiency. Current solutions and their limitations. Problem statement. Solution methodology. Expected contributions. Assumptions and objectives for supporting open-source multi-cloud deployment. Key challenges while deploying enterprise computing in Multi-Cloud Environment. Open-source Multi-Cloud deployment solution framework. Solution to major multi-cloud challenges.

2. Design Patterns and Frameworks for Effective Multi-Cloud Deployment

2.1. Multi-Cloud Foundation Patterns focused on Fault Tolerant Deployment Solutions

2.1.1. Initial Multi-Cloud General Deployment Fault Tolerance Challenges

2.1.2. Multi-Cloud Management after initial deployment and Dealing with Failures

2.1.3. Cost Efficiency Problems

2.1.4. Security Problems

2.1.5. Application Deployment in Multi-Cloud Environment

2.2. Multi-cloud management after initial Deployment and dealing with failures in automated way

2.2.1. title

2.3. Cost efficiency patterns

2.3.1. title

2.4. security related patterns

2.4.1. title

2.5. multi-cloud control plane framework

3. experimental validation

4. Conclusion

Conteúdo da sub-seção conteúdo da sub-seção conteúdo da sub-seção conteúdo da sub-seção. Conteúdo da sub-seção conteúdo da sub-seção conteúdo da sub-seção conteúdo da sub-seção.

conteúdo da sub-subseção Meu primeiro doc tex. À é ã ç %

Texto negrito. - Selecciona e CTRL + b

Text itálico. - Selecciona e ctrl + i

Text negrito e itálico

Text underline

Texto à esquerda

Texto centralizado.

Texto à direita.

Texto pequeno.

Texto Grande.

Texto Normal.

Parte I

title

Bla bla bla