

Aluno/a: \_\_\_\_\_ Mat: \_\_\_\_\_

## Terceira Avaliação

### 1. Descrição do Relatório Técnico

Cada projeto constitui uma atividade que precisa ser completada através de duas tarefas, implementação e relatório.

A primeira envolve-se com a construção de cada 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 o código fonte que foi produzido como parte do experimento.

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

## 2. Descrição do Projeto

**Processamento Paralelo e Distribuído.** Servidores de alto desempenho são necessários para uma abundância de aplicações que necessitam de processamento intenso, por exemplo, processamento de criptografia, renderização, multiplicação de tensores de alta dimensionalidade. Novos paradigmas como *Cloud* e *Mobile Edge Computing* centralizam em certos limites o processamento. Novos modelos de rede como *Software-Defined Network* (SDN), vital para o 5G e futuras redes, centralizam a lógica da rede no controlador, que exerce um papel centralizador da coerência da rede. Portanto, o projeto de componentes guiados para suportar a alta demanda de computação é vital durante a fase de projeto inserido nos contextos supracitados e análogos.

**O projeto.** Desenvolver um *super-servidor* para multiplicação de matrizes, doravante, simplesmente chamado de o servidor. O servidor deve atender vários clientes, cada cliente deve enviar duas ou mais matrizes e receber uma terceira matriz resultado do produto da multiplicação das enviadas.

O servidor pode ser acessado via UDP pelos clientes. Quando o servidor estiver atendendo a requisição de um cliente ele também deve estar apto a atender outros. Logo o servidor

deve fazer uso de *Threads* para diferentes conexões e clientes. O objetivo é que o servidor implemente a alta disponibilidade de acesso.

O servidor pode no máximo atender  $n$  requisições ao mesmo tempo, o valor de  $n$  deve ser passado na hora da execução do servidor, as demais requisições ( $n+1$ ) ele deve passar aos servidores parceiros. Os servidores parceiros são desconhecidos pelos clientes. Todo servidor-parceiro quando é executado deve conectar ao servidor via TCP e se registrar, no registro deve informar quantos clientes consegue atender. Logo, o servidor e os servidores parceiros trabalham de forma colaborativa, ademais, a interação entre o servidor e os servidores-parceiros se dará por meio de *SubProcessos* e não por *Threads*, assim vamos prevenir *race-condition* na codificação.

Os clientes não devem conhecer a forma como o servidor funciona e nem da colaboração com seus parceiros, o que ele espera é receber a multiplicação de duas ou mais matrizes.

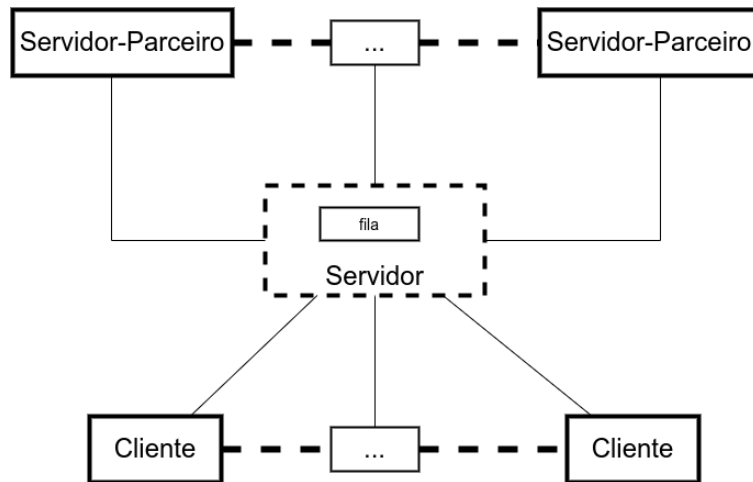
Os servidores parceiros devem ser mantidos em uma fila (a fila mantida pelo servidor) e ordenados pela taxa de desempenho. Os servidores mais rápidos devem ficar na frente. Sempre que um servidor-parceiro for escolhido ele sai da fila de disponíveis e vai para fila de indisponíveis, ou seja, os indisponíveis são os que estão em uso e, portanto, não podem ser escolhidos. O servidor-parceiro ao terminar a multiplicação deve enviar a resposta ao servidor, e, conseqüentemente, o servidor deve recalcular os índices de desempenho e retornar o servidor-parceiro para a fila de disponíveis.

A taxa de desempenho ( $td$ ) é calculada pelo tamanho do processamento a ser feito considerando o tempo de execução. O tamanho do processamento ( $tp$ ) é a multiplicação do tamanho das matrizes, por exemplo, seja  $A[la,ca]$  e  $B[lb,cb]$ , logo o  $tp = la \cdot cb$ . Se tiver mais de duas tabelas, realize este cálculo de forma recorrente. Dado o  $tp$ , e seja o tempo de execução denotado por  $t$ , então a  $td = tm/tp$ . Portanto, quanto menor for o  $tp$  maior será o  $td$ , logo a fila de servidores-parceiros será ordenada de forma decrescente pela  $td$ .

**Aspectos de Simulação.** Utilize-se do *Docker* para simular as máquinas clientes, o servidor e os servidores-parceiros. Os clientes devem ter menos CPU e RAM do que os servidores a fim de garantir menos poder computacional. O *Docker* pode ser substituído por VM ou máquinas reais.

**Análise dos Resultados.** Por fim, mostre um gráfico com o tempo de execução de 30 multiplicações de duas matrizes de  $[n,n]$ . O gráfico deve mostrar os tempos do cliente (*standalone*) e através do servidor. As matrizes devem ser as mesmas nos dois tipos de execução. No seu relatório, justifique a diferença de tempo entre os dois tipos. Existe um valor de  $n$  que irá favorecer algum dos tipos?

## Visão Geral da Arquitetura



**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 3 alunos.
  - a. O NOME dos alunos deve constar no relatório!!!!
2. Envio somente pelo SIGAA.
3. Data de entrega até 13/05 às 23:55.
4. 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
5. 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.
6. Os trabalhos deverão ser apresentados, este requisito pode ser dispensado pelo professor em qualquer momento para todos os grupos de forma indiscriminada.