

UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CURSO DE SISTEMAS DE INFORMACAO

LEONARDO AUGUSTO METZGER

OPTVM: UM SERVIÇO DE SUPORTE PARA MIGRAÇÃO DE VMS

TRABALHO DE CONCLUSÃO DE CURSO

SÃO BENTO DO SUL
2019

LEONARDO AUGUSTO METZGER

OPTVM: UM SERVIÇO DE SUPORTE PARA MIGRAÇÃO DE VMS

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Mário Ezequiel Augusto
Universidade do Estado de Santa Catarina

SÃO BENTO DO SUL
2019

TODO

AGRADECIMENTOS

TODO

RESUMO

METZGER, Leonardo. OptVM: Um serviço de suporte para migração de VMs. 2019. 17 f. Trabalho de Conclusão de Curso – Curso de Sistemas de Informacao, Universidade do Estado de Santa Catarina. São Bento do Sul, 2019.

O Resumo é um elemento obrigatório em tese, dissertação, monografia e TCC, constituído de uma seqüência de frases concisas e objetivas, fornecendo uma visão rápida e clara do conteúdo do estudo. O texto deverá conter no máximo 500 palavras e ser antecedido pela referência do estudo. Também, não deve conter citações. O resumo deve ser redigido em parágrafo único, espaçamento simples e seguido das palavras representativas do conteúdo do estudo, isto é, palavras-chave, em número de três a cinco, separadas entre si por ponto e finalizadas também por ponto. Usar o verbo na terceira pessoa do singular, com linguagem impessoal, bem como fazer uso, preferencialmente, da voz ativa. Texto contendo um único parágrafo.

Palavras-chave: Palavra. Segunda Palavra. Outra palavra.

ABSTRACT

METZGER, Leonardo. Title in English. 2019. 17 f. Trabalho de Conclusão de Curso – Curso de Sistemas de Informacao, Universidade do Estado de Santa Catarina. São Bento do Sul, 2019.

Elemento obrigatório em tese, dissertação, monografia e TCC. É a versão do resumo em português para o idioma de divulgação internacional. Deve ser antecedido pela referência do estudo. Deve aparecer em folha distinta do resumo em língua portuguesa e seguido das palavras representativas do conteúdo do estudo, isto é, das palavras-chave. Sugere-se a elaboração do resumo (Abstract) e das palavras-chave (Keywords) em inglês; para resumos em outras línguas, que não o inglês, consultar o departamento / curso de origem.

Keywords: VM. Optimization. Rest.

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 ORGANIZAÇÃO DO TRABALHO	1
2 – REVISÃO DE LITERATURA	2
2.1 Service Oriented Architecture	2
2.1.1 Simple Object Access Protocol	2
2.1.2 Representational State Transfer (REST)	3
2.2 Otimização multi-objetivo	3
2.2.1 Algoritmos Evolucionários	4
2.2.2 Algoritmos Genéticos	4
2.3 Trabalhos relacionados	4
2.3.1 Migração de máquinas virtuais	4
2.3.2 Otimização na escolha do host	5
3 – metodologia	6
3.1 delineamento da pesquisa	6
3.2 coleta e tratamento de dados	6
4 – OPTVM	7
4.1 Comunicação	7
4.2 Representação do serviço	7
4.3 Representação para o algoritmo	8
4.4 Utilização do algoritmo	8
5 – resultados	9
5.1 TRABALHOS FUTUROS	10
5.2 CONSIDERAÇÕES FINAIS	10
Referências	11
 Apêndices	 12
APÊNDICE A – Nome do apêndice	13
APÊNDICE B – Nome do outro apêndice	14
 Anexos	 15
ANEXO A – Nome do anexo	16
ANEXO B – Nome do outro anexo	17

1 INTRODUÇÃO

Com a evolução da computação em nuvem, surgiram necessidades cada vez maiores de utilizar ao máximo o poder dos computadores sem sobrecarregá-los. Estas necessidades surgem para atender requisitos de diminuição de custos, aumento de desempenho, entre outros objetivos que fazem com que usuários de computação em nuvem e empresas que usam este tipo de serviço obtenham vantagem no uso dela.

Para isso, é muito comum que para otimizar o uso dos computadores de um ambiente em nuvem, os provedores utilizem o mecanismo de virtualização. Hoje, os *datacenters* são compostos por máquinas físicas(FMs) e máquinas virtuais(VMs), sendo que, cada FM normalmente possui pelo menos uma ou mais VMs. Essa utilização das VMs permite que seja construído um ambiente flexível.

