

Aluno/a: _____ Mat: _____

Terceira Avaliação - 2022-2

1. Descrição do Relatório Técnico

O projeto constitui de uma atividade que precisa ser completada através de duas tarefas: (i) implementação e (ii) relatório. A primeira tarefa envolve-se com a construção do projeto que deve ser acompanhado de um relatório (conforme instruções) no formato de Artigo Técnico modelo SBRC. Os artigos podem ser escritos em português ou inglês e devem ser submetidos somente no formato PDF. Cada relatório não tem limite de páginas. Os trabalhos devem ser formatados seguindo o modelo de artigos da SBRC, disponível em [\[https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros\]](https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros) (Modelos para Publicação de Artigos). O relatório deverá conter os tópicos abaixo discriminados:

- **Título, Autores:** dispostos na parte superior da página, de modo que logo em seguida venha o resumo do projeto.
- **Implementação e Testes:** importante considerar a escrita para alguém não familiar com o programa e que o mesmo consiga entender o que foi feito. Na sequência, resultados obtidos da execução do programa devem ser listados e apresentados. Como mencionado anteriormente, a forma mais apropriada para apresentar os dados coletados é no formato de tabela.
- **Análise e Resultados da Implementação:** nesta seção, os resultados apresentados anteriormente serão analisados, discutidos e explicados, ou seja, algo que faça sentido deverá ser apresentado. Ela poderá se referir aos propósitos do experimento. Lembre-se também de explicar anomalias encontradas nos resultados. Tendências nos resultados devem ser evidenciadas bem como explicadas. De modo geral, esta seção oferece ao leitor uma explanação se e como o programa agiu e se comportou segundo os propósitos descritos e contemplados na programação. Gráficos são excelentes ferramentas para ilustrar os resultados e apoiar a escrita.

- **Conclusão:** deve reforçar a informação já apresentada pelo estudante, bem como apresentar resumidamente o experimento. Adicionalmente, incluir uma ou duas sentenças que respondam às seguintes perguntas:
 - Qual o propósito do experimento?
 - Tal propósito foi alcançado?
 - Quais foram os erros e dificuldades? Quais foram superados e como?
 - Quais foram os resultados do experimento?
 - Que informação foi descoberta com a execução dos programas?
 - O que foi aprendido com a realização dos experimentos?
 - O que se pode fazer para melhorar?
- **Apêndice:** deverá conter fragmento dos códigos fontes que foi produzido como parte do experimento e considerado pelos autores pontos chaves de sua solução e suporte para explicação do texto.

2. Descrição do Projeto

A segunda tarefa refere-se ao entendimento, projeto, implementação e execução do programa a ser desenvolvido que trata de assuntos cobertos em sala de aula e na teoria.

Alocação de Recursos - O problema de mapeamento está diretamente relacionado com a alocação de recursos, esse é um assunto que permeia várias áreas da computação. O mapeamento é intrinsecamente relacionado com escolhas, logo, naturalmente é um problema combinacional. O mapeamento é assunto em várias disciplinas da computação, como por exemplo: sistemas operacionais (mapeamento de processos na memória) e redes de computadores (mapeamento de funções de rede) e engenharia de software (mapeamento de serviços na arquitetura de micro-serviços).

No sistema 5G o mapeamento ocorre dentro do sistema de criação de fatias de redes, nomeado de *network slicing as a service* (NSaaS). Fatias são redes virtuais criadas sob demanda. Os desafios entre a associação de redes virtuais em redes reais é a longo prazo investigado na matemática, e nessa ciência o problema é conhecido como *virtual network embedding* (VNE). Desse modo, as fatias de redes são uma instância do problema de VNE.

Problema de Mapeamento - Formalmente, o problema VNE pode ser descrito como um grafo. Seja $SN = (S, L)$ o substrato da rede, onde N representa o conjunto de nós do substrato e L o conjunto de enlaces. Seja, $VNR_i = (N_i, L_i)$ um conjunto de $i = 1, \dots, n$ requisições de redes virtuais (em inglês *Virtual Network Request* - VNR) onde N_i e L_i representa o conjunto de nós e enlaces virtuais de uma respectiva VNR_i . Adicionalmente, seja R um vetor de espaço de recursos R_1, \dots, R_m e seja cap uma função que associa recursos disponíveis aos elementos do substrato da rede. Finalmente, para cada VNR_i , seja dem_i uma função que associa demandas de todos elementos de todas requisições [1].

Logo, VNE consiste de funções f_i e g_i que devem respeitar a disponibilidade de recursos durante todo o processo de mapeamento, uma vez que a inclusão de uma rede virtual implica na redução de recursos de forma proporcional à sua demanda. Juntos eles formam um mapeamento para uma requisição VNR_i . Não é requerido que essas duas funções sejam calculadas em uma única etapa, o cálculo pode ser dividido em múltiplas etapas [1]. A Tabela 01 resume as definições apresentadas nesta seção.

