Programação Concorrente e Distribuída

CLB Python

Por que fazer o projeto em python? *Isso é uma demanda do professor, mas além disso, python tem alguns pontos fortes para esse projeto: É a linguagem que utilizamos por todo o semestre para aprender o conteúdo, tem bibliotecas fundamentais para acelerar o desenvolvimento do projeto e é simples de ser utilizada.*	
Quais recursos a linguagem python tem implementado para a tarefa? *As bibliotecas são o principal ponto. Bibliotecas interessantes a serem usadas são a multiprocessing, threads, pandas, matplotlib, seaborn, etc. Outros recursos que vão ser úteis é a própria linguagem. Por se tratar de uma linguagem de alto nível multipropósito é possível criar qualquer tipo de programa nessa linguagem.*	*https://docs.python.org/3/library/multiprocessing.html* *https://docs.python.org/3/library/threading.html#module-threading*
Quais são os pontos fortes e fracos de fazer o projeto em python? *O ponto fraco de python que mais me preocupa é a lerdeza de processamento da linguagem. Por ser uma linguagem interpretada esse efeito já é grave, mas python precisa de muito tempo para se comunicar com a máquina, o que só piora o processo. Além disso, o intenso uso de memória em python é preocupante também. Os pontos fortes incluem o fato de ser uma linguagem fácil de se desenvolver e sua extensa biblioteca de módulos e comunidade.*	*https://www.geeksforgeeks.org/disadvantages-of-python/* *https://www.w3schools.com/python/python_intro.asp*
Como programar de forma eficiente em python? *Utilizar os recursos da linguagem de forma inteligente é a melhor forma de atingir esse objetivo. É por isso que devs podem se especializar em desenvolvimento exclusivo em python, pois existem truques que só pessoas imersas nessa linguagem vão conhecer.*	*https://medium.com/@yuxuzi/comprehensive-guide-on-how-to-write-efficient-python-code-8c4b78a25047*

Qual a melhor forma de garantir a performance do projeto? *Eu quero que o projeto como um todo seja eficiente, puxando o python ao limite do que é possível em termos de performance. Para isso, desde as bibliotecas até a definição de funções precisa ser cuidadosamente pensada para máxima performance, cada linha deve ser eficiente em funcionamento, manter o código limpo e curto é uma questão essencial por conta da forma que o python roda código (1 linha de cada vez). Depois que o código estiver concluído, testes extensos devem ser rodados para encontrar pontos de melhoria.*	
Como abordar os pontos críticos de performance da maneira mais eficiente? *Os pontos críticos do programa devem rodar em paralelo da melhor forma possível. Operações como extração, transformação e análise de dados são computacionalmente custosas e elas são o foco do programa. São nessas operações onde eu preciso colocar meus esforços para otimizar o projeto.*	
Como identificar atrasos e os melhorar no projeto? *Testes diversos e análise cuidadosa vão mostrar os pontos de melhoria do projeto. Testes unitários, de integração e regressão vão ser precisos para esse objetivo, utilizando bibliotecas como unittest ou pytest é essencial, além de checagem de vazamento de memória com o valgrindr.*	*https://realpython.com/python-testing/*
Como acessar e ler arquivos em python? *O python tem funções como open() para cuidar de operações com arquivos, bem como o pandas e suas funções read.(). A mais rápida precisa ser a escolhida.*	*https://www.geeksforgeeks.org/file-handling-python/* *https://medium.com/@sanyagubrani/data-manipulation-with-pandas-399213045b91*
Como funciona o processo ETL? *Está descrito do documento do projeto como sendo o processo de extração, transformação e carga de dados, nessa ordem. Para arquivos tão grandes quanto 1 giga esse procedimento deve ser muito bem otimizado.*	

Como esse processo pode ser implementado de forma eficiente? *Práticas como se livrar de dados inúteis o mais rápido possível, incrementar os dados ao invés de os atualizar completamente e paralelizar as operações que os envolvam é a base da otimização do processo de ETL.*	*https://www.precisely.com/blog/big-data/etl-best-practices* *https://www.integrate.io/blog/7-tips-improve-etl-performance/*
Como funciona a leitura de um arquivo csv? *O pandas tem várias operações úteis para ler e processar arquivos csv extremamente grandes de forma eficiente. Por exemplo o parâmetro chunksize faz com que o programa apenas leia uma quantidade determinada de linhas de uma vez antes de continuar o processo de leitura, ou o parâmetro usecols, que realiza a leitura em colunas especificas definidas pelo programador*	*https://www.datacamp.com/tutorial/pandas-read-csv*
O quão importante é gerenciar a memória do computador? *Memória pode ser um limitador muito grande na execução do programa, pois é isso que vai limitar a quantidade de dados que podem ser lidos de uma vez. Otimizar esse processo da melhor forma é essencial.*	
Quais são as bibliotecas que vão ser usadas? *Pandas é praticamente certo de ser usado por sua alta performance e facilidade de uso. Uma biblioteca de visualização de dados é essencial para mostrar os resultados dos tribunais, algo como matplotlib ou seaborn, a depender da performance e facilidade de uso dessas bibliotecas. Por fim a biblioteca de paralelismo, threading e multiprocessing são duas ótimas bibliotecas, mas somente a biblioteca que ofereça maior speedup pode ser escolhida.*	

