

Avaliação substitutiva - Programação Concorrente e Distribuída

Introdução

A avaliação substitutiva da disciplina consiste em um exercício de comparação experimental de performances entre versões de um mesmo algoritmo, variando o número de *threads* utilizadas para executá-lo.

O algoritmo consiste na realização de requisições HTTP para coleta e processamento de dados climáticos das 27 capitais brasileiras durante janeiro deste ano (2024). A partir dos dados retornados, compostos do registro de hora em hora de temperaturas, deve ser feito o cálculo das temperaturas média, mínima e máxima por dia para cada cidade em janeiro. Ou seja, com o dado coletado, cada cidade deverá conter, ao fim do experimento, 93 medidas (31 dias x 3 medidas).

Para coleta dos dados, deve ser utilizada uma API gratuita que provê os dados (<https://open-meteo.com/en/docs/>) de temperatura. **A API possui um limite de 10000 requisições por dia por usuário, então cuidado ao realizar testes.**

As requisições recebem como parâmetro a latitude e longitude do local consultado. Assim, será enviado junto a esse PDF um outro arquivo com essas duas informações para as 27 capitais.

Experimento

Quatro versões do experimento devem ser criadas e executadas, tomando como base:

- Uma versão de referência, sem uso de *threads* (ou, mais precisamente, apenas com a *thread* principal da main);
- Uma versão com 3 *threads*, sendo cada uma responsável pela requisição de 9 capitais;
- Uma versão com 9 *threads*, sendo cada uma responsável pela requisição de 3 capitais;
- Uma versão com 27 *threads*, sendo cada uma responsável pela requisição de 1 capital.

Cada versão do experimento deve consistir na execução de 10 repetições/rodadas de um algoritmo composto por:

1. 27 requisições HTTP (uma requisição por capital);
2. Processamento do dado recebido, calculando as temperaturas média, mínima e máxima por capital por dia;
3. Armazenamento dos dados gerados em memória.
4. Exibição dos dados gerados no console.

O objetivo é computar o tempo de execução de cada rodada e, ao fim das 10 rodadas, calcular o tempo médio para de execução (soma dos tempos de cada rodada dividido por 10).

Nas versões com *threads*, cada execução deve ter o tempo final computado **após o fim de todas as threads secundárias existentes** (exemplo de como contabilizar o tempo [aqui](#)). **Atente-se para isso: apenas após o fim das threads criadas explicitamente no código deve ser gerado o tempo de finalização do experimento.**

Entregas

A entrega do trabalho consistirá em 2 artefatos: um projeto em Java 17 com o código dos experimentos em Java e um relatório com os resultados obtidos.

Projeto Java

O projeto deve conter os códigos das quatro versões dos experimentos, responsáveis por realizar as requisições com diferentes números de *threads*, conforme explicado acima, e utilizar os dados obtidos para computar as temperaturas média, mínima e máxima de cada capital para cada dia de janeiro.

Relatório

O relatório deve ser feito no formato A4, com espaçamento 1,5 linhas e fonte Times New Roman 12, e deve conter no mínimo as seguintes informações:

- Explicação teórica, com no mínimo uma página e uma citação de referência bibliográfica (artigo científico ou livro), tratando sobre os seguintes tópicos:
 - O que são *threads*;
 - Como *threads* funcionam computacionalmente;
 - Como o uso de *threads* pode afetar o tempo de execução de um algoritmo;
 - Qual a relação entre os modelos de computação concorrente e paralelo e a performance dos algoritmos.
- Exibição e explicação dos resultados obtidos, com no mínimo uma página, comparando as 4 versões do experimento e comentando sobre o que foi obtido. Aqui, deve haver um gráfico mostrando a comparação das 4 versões, relacionando o tempo médio de execução e o número de *threads* usadas.
- Listagem das referências bibliográficas utilizadas, ao fim.

Observações

- O trabalho deve ser feito individualmente, mas qualquer tipo de plágio/cola será penalizado (o projeto receberá nota 0 (SEM EXCEÇÕES!!!!!!!!!!!!!!!!!!!!)).
- Projetos com erro de sintaxe ou que não possam ser executados irão receber nota 0 também
- A entrega será dia 30/06, até às 23:59, pelo AVA.
- O envio deve conter apenas um link para o repositório do GitHub (que deve estar público) com o código do projeto e o relatório NO FORMATO PDF.
 - NÃO ENVIEM UM ZIP OU COLOQUEM O PROJETO ZIPADO NO GITHUB!

Critérios de avaliação

Código (5 pontos)

- **Criação das 4 versões do experimento:** 1.5
 - Implementação das 4 versões conforme requisitos apresentados acima;
- **Realização das requisições e tratamento das respostas:** 1.5
 - Implementação e corretude do código que realiza as requisições e processa as respostas
- **Estruturação e organização do código:** 0.5;
 - Divisão do projeto em classes;
 - Modularização do código, utilizando métodos sempre que possível;
 - Organização/clareza do código (nomes significativos de variáveis, indentação, etc.);
- **Apresentação dos dados obtidos e gerados:** 1.0
 - Apresentação correta e organizada dos dados no console.
- **Organização do GitHub (README, comentários e tamanho dos commits, etc.)** (0.5);
 - README: como executar o projeto, no que ele consiste, etc.

Relatório/documentação (5 pontos)

- **Explicação teórica:** 2.5
 - Explicação correta e contextualizada dos conceitos pedidos;
 - Citação válida e coerente do referencial teórico e listagem das referências utilizadas ao fim do relatório;
- **Resultados obtidos:** 2.0
 - Explicação dos resultados obtidos, alinhado com a explicação teórica;
 - Utilização e apresentação correta do gráfico.
- **Clareza, organização e uso formal do Português:** 0.5
 - Uso correto da variedade formal da Língua Portuguesa;
 - Uso correto da formatação pedida;
 - Organização das informações.

Material de apoio

Java

- Do a Simple HTTP Request in Java. Disponível em: <https://www.baeldung.com/java-http-request>
- How to send HTTP request in Java?. Disponível em: <https://stackoverflow.com/questions/1359689/how-to-send-http-request-in-java>
- Trabalhando com JSON em Java: o pacote org.json. Disponível em: <https://www.devmedia.com.br/trabalhando-com-json-em-java-o-pacote-org-json/25480>

API (Open-meteo)

- Documentação e exemplo de requisição:

https://open-meteo.com/en/docs/historical-forecast-api#latitude=52.52&longitude=13.41&start_date=2024-01-01&end_date=2024-01-31&hourly=temperature_2m

Ferramentas de busca de referências

- <https://pergamum.ucb.br/>
- <https://arxiv.org/>
- <https://scholar.google.com.br/?hl=pt>