Tabela 01: Formulação matemática do problema de incorporação de redes virtuais.

Termo	Descrição
$SN = (N, L)$	SN é um substrato da rede, consistindo de $ N $ nós e $ L $ enlaces.
$VNR^i = (N^i, L^i)$	VNR^i denota o i -ésimo requisição, consistindo de $ N^i $ nós e $ L^i $ enlaces.
$R = \prod_{j=1}^m R_j$	Quantidade total de recursos disponíveis.
$cap : N \cup L \rightarrow R$	Função de associação de recursos aos nós e enlaces. Faz um acréscimo na quantidade disponível de recursos associados aos nós e enlaces físicos.
$dem_i : N_i \cup L_i \rightarrow R$	Função de alocação das demandas dos nós e enlaces virtuais aos reais. Faz um decréscimo da disponibilidade de recursos.
$f_i : N_i \rightarrow N$	função de mapeamento de um nó virtual pertencente a uma requisição i para um nó físico do substrato.
$g_i : L_i \rightarrow L \subseteq SN$	função de mapeamento de um enlace virtual pertencente a uma requisição i para um nó físico do substrato.

Esta formulação é uma abstração em alto nível do problema de mapeamento. Quais são os recursos dos nós e dos enlaces não estão pontualmente descritos, assim utiliza-se apenas o termo recurso de forma generalista. A incorporação de redes virtuais em cenários concretos implica em descrever as restrições de tais cenários, por exemplo, restrições como a localização dos nós ou das características dos enlaces.

Resolução do Problema - Em resumo, desenvolver uma solução para o problema de mapeamento de redes virtuais utilizando algoritmos genéticos e paralisação e distribuição para reduzir o tempo de processamento.

As requisições e a rede física (infraestrutura ou substrato) devem ser representadas por meio de grafos segundo a formulação matemática definida anteriormente. O recurso do nó é somente a CPU e o recurso do enlace é somente a largura de banda, ambos descritos em valores inteiros e positivos. Lembre-se que uma requisição onde o número de nós e enlaces virtuais forem maior que o número dos reais, essa requisição deve ser rejeitada. Se o algoritmo de mapeamento não conseguir encontrar uma solução, então a requisição deve ser rejeitada. A saber, no problema de mapeamento, a dimensionalidade se refere a quantidade de nós virtuais; e a taxa de mapeamento é a quantidade de requisições atendidas, ou seja, mapeadas. Além da taxa de mapeamento, o tempo é um fator importante para qualificar um algoritmo de mapeamento. Este trabalho envolve as seguintes atividades de pesquisas e desenvolvimento:

1. Resolver o problema de mapeamento usando Algoritmo Genético.
 - a. Criar um indivíduo que representa um possível mapeamento;
 - b. Criar uma função de avaliação para pontuar um mapeamento;
 - c. Criar uma forma para qualificar a avaliação.
 - d. Criar uma técnica para maximizar a quantidade de mapeamentos.
2. Paralelizar o algoritmo genético a fim de reduzir o tempo total do mapeamento.
3. Utilizar da distribuição para reduzir o tempo total do mapeamento.
4. Comparar: Sequencial versus Paralelo e Distribuído.

Recomendações

1. Antes de projetar a versão paralela e a distribuída é necessário ter a certeza que seu algoritmo:
 - a. resolve o problema.
 - b. está correto, ou seja, está respeitando as regras do mapeamento.
 - c. está otimizado.

2. A versão paralela pode ser por Threads ou SubProcess. Qual será a melhor para qual situação? Sabemos que Thread compartilha os dados e que subprocessos aproveitam melhor a quantidade de unidades lógicas de execução do processador.
 - a. Perceba que em certas configurações das tuplas [(repetição, população)] uma técnica pode se comportar melhor que a outra!
 - b. Ao projetar uma versão paralela há necessidade de refletir qual sub tarefa que pode ser paralelizada. Veja que o sistema tem que mapear uma requisição e procurar a quantidade de requisições mapeadas. Será que podemos pensar em alguma heurística para isto? Vocês conhecem o problema da mochila?
3. Distribuição - Podemos distribuir partes da lógica do algoritmo ou ter versões completas do algoritmos agindo de forma competitiva ou colaborativa. A distribuição sempre implica em acréscimo da latência ao tempo final. A paralelização horizontal é uma paralelização que usa Distribuição [2,3]. Ao fazer a distribuição temos que garantir que haverá máquinas/containers com um servidor/processo para receber a tarefa. É de se esperar que eles recebam os meta-parâmetros (repetição e tamanho da população) e/ou a população. Como retorno podem enviar a última população ou apenas o indivíduo mais apto.
4. População - A população inicial é criada aleatoriamente, cada indivíduo é criado usando valores randômicos. Assim, ao testar várias versões, temos que garantir de usar sempre a mesma população, a fim de evitar que a aleatoriedade favoreça uma instância qualquer.

