Sistemas de Gestão de dados 2021/2022

Syllabus

Nota: leia também as regras gerais de funcionamento e avaliação da cadeira em informação geral da cadeira do inforestudante.

Link par aulas online também está na informação geral da cadeira no inforestudante.

Contactos:
e-mail: pnf@dei.uc.pt
whatsup:910400254 (a qualquer hora!)

Avaliação:

A informação de avaliação da cadeira está em informação geral da cadeira no inforestudante, bem como outras regras gerais. Pode capturar o ecran da info geral para ter uma fotografia imutável.

Programa (tentativa)

Nota: o programa pode variar um pouco de acordo com os tempos reais das aulas e outros factores

W1: Modern data processing architectures
W2: SGBDs and Application benchmarking
W3: ETL
W4: Phys org & NoSQL
W5: NoSQL & Presto + graph algs
W6: Bigdata, Hadoop, MR, Spark
W7: Spark ML
W8: Stream processing
W9: Python EDA
W10:Python data process

Parte pratica da cadeira

Os alunos organizar-se-ão em grupos de 2, e a avaliação será do grupo e individual.

As aulas serão de execução de tutoriais e de duvidas, aulas de duvidas, ou parte de aulas de duvidas dependendo das necessidades dos alunos. Nas aulas de tutoriais experimenta-se uma tecnologia. É suposto todos os alunos fazerem ao mesmo tempo o tutorial, o professor escolhe aleatoriamente quem ira fazer X passos, podendo mudar a meio. Cada aluno vai registando os passos ou capturando écrans. No final da aula cada aluno gera um pdf e faz upload do que fez. Nota da avaliação da aula: esteve presente, fez, participou e trabalhou = 100%, não fez ou não esteve presente 0%.

Entrega de documento e defesa intermédia do projecto 2

Esta entrega deverá ser sensivelmente meio do semestre, durante a execução do projecto 2, a data está como sempre na submissão de trabalhos da cadeira e conta para avaliação (ver info geral da cadeira). O objetivo é garantir que cada grupo já saberá exactamente o que vai fazer, que dataset vai usar, que perguntas vai responder e já definiu os detalhes relevantes para o projecto 2. Dessa forma evitaremos que só peguem no projecto no final da cadeira, o que seria trágico. O template esta em: ApresentacaoPlaneamentoDetalhadoProjecto2EXEMPLO

Projectos

Datas de entrega: vamos tentar definir desde os primeiros dias de aulas, em submissão de trabalhos em inforestudante (esperemos que não nos obriguem outra vez a migrar mais coisas para o UC-student)

Fundamental
Start early, and you need to discuss your task with the teacher to better define its details. Always keep in touch with the teacher, DO NOT TRY TO DO THE WORK ONLY WHEN THE DEADLINE IS APPROACHING, YOU WILL FAIL AND GET MAD ABOUT IT. ALSO, DO THE WORK IN LAB CLASSES, YOU NEED TO UNDERSTAND THNGS, DO NOT TRY TO DO IT ON YOUR OWN ONLY AND LATE.

Proj 1: (TPC-H perf bench e desenho do processamento)

PODE ACEDER AO ENUNCIADO EM SUBMISSAO DE TRABALHOS, PROJECTO 1.
Nesta parte pretende-se gerar e carregar o benchmark TPC-H para dois motores de bases de dados (postgres e um segundo), avaliar velocidade a executar (avaliação de performance), analisar a questão, melhorar e comparar. O TPC-H devera ser carregado com 25 GB de dados. O segundo motor de base de dados será uma escolha de modo a vários motores alternativos serem experimentados por diferentes grupos:

Grupos 1 a 4: Postgres + mysql -> 25 GB em ambos, todas as queries e os pontos 6 a 8 aplicam-se a ambos os motores neste caso.