Em um cenário que o ambiente em que temos a possibilidade de utilizar as FMs como host de múltiplas VMs, é possível que as VMs da nuvem sejam organizadas de diferentes maneiras em relação as FMs para que atinjam os objetivos dos interessados. Os objetivos podem ser os mais variados. Por exemplo, uma empresa que use o serviço da nuvem pode querer ter um alto desempenho, assim como pode querer ter o menor custo possível. Por esse motivo, existem pesquisas que buscam maneiras de otimizar esses objetivos e buscar uma forma de resolver este tipo problema, o qual é chamado otimização baseado em múltiplos objetivos.

A migração de uma VM envolve algumas etapas, como, a descoberta da necessidade de migração, a escolha de uma VM a ser migrada e a escolha de um host de destino para essa VM. A etapa em que este trabalho está interessado é a escolha de um host de destino para a VM. Considerando que uma migração seja considerada cara do ponto de vista computacional. O momento da migração deve ser bem escolhido para evitar que a própria migração não incorra em prejuízos. Assim como o momento da migração é importante, a escolha de um destino também é, pois o host selecionado tem que atender os objetivos e restrições que a VM necessita, para que não haja a sobrecarga do host de destino e implique em uma nova migração.

O trabalho tem papel de servir como apoio para a migração de VMs em ambientes de computação em nuvem. O trabalho faz uso de uma abordagem em que um usuário, que precise migrar uma VM, possa utilizar um serviço que selecionará as melhores opções de host para fazer a migração de uma VM. O serviço possui uma abordagem que utiliza algoritmos que fazem a seleção do host baseado em múltiplos objetivos. Contudo, o serviço é uma caixa preta, esta característica traz uma grande vantagem, o usuário não precisa conhecer nada sobre os algoritmos utilizados, precisa apenas utilizar a interface que é definida pelo serviço.

1.1 ORGANIZAÇÃO DO TRABALHO

TODO

2 REVISÃO DE LITERATURA

Neste capítulo, são apresentados alguns conceitos e termos utilizados no decorrer do trabalho. Estes, também darão base para compreender a escolha das técnicas utilizadas. Os conceitos são relacionados a construção do sistema. Os conceitos estão relacionados a otimização multiobjetivo e comunicação cliente/servidor, que são os dois pilares do sistema que será desenvolvido.

2.1 Service Oriented Architecture

O desenvolvimento de software para um ambiente corporativo é uma tarefa complexa. Conforme Brown [Brown Simon Johnston 2002], no decorrer dos anos, a comunidade de desenvolvimento de software se dedicou em desenvolver novas abordagens, processos e ferramentas para a construção de softwares de grande escala.

Brown considera que uma maneira de descrever um sistema de software é como sendo um composto de uma coleção de serviços. Cada serviço, provém um conjunto de funcionalidades bem definidas. As funcionalidades do serviço sendo bem definidas e sólidas, torna possível a construção de serviços compostos, ou seja, uma funcionalidade que faça a utilização de outras funcionalidades ou serviços. Esta modularização e coordenação de serviços e funcionalidades caracteriza um Service Oriented Architecture (SOA).

Segundo [Valipour Bavar Amirzafari e Daneshpour 2009], SOA pode ser definido como um design de software utilizado para conectar negócios e recursos computacionais sob demanda, e isso possibilita os usuários do serviço (podendo ser outros serviços ou usuários finais) alcançarem seus objetivos.

Existem diversas maneiras de implementar uma aplicação baseada em SOA, o importante é que sua interface seja bem definida com as operações que podem ser realizadas. Uma das grandes vantagens do SOA, é a facilidade que ele provém na integração de sistemas. Segundo [Valipour Bavar Amirzafari e Daneshpour 2009], com as operações bem definidas e disponíveis, o consumidor do SOA, pode se preocupar somente com o que determinado serviço faz e não como é implementado.

As principais características de um software feito utilizando SOA são que ele é auto contido e modular, interoperável, fracamente acoplado, passível de composição e possui transparência de localização. Como SOA não limita a estratégia utilizada para o desenvolvimento do mesmo, pode-se utilizar qualquer técnica para implementá-lo. No ambiente corporativo, os serviços comumente são implementados utilizando web services SOAP, REST ou chamadas RPCs.

2.1.1 Simple Object Access Protocol

SOAP é um protocolo de comunicação baseado em XML(eXtension Markup Language) que foi criado no final dos anos 90. Seu objetivo é fazer a comunicação entre o cliente e o servidor através de informações passadas através de um documento XML. O protocolo utiliza um *schema* XML, que é uma maneira de descrever e validar o

formato os dados das requisições e respostas. Esse *schema* é utilizado pelo cliente e pelo servidor para saber como interpretar a resposta, no caso de recebimento de mensagem, e formatar a requisição, no caso de envio.

