

Computação Escalável

Trabalho AS

Thiago Pinheiro de Araújo

EMAp/FGV

2022.1

Escrever um simulador de cenários explorando **paralelismo e concorrência**, e a partir dele criar um **pipeline escalável** capaz de gerar análises sobre os dados.

Objetivos

- Os objetivos do trabalho são:
 - Criar um programa capaz de **simular o tráfego de veículos** em uma cidade explorando as características de paralelismo e concorrência abordadas na disciplina.
 - Criar um pipeline capaz de **gerar análises** a partir de múltiplos cenários de simulação de tráfego
 - Modelar a arquitetura do pipeline
 - Escolhendo os mecanismos de comunicação e processamento paralelo adequados.
 - Extrair os dados de forma eficiente
 - Buscando reduzir o impacto no desempenho do sistema-alvo.
 - Processar os dados de forma paralela gerando as análises.
 - Prover um mecanismo para visualizar as análises.

Requisitos do simulador

- O simulador de tráfego deverá representar as ruas e os cruzamentos de uma cidade
 - Ruas devem ser representadas como arestas e cruzamentos como vértices de um grafo.
 - A cada execução do programa o grafo deverá ser gerado de forma aleatória durante a inicialização.
- O grafo deverá possuir vértices terminais que conectam a cidade simulada com cidades vizinhas
 - Cada vértice terminal deverá adicionar um número aleatório de veículos a cada ciclo.
 - Cada veículo que atingir um vértice terminal deverá ser removido da simulação.

Requisitos do simulador

- Ruas poderão ser trafegadas em ambos os sentidos
 - No entanto ao entrar em uma rua o veículo não pode trocar o sentido.
- Cada rua deverá possuir uma capacidade máxima veículos para cada sentido.
- O comprimento de uma rua deverá ser definido de forma aleatória durante a inicialização
 - O número de ciclos para atravessar uma rua deverá ser baseado na velocidade do veículo e no comprimento da rua.
 - Ao passar por um cruzamento a velocidade do veículo deverá variar de forma aleatória (considere limites máximo e mínimo para a variação).

Requisitos do simulador

- Cada cruzamento deverá ser controlado por um conjunto de semáforos de trânsito
 - Ao atingir um cruzamento o veículo deverá escolher de forma aleatória dentre as arestas associadas ao mesmo, exceto a aresta de entrada.
- O veículo somente poderá trafegar entre duas arestas caso:
 - A capacidade máxima da aresta de destino não tenha sido atingida.
 - Tenha obtido acesso exclusivo ao par de arestas (problema dos filósofos jantando).
- O programa deverá fornecer um mecanismo para visualizar o estado de cada ciclo ao longo da simulação.

Requisitos do pipeline de análises

- A **fonte de dados** a ser utilizada no pipeline consistirá em múltiplas instâncias da aplicação de simulação.
- A aplicação deverá ser instrumentada de forma que notifique o seu estado a cada ciclo da simulação.
- O simulador deverá executar cada cenário por um tempo definido de forma aleatória
 - Ao concluir a execução de um cenário o programa deverá gerar um novo cenário.

Requisitos

- O conjunto de análises deverá apresentar as seguintes informações considerando todos os cenários de simulação:
 - Número total de cenários em execução.
 - Número total de veículos nos cenários em execução.
 - Número total de ruas e cruzamentos nos cenários em execução.
 - A velocidade média dentre todos os veículos de todos os cenários executados.
 - A maior distância percorrida por um veículo considerando todos os cenários executados.
 - O tempo médio de execução dos cenários.
 - O cenário que trafegou o maior número de veículos e a quantidade.

Requisitos

- O usuário da ferramenta de análise deverá ser capaz de escolher um cenário para visualizar as seguintes análises sobre o mesmo:
 - Número de veículos.
 - Número de ruas e cruzamentos.
 - A velocidade média dentre todos os veículos.
 - A maior distância percorrida por um veículo.
 - Duração da execução corrente.
 - Número de veículos em cada rua.

Observações

- O objetivo do trabalho **não é** abordar:
 - Técnicas, ferramentas e algoritmos de simulação.
 - Técnicas de IA para tomada de decisão.
 - Algoritmos de escolha de caminho (*pathfinding*).
 - Técnicas e ferramentas de visualização de dados.
 - Modelagem dimensional.

Implementação do simulador

- O programa de simulação de cenários poderá ser escrito em C++ (utilizando threads) ou Python (utilizando o pacote *multiprocessing*).
- Os veículos devem ser divididos em N (configurável) linhas de execução independentes a fim de explorar o paralelismo e os mecanismos de controle de concorrência ao computar as suas ações.
- A atualização do ambiente também deve ser computada em uma linha de execução diferente da principal.
- A saída do programa de simulação deverá exibir uma representação do grafo contendo as ruas com o número de veículos em cada sentido, e os cruzamentos com o estado de cada semáforo.

Implementação do pipeline

- A extração do dado deverá ser realizada utilizando um dos mecanismos de comunicação abordados na disciplina
 - O grupo deverá definir a estratégia de ingestão do dado, o momento em que a mesma deverá ser executada, e o conteúdo da mensagem.
- Os dados extraídos deverão ser armazenados em um repositório de dados
 - O grupo deverá escolher o banco de dados e definir o seu modelo.
- Os dados devem ser processados de forma paralela utilizando um dos mecanismos abordados na disciplina.
- As análises devem ser apresentadas em tempo real através de uma aplicação linha de comando.

Entrega

- Grupos devem ter até 4 integrantes.
- O trabalho deverá ser entregue em um arquivo zip contendo
 - Um documento descrevendo a modelagem geral, as principais decisões de projeto e a solução para cada um dos problemas de concorrência identificados.
 - Um manual (readme) com instruções para compilar e executar.
 - O código-fonte de cada componente do sistema.
- A entrega deverá ser realizada no eClass até o dia 04/07/2022 às 23:59
 - Trabalhos entregues após esta data não serão aceitos.

- A avaliação será baseada nos critérios a seguir:
 - Documentação do projeto contendo a modelagem, os problemas identificados e as soluções.
 - Modelagem geral do sistema.
 - Solução para computar o comportamento dos veículos de forma paralela e concorrente.
 - Solução para veículos atravessarem cruzamentos.
 - Solução para computar as análises no pipeline.
 - Organização do projeto.
 - Qualidade do código.
 - Manual de instruções.