Quais são os pontos fortes e fracos das bibliotecas a serem utilizadas? *Pandas é construído em cima de C, por isso é rápida, além disso essa biblioteca foi construída para ser fácil de ser usada, com funções muito úteis que facilitam o desenvolvimento. A biblioteca para a visualização de dados é uma escolha menos importante, mas que pode ser um limitador pela quantidade de dados a serem analisados. Matplotlib cria gráficos altamente customizáveis, porém estáticos, enquanto seaborn é mais complexo por permitir a construção de gráficos dinâmicos, é também mais integrado para funcionar com pandas. A biblioteca de paralelismo é realmente sobre a questão de qual oferece o maior speedup para o projeto. Threading roda o código em múltiplas threads, enquanto multiprocessing cria vários processos do código que então são gerenciados pelo sistema.*	*https://www.w3schools.com/python/pandas/default.asp* *https://www.datacamp.com/tutorial/seaborn-python-tutorial*
Qual a vantagem da threading sobre a multiprocessing? *Enquanto threading foi a biblioteca utilizada pelo professor para a matéria e é mais eficiente em operações de I/O, multiprocessing consegue criar um paralelismo verdadeiro ao criar vários processos do código.*	*https://medium.com/@arjunprakash027/threading-vs-multiprocessing-in-python-a-comprehensive-guide-cae3ce0ca6c1*
É preciso usar as fórmulas do documento do projeto?	
Como programar as fórmulas do documento do projeto?	
Como fazer a análise assintótica do projeto?	
Como calcular o speedup da paralelizarão do projeto?	
Fazer testes de mesa é valido para o projeto?	
O que define o melhor trabalho? *Eu penso que o projeto que tiver o melhor ganho e terminar a tarefa o mais rápido possível deve ser o melhor, mas pontos como a documentação e a legibilidade do código podem ser um fator importante a ser considerado, além é claro, da acurácia do programa ao analizar os dados.*	

CBL Paralelização

Como paralelizar o programa da melhor forma possível? *Tudo que não for inerentemente sequencial precisa ser paralelizado. Uma excessão a essa regra é que caso o ganho por paralelizar uma operação seja negativo ou muito baixo, essa operação não pode ser paralelizada em prol da performance total do projeto. Partindo de uma abordagem totalmente otimizada de programação, a paralelização também precisa ser otimizada.*	
O que a biblioteca threading pode fazer? *Essa biblioteca funciona chamando threads do processador para executar o paralelismo do programa e atribuindo uma função ou faixa de código para ser executada por aquela thread.*	
Existe outra biblioteca melhor para o projeto?	
Por que usar a biblioteca time com a threading?	
Paralelizar somente as funções é o melhor a ser feito? *Não, idealmente paralelizar todas as operações internas e externas que o projeto fazer é o ideal, desde que isso proporcione ganho. Para isso existem várias abordagens possíveis, uma delas sendo a paralelização por meio da implementação de vários compiladores para rodar o programa. Outras abordagens mais convencionais, se bem implementadas, também pode ser muito válido.*	*https://github.com/RishiRaj22/PythonParallelism* *https://rishiraj.me/articles/2024-04/python_subinterpreter_parallelism*
Existe uma forma de paralelizar massivamente o projeto? *Multiprocessing é melhor do que threading nesse sentido por criar vários processos do programa, esquivando das limitações inerentes ao GIL do python. Outras formas de paralelizar de forma eficiente o projeto é dividir bem o problema e as tarefas além de utilizar programação assíncrona.*	*https://www.reddit.com/r/Python/comments/1c6sdyj/achieve_true_parallelism_in_python_312/*
Como paralelizar tendo em mente a quantidade diferente de recursos de cada computador? *A biblioteca psutil pode oferecer informações muito boas sobre as informações do computador onde o código irá rodar para que o projeto sempre rode de forma consistente.*	
O que é paralelismo de dados?	

Resources

https://www.youtube.com/watch?v=ELTtiMnHqMo
https://www.sitepoint.com/python-multiprocessing-parallel-programming/

https://rishiraj.me/articles/2024-04/python_subinterpreter_parallelism