O objetivo do SOAP, é expor regras de negócio de aplicação através de serviços. Por esse motivo, o SOAP é uma opção comumente utilizada na construção de aplicações SOA. Outra característica do SOAP, é que ele não precisa ser implementado sobre um protocolo de transporte específico, é possível implementar utilizando outros protocolos, porém, na maioria das vezes é utilizado HTTP.

O SOAP algumas vezes é comparado com o REST, pois os dois podem ser utilizados para um objetivo semelhante, porém, os dois tem um foco diferente, onde o SOAP tem como expor regras de negócio como serviço e o REST visa representar um determinado estado e manipulá-lo através de operações bem definidas.

2.1.2 Representational State Transfer (REST)

REST foi formalizado por Fielding [Fielding 2000] em sua tese de doutorado, onde ele tem por objetivo apresentar uma arquitetura para criação de sistemas network-based. Na tese, REST é definido como um estilo arquitetural. Ele define uma série de restrições que devem ser respeitadas na criação de um software que é implementado utilizando este estilo arquitetural. As características de um software que utiliza o estilo REST serão apresentadas a seguir, assim como algumas de suas vantagens e desvantagens.

Uma das principais características é que o REST é implementado utilizando o modelo de comunicação cliente-servidor. Isso ajuda com a separação de responsabilidades, e permite que uma portabilidade de clientes do serviço implementado. Além disso, essa separação também permite que outros serviços façam uso do serviço REST. Do ponto de vista de arquitetura de software, isto é muito importante, pois permite que componentes fiquem bem modularizados.

Outra característica que o estilo arquitetural tem, é que os serviços devem ser stateless, ou seja, as requisições devem ser auto-contidas, não podem assumir algum estado ou contexto que o servidor tenha previamente armazenado.

Fielding [Fielding 2000] destaca que uma das características centrais do REST, e o que difere ele de outros estilos arquiteturais, é a utilização de uma interface uniforme entre os componentes. Esta interface uniforme é um ponto muito positivo, pois permite que a comunicação entre os componentes da arquitetura seja feita de maneira genérica, o que permite escalar a comunicação entre aplicações.

2.2 Otimização multi-objetivo

Conforme Nos dias de hoje, problemas de otimização buscam um bom resultado. Segundo Veloso [Matos 2017] para os problemas de otimização, existem de maneira geral, dois tipos de problemas, os mono-objetivos e os multi-objetivos. Os mono-objetivos buscam otimizar uma solução baseando-se em um único objetivo, por consequência, problemas mono-objetivo resultam em um único resultado, que pode ser considerada a solução ótima para o problema. Já os multi-objetivo, buscam atender vários fatores, e isso torna a solução ótima mais difícil de ser encontrada.

Conforme Ticona [Ticona 2003] um problema de otimização multi-objetivo, é representado por um conjunto de funções objetivo que devem ser otimizadas.

2.2.1 Algoritmos Evolucionários

Segundo [Ticona 2003], algoritmos evolucionários (AE) tem sido muito utilizados para problemas de otimização. Um dos motivos do uso deles, é por causa da possibilidade de resolver problemas que envolvam múltiplos objetivos. A abordagem utilizada neste tipo de algoritmo é baseada na evolução humana. O processo é baseado seleção natural de Darwin, da mesma maneira que acontece com a seleção das espécies. O algoritmo reproduz artificialmente o processo de seleção natural para encontrar os mais aptos a resolver determinado problema. O objetivo desses algoritmos é encontrar aproximações da solução perfeita para problemas difíceis.

Dentro da categoria dos AEs para otimização baseada em múltiplos objetivos, existem diferentes modelos.

O modelo de utilizado neste trabalho é a de algoritmos genéticos (AG). Esta é uma classe de algoritmos muito utilizada em otimizações multi-objetivo.

2.2.2 Algoritmos Genéticos

Existem duas abordagens principais para AGs multi-objetivo. Uma delas utiliza pesos para objetivos únicos. E a outra forma, seleciona um subconjunto de um conjunto de possíveis soluções que não é dominada por nenhuma das outras soluções, este subconjunto é chamado de lista de Pareto, também conhecida por soluções Pareto-ótimas.