Restantes grupos: Postgres 25GB e 10 queries + procura um motor moderno alternativo, estuda o motor e carrega o TPC-H, mas dependendo do motor pode ter de decidir tamanho muito mais pequeno (por não ser possível experimentar 25 GB nesse motor), e basta experimentar 10 queries também.
Procure o motor a experimentar, pode ver com o professor tb (exemplos de alternativas: oracle, monetdb, mongodb, cassandra, hbase, hypertable, voltdb, outros). Pode começar também por procurar no Google por database engines ou algo do genero)

Serão os seguintes os itens a avaliar:

1. gráfico de load time (carregar sem chaves)
2. gráfico de query time (pesquisar sem chaves), bem identificadas as queries e seus tempos
3. keys times (tempo de criação de cada chave PK e FK, bem identificado cada caso)
4. query times (pesquisas com chaves)
5. descrever/perceber/explicar que optimizações seriam possíveis de acordo com o fabricante de cada um dos motores para as pesquisas sobre um dataset como o TPC-H correrem mais depressa, e como funcionam essas optimizações.

O seguinte aplica-se a postgres apenas, excepto para os grupos que fazem postgres + mysql, caso em que tb têm de aplicar ao mysql e comparar com os resultados em postgres:

6. Agora experimente correr 2, 5 e 10 pesquisas ao mesmo tempo no motor (essas pesquisas devem ser diferentes ou com valores diferentes), meça os tempos e faça um gráfico de evolução com o numero de pesquisas simultâneas.
7. Tente melhorar a performance usando técnica(s) do ponto 5. Obtenha resultados (para pelo menos 5 pesquisas) e verifique se/que melhorou (ou não)
8. faça três pesquisas que selecionem (1) menos de 5000 linhas filtrando na data, (2) um so produto filtrando no produto, (3) um só supplier filtrando no supplier. Agora tente criar índices para tornar cada uma destas pesquisas mais rápida. Conseguiu? Mostre o que fez e os resultados.

Relativamente aos pontos 1 a 5, devem ser feitas comparações entre os motores.

Proj2: escolha uma das seguintes hipóteses, conforme prefira o tema de analise de dados ou algoritmos de processamento de grafos

Proj2a: data analysis of big dataset searched by the group

Pretende-se que encontrem um ou preferivelmente um conjunto de datasets, se possivel grandes (e.g. >50000 linhas no total, e/ou vários datasets a relacionar de alguma forma) e de um assunto interessante para analise. A razão para serem vários datasets será descobrirem, através de interligação, aspectos interessantes, e evitarem analises já feitas por outros. Por exemplo, podemos comparar a meteorologia em Portugal com a meteorologia em Inglaterra e no resto da europa se obtivermos dados de estacoes meteorológicas cá e lá. Pretende-se que proponham uma analise completa do tema. Não poderão usar dados cuja análise já esteja feita e disponível (na web ou trabalho de alguém em cadeira do curso). Normalmente para estes trabalhos procura-se dados abertos na web. Actualmente existem muitos dados abertos na web, procurando bem (porque todos os anos aparecem mais datasets interessantes) encontram-se muitos exemplos bastante interessantes. A escolha de um, ou ate preferivelmente vários datasets, é importante porque condiciona tudo o que se consegue fazer a seguir. Se o dataset for algo demasiado elementar, o trabalho ficara muito limitado e a nota também.

Do ponto de vista da aprendizagem na cadeira (e de avaliação do trabalho), o objectivo será aplicarem ferramentas que aprendemos na cadeira para processarem e analisarem os dados. Isso significa que podem usar motores, tais como Spark, ou usar apenas python para a analise, normalmente é suposto fazerem as analises como aprendemos na cadeira, incluindo carregar os dados, analisa-los (Exploratory Data Analysis), agrupar de formas que faz sentido, correlacionar vários datasets e fazer merge, obter estatísticas e correlações dos dados para explicar coisas, analisar series temporais, aplicar algoritmos como clustering, classificação e/ou outros conforme o problema. Para alem disso, o objectivo e a avaliação focará também o que aprendemos de relevante com a analise, porque foi importante, o que contribui para o mundo?
Vou dar exemplos de temas, mas claro que poderá haver exemplos ainda muito mais interessantes:

Dados da qualidade de ar: quais os países com pior e com melhor qualidade de ar, etc?
Dados de poluição na Europa: quais os países mais poluídos? Etc.
Dados de meteorologia (existem estações por todo o mundo com dados disponíveis online)
Dados de redes sociais (existem datasets abertos com dados deste tipo para analise)
Dados do covid no mundo (worldometer, outros)
Dados de cancros no mundo
Dados de produção e consumo de electricidade.
Dados de tipos de produção de electricidade. Por exemplo, quão verde será a produção de que tipos de produção e como varia entre países?
Dados geográficos de googlemaps ou outro sistema geografico, tais como localização e população de cidades
NYC táxi cab dataset
Caracterização completa dos países o mundo em termos de produção, PIB, população, variados indicadores, bem como clustering e classificação destes
...
...