Artefatos de apoio

Artefato	Link
População	https://drive.google.com/file/d/1bweZ6XXzVIWBVQxcCh8OSLkrkKrDk-hL/view?usp=sharing
Requisições	https://drive.google.com/file/d/1aiHtj5BfK69kS_EP_-c-Yxu5WEKpqPR/view?usp=sharing
Substrato	https://drive.google.com/file/d/1hy0bE65h9wBdYdpiGQZaaJzOUxES_fge/view?usp=sharing

JSON (vídeo)	https://www.youtube.com/watch?v=pTT7HMqDnJw
Networkx (vídeo)	https://www.youtube.com/watch?v=PfT8_2sKReo

Implementação

O projeto compreende a implementação de três versões da solução do mapeamento usando algoritmo genético: versão sequencial, paralela e distribuída. Todas as versões devem ser implementadas usando Python. Para a versão distribuída deve se usar Docker e a comunicação pode ser implementada por RPC, Socket ou MOM. Todos os processos serão abrigados na mesma máquina, mas cada em um *containers* diferente. A Tabela a seguir apresenta as especificações que devem ser seguidas.

Especificação	Opções
Sistema operacional	Windows ou Linux
Arquitetura de comunicação	TCP/IP
Linguagem de programação	Python, versão > 3.0
Comunicação entre processos	Sockets, RPC ou MOM
Interface de uso	Modo texto
Interface de teste	Modo texto

Resultados

Com este projeto esperamos responder algumas perguntas:

1. Qual a vantagem e desvantagem do Sequencial, Paralelo e Distribuído?
2. Como sua solução resolve o mapeamento?
3. Como a sua solução maximiza a quantidade de mapeamentos?
4. Qual a técnica para paralelizar a solução?
5. Qual a técnica para distribuir a carga de trabalho?

Pontuação

Entrega	Valor (pontos)
Relatório em PDF	2
Sequencial	2
Paralelo	2
Distribuído	2
Apresentação	2

Penalização

Ausência	Penalidade
Ausência do relatório	trabalho rejeitado
Ausência do código fonte	trabalho rejeitado
Não utilização do Python	trabalho rejeitado
Código fonte não roda conforme o relatório	trabalho rejeitado
Entrega além do prazo	Data de entrega é 14/03/2023, o trabalho poderá ser entregue até 16/03/2023, porém o trabalho valendo 9 de 10, ou seja, 1 ponto de penalização.

Realização e Entrega

Observe os itens abaixo, eles são obrigatórios, o não atendimento implica em eliminação da 3ª avaliação, logo o aluno deve atender a todos os pontos para validar o trabalho. São eles:

1. Trabalho individual ou grupo de até 2 alunos. **O NOME dos alunos deve constar no relatório!!!!**
2. Envio somente pelo SIGAA.
3. Data de entrega até 14/03/2023 às 23:55.
4. Data de apresentação: 16/03 - 23/03
5. Exame Final: 28/03/2023

6. O relatório e os códigos devem ser enviados em um único arquivo compactado (ZIP ou TAR.GZ) contendo:
 - a. Relatório
 - b. Arquivos fontes dos códigos
7. O código fonte deve conter comentários informativos. Para cada função ou método espera-se um texto explicativo sobre o bloco de código, uma descrição dos parâmetros e o que ele deve retornar.
8. Os trabalhos deverão ser apresentados, este requisito pode ser dispensado pelo professor em qualquer momento para todos os grupos de forma indiscriminada.
9. As apresentações serão marcadas individualmente com os grupos.

REFERENCIAL TÉCNICO

1. Fischer, A., Botero, J. F., Beck, M. T., de Meer, H., & Hesselbach, X. (2013). Virtual network embedding: A survey. *IEEE Communications Surveys and Tutorials*, 15(4), 1888–1906.
<https://doi.org/10.1109/SURV.2013.013013.00155>
2. Gomes, R., Vieira, D., & Castro, M. F. de. (2022). Vertical Parallelization of Differential Evolution Heuristic for Network Slicing in 5G Scenarios. *Anais Do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2022)*, 308–321.
<https://doi.org/10.5753/sbrc.2022.222314>
3. Gomes, R., Vieira, D., Leonel, F., Silva, F., & de Castro, M. (2022). Parallel Differential Evolution Meta-Heuristics and Modeling for Network Slicing in 5G Scenarios. In IARIA (Ed.), *The Eighteenth International Conference on Networking and Services - ICNS 2022*. International Conference on Networking and Services ICNS.
<https://www.iaria.org/conferences2022/ICNS22.html>
4. Gomes, R., Vieira, D., & de Castro, M. F. (2022). Application of Meta-Heuristics in 5G Network Slicing: A Systematic Review of the Literature. *Sensors*, 22(18), 6724.
<https://doi.org/10.3390/s22186724>

Bom Trabalho!