Nas otimizações multi-objetivo que utilizam soluções pareto-ótimas, usa-se o conceito de dominância de Pareto para alcançar soluções que sejam mais adequadas para determinado problema. Segundo Ticona [Ticona 2003], a dominância de um item \mathbf{x} sobre um item \mathbf{y} se dá quando as seguintes condições são atendidas:

1. A solução \mathbf{x} é igualmente ou mais adequada que a solução \mathbf{y} em todas as funções objetivo
2. A solução \mathbf{x} é melhor do que a solução \mathbf{y} em algum objetivo

2.3 Trabalhos relacionados

Neste capítulo será apresentado o problema em que o OptVM se propoem resolver. Isso será feito através da apresentação de alguns fatos relacionados a nuvens computacionais e como elas costumam ser utilizadas nos dias de hoje.

2.3.1 Migração de máquinas virtuais

As nuvens computacionais são utilizadas pela maioria das empresas de software da atualidade. Por esse motivo, as maiores empresas do setor investem muito neste segmento, oferecendo vários tipos de serviços diferenciados para seus consumidores. Estas empresas concorrem em alguns aspectos, como: velocidade, preço, disponibilidade e etc. Para aumentar sua competitividade nesses aspectos, muitas vezes é utilizada a virtualização.

Com a virtualização é possível alocar partes de um recurso físico para diferentes consumidores, fazendo com que um recurso físico se torne melhor utilizado. Isso deixa a alocação de recursos muito mais flexível, e torna possível obter uma elasticidade nos serviços oferecidos.

Uma dos objetivos de utilizar a virtualização é obter uma elasticidade dos recursos oferecidos. Ou seja, é possível aumentar sua capacidade de processamento, armazenamento mesmo depois que o já foi alocada uma VM para o usuário. Isso permite que um consumidor do serviço possa escolher o quanto precisa para executar as tarefas que deseja, assim como o provedor também consegue otimizar o uso de seus recursos

Essa realocação de uma VM pode ser feita a nível de nuvem, datacenter(DC) ou host. Quando a realocação é feita em nível de DC ou nuvem, é muito provável que seja necessário migrar uma VM do local em que ela se encontra. Caso seja necessário uma migração de uma máquina, existem alguns pontos que devem ser avaliados.

Três momentos podem ser considerados os pontos principais a serem avaliados para uma migração, são eles:

1. A descoberta de uma necessidade de migração
2. Qual máquina virtual deve ser migrada
3. Para onde deve ocorrer a migração

Estas otimizações e migrações das VMs são necessárias em ambientes que envolvem uma infraestrutura grande, onde existem múltiplos hosts, datacenters e nuvens. Por esse motivo, não é recomendado que um sistema ou serviço gerencie a infraestrutura inteira sozinho, pois sua escalabilidade poderia se tornar um gargalo. Isso faz com que sejam construídos diferentes serviços e aplicações que se integram e gerenciam a infraestrutura.

2.3.2 Otimização na escolha do host

Como citado anteriormente uma das partes essenciais na migração de uma VM é a escolha de um destino para ela. Para isso, é importante escolher um destino que aloque muito bem a VM e não seja necessário fazer uma outra migração logo em seguida.

3 metodologia

cada capítulo deve conter uma pequena introdução (tipicamente, um ou dois parágrafos) que deve deixar claro o objetivo e o que será discutido no capítulo, bem como a organização do capítulo.

3.1 delinearmento da pesquisa

3.2 coleta e tratamento de dados

4 OPTVM

O OptVM é um sistema que tem o propósito de dar suporte para a migração de VMs através de serviços utilizando o modelo cliente/servidor. O sistema possui dois principais serviços para atingir seu objetivo: um faz a filtragem de hosts aplicando restrições definidas pelo cliente do serviço e a outra destina-se a quem precisa definir os melhores hosts para migrar uma VM baseando-se em objetivos também definidos pelo cliente.

Por lidar com conhecimentos específicos, o OptVM busca ser uma solução caixa preta, onde, o usuário não necessita saber nada sobre o funcionamento interno, algoritmos utilizados, etc. Basta utilizar suas APIs para fazer uso de suas funcionalidades.

Neste capítulo, serão apresentadas uma visão geral da implementação do OptVM. No primeiro momento será falado sobre o modelo de comunicação que foi escolhido e o porquê. Após isso, técnicas e ferramentas utilizadas para a construção dos serviços de aplicação das constraints e também da otimização.

4.1 Comunicação

Em termos gerais, uma API é uma interface de software que pode se

Como o OptVM é um serviço que deve ser disponibilizado para uma arquitetura de cliente/servidor de maneira distribuída, haviam três possíveis maneiras de implementá-lo, que eram REST, SOAP e via chamadas RPC. Para o desenvolvimento do OptVM o foi escolhido implementação utilizando o modelo REST. A escolha desta opção se deu pelos seguintes motivos:

1. É um padrão arquitetural bastante maduro;
2. É agnóstico em relação a linguagens de programação;
3. É bastante flexível em relação ao modelo de comunicação.