Como organizar a análise:

1. Escolha do tema que interesse e para o qual conseguirá dados – exige procura activa da vossa parte
2. O que pretendemos perceber sobre o problema? Encontre-se várias perguntas que podem estar relacionadas
3. Formulação das perguntas e sua escrita. Normalmente deve ter pelo menos umas 10 perguntas principais, e cada uma delas dará em média origem a uns três ou 4 gráficos e estudos.
4. Desenho da forma como analisará os dados, com escrita dos algoritmos
5. Analise dos dados, obtenção de gráficos e conclusões
6. Escrita de relatório completo.

Proj2b: complex processing of graph data

Procure um dataset que seja interessante analisar usando “graph algorithms” do neo4J e do Apache Spark. Pode ser um dataset georeferenciado, uma rede social ou algo completamente diferente mas para o qual seja interessante analisar grafos. Para esse dataset faça primeiro uma descrição e caracterização dos dados apoiada em estatísticas, mapas e gráficos (e.g. no dataset de táxis em Nova Iorque isso será por exemplo quantos serviços foram feitos em cada àrea de nova Iorque, temporalmente). Depois corra os algoritmos/código do livro da cadeira “Neo4J Graph Algorithms” sobre o dataset em Neo4J e em Spark, criando exemplos para cada algoritmo. Conclua coisas interessantes através da analise. Organize bem todas as suas analises num documento e faça acompanhar por dataset e código. Mais uma vez, as suas analises não podem ser copia de algo já feito por outros.

Exemplos de forma de pesquisa de dataset: no google por “graph datasets”, procure também por “nyc taxi dataset”

Algumas indicações extra para procura de datasets:

“europe air quality open data”

Alguns datasets de grafos, do livro da bibliografia (graph algs on neo4j):

Finding Datasets

Finding a graphy dataset that aligns with testing goals or hypotheses can be challeng- ing. In addition to reviewing research papers, consider exploring indexes for network datasets:

• The Stanford Network Analysis Project (SNAP) includes several datasets along with related papers and usage guides.

• The Colorado Index of Complex Networks (ICON) is a searchable index of research-quality network datasets from various domains of network science.

• The Koblenz Network Collection (KONECT) includes large network datasets of various types in order to perform research in network science.

Most datasets require some massaging to transform them into a more useful format.

NOTA: alternativa ao projecto 2, mas apenas mediante verificação detalhada com o professor, se após a analise em conjunto o professor achar que realmente seria interessante e que teria a dimensão:

Se encontrar alguma bibliografia que estude a utilização detalhada de algum motor de armazenamento e/ou processamento de dados que tenha um exemplo completo com um dataset interessante, ou que possa aplicar-se a um dataset que encontrem, e que seja de dimensão semelhante ao projecto 2, pode analisar com o professor de será uma possibilidade fazer a analise desse dataset seguindo as instruções da bibliografia como projecto 2. Isto só será possível/útil se for suficientemente grande e interessante. O resultado final será um relatório completo e código associado, que ficará disponível para alunos em edições futuras da cadeira.

Exemplo: será interessante explorar o livro da bibliografia spark and neo4J? So vendo...
Apache Spark e neo4j, começando pelos exemplos completos (últimos capítulos) e depois os exemplos de algoritmos (os restantes capítulos). O código devera gerar também os gráfos resultados das pesquisas e mostra-los usando a ferramenta que possa ser mais util. O grupo dará uma apresentação com os highlights do trabalho e disponibiliza para os colegas presentes e futuros um documento organizado com cada pergunta, o código e a resposta, com parte visual sempre que útil, juntamente com o código exacto actualizado para estes exemplos (o código já existe no livro, mas é útil actualizar e organizar correctamente o detalhe).

Exemplo: Explore other code book on complex processing for analysis of data using python, spark or other choices, or some complete use of some database or data processing engine for analysis of some dataset.