O padrão REST, definido por Fielding, sugere que se deve criar uma interface para interação com o sistema. Essa interface é representada através de recursos.

O OptVM trabalha em cima de um único recurso, chamado *otpimizations*.

4.2 Representação do serviço

O padrão arquitetural REST é agnostico em relação ao formato utilizado para fazer a comunicação dos dados. O *encoding* dos dados pode ser feito da maneira que for mais conveniente para o usuário. No caso do OptVM

```

1 {
2   "id": 1,
3   "objectives": [
4     "MIN_SOMETHING",
5     "MAX_SOMETHING"
6   ],
7   "hosts": [

```

```
8 {
9   "id": 1,
10  "memory": 2000,
11  "bandwidth": 1000,
12  "vms": [
13    {
14      "id": 10,
15      "space": 20,
16      "memory": 150
17    }
18  ]
19 }
20 ]
21 }
```

4.3 Representação para o algoritmo

Nos algoritmos genéticos, existem vários tipos de representações que podem ser utilizadas para representar nossa solução. Quando utiliza-se o algoritmo, uma boa representação ajuda a melhorar os resultados do algoritmo.

A representação escolhida para esse serviço foi a representação por inteiros, onde, cada inteiro representa um host e a posição em um vetor de inteiros representa uma VM.

A primeira posição "0" significa que a VM 0 está presente no HOST 10, a VM 1 está presente no HOST 20, e assim sucessivamente.

4.4 Utilização do algoritmo

TODO

5 resultados

cada capítulo deve conter uma pequena introdução (tipicamente, um ou dois parágrafos) que deve deixar claro o objetivo e o que será discutido no capítulo, bem como a organização do capítulo.

chapterCONCLUSÃO

5.1 TRABALHOS FUTUROS

5.2 CONSIDERAÇÕES FINAIS

Referências

BROWN SIMON JOHNSTON, K. K. A. Using service-oriented architecture and component based development to build web service applications. Rational Software, 2002. Citado na página 2.

FIELDING, R. T. Architectural styles and the design of network-based software architectures. University of California, 2000. Citado na página 3.

MATOS, A. V. de. A migração de máquinas virtuais no gerenciamento de recursos em ambientes de nuvens computacionais. Universidade Federal do Paraná (UFPR), 2017. Citado na página 3.

TICONA, A. C. B. D. W. G. C. Algoritmos evolutivos para otimização multi-objetivo. Universidade de São Paulo, 2003. Citado na página 4.

VALIPOUR BAVAR AMIRZAFARI, K. N. M. M. H.; DANESHPOUR, N. A brief survey of software architecture concepts and service oriented architecture. Department of Electrical and Computer Engineering Shahid Rajaee University, 2009. Citado na página 2.

Apêndices

APÊNDICE A – Nome do apêndice

Lembre-se que a diferença entre apêndice e anexo diz respeito à autoria do texto e/ou material ali colocado.

Caso o material ou texto suplementar ou complementar seja de sua autoria, então ele deverá ser colocado como um apêndice. Porém, caso a autoria seja de terceiros, então o material ou texto deverá ser colocado como anexo.

Caso seja conveniente, podem ser criados outros apêndices para o seu trabalho acadêmico. Basta recortar e colar este trecho neste mesmo documento. Lembre-se de alterar o "label" do apêndice.

Não é aconselhável colocar tudo que é complementar em um único apêndice. Organize os apêndices de modo que, em cada um deles, haja um único tipo de conteúdo. Isso facilita a leitura e compreensão para o leitor do trabalho.

APÊNDICE B – Nome do outro apêndice

conteúdo do novo apêndice

Anexos

ANEXO A – Nome do anexo

Lembre-se que a diferença entre apêndice e anexo diz respeito à autoria do texto e/ou material ali colocado.

Caso o material ou texto suplementar ou complementar seja de sua autoria, então ele deverá ser colocado como um apêndice. Porém, caso a autoria seja de terceiros, então o material ou texto deverá ser colocado como anexo.

Caso seja conveniente, podem ser criados outros anexos para o seu trabalho acadêmico. Basta recortar e colar este trecho neste mesmo documento. Lembre-se de alterar o "label" do anexo.

Organize seus anexos de modo a que, em cada um deles, haja um único tipo de conteúdo. Isso facilita a leitura e compreensão para o leitor do trabalho. É para ele que você escreve.

ANEXO B – Nome do outro anexo

conteúdo do outro anexo