Sistemas de Gestão de dados 2021/2022

Syllabus

(english version in the end of this document)

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

Link para zoom também está na informação geral da cadeira no inforestudante (o zoom é usado nas aulas praticas para irmos fazendo os exercícios em conjunto)

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", "bigdata datasets", "open datasets", etc

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

• 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.

Syllabus English version

Note: also read the general rules of operation and evaluation of the chair in general of the inforestudante chair.

Link to zoom is also in the general information of the course on inforestudante (zoom is used in practical classes to do the exercises together)

Contacts:
e-mail: pnf@dei.uc.pt
whatsup:910400254 (Anytime!)
Assessment:

The chair evaluation information is in general chair information on inforestudante, as well as other general rules. You can capture the general info screen to have an unchanging picture.

Program (attempt)

Note: the program may vary slightly based on actual class times and other factors

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

Practical part of the subject (lab classes):

The students will be organized in groups of 2, and the evaluation will be of the group and individual.

The classes will be of tutorials and doubts, classes of doubts, or part of classes of doubts depending on the needs of the students. In tutorial classes, technology is experimented with. All students are supposed to do the tutorial at the same time, the teacher randomly chooses who will do X steps, changing in the middle. Each student will record the steps or capture screens. At the end of the class each student generates a pdf and uploads what they have done. Class evaluation grade: attended, attended, participated and worked = 100%, did not attend or was not present 0%.

Delivery of document and intermediate defense of the project 2

This delivery should be approximately halfway through the semester, during the execution of project 2, the date is as always in the submission of assignments for the course and counts for evaluation (see general info of the course). The objective is to ensure that each group already knows exactly what they are going to do, which dataset they are going to use, which questions they are going to answer and that they have already defined the relevant details for project 2. The template is at: PresentationDetailedPlanningProject2EXAMPLE

Delivery dates: defined from the first days of classes, in submission of works in inforestudante (hopefully we will not be forced to migrate more things to UC-student again)

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.

Project 1: (TPC-H perf bench and processing design)

In this part we intend to generate and load the TPC-H benchmark for two database engines (postgres and a second), evaluate the speed to be executed (performance evaluation), analyze the queries performances, improve and compare.

The TPC-H must be loaded with 25 GB of data. The second database engine will be a choice so that several alternative engines can be tried by different groups:

Groups 1 to 4: Postgres + mysql -> 25 GB in both all queries and points 6 to 8 apply to both engines in this case.

Other groups: Postgres 25GB and 10 queries + look for a modern alternative engine, study the engine and load the TPC-H, but depending on the engine you may have to decide on a much smaller size (because it is not possible to try 25GB on that engine), and just try 10 queries too.
Look for the engine to try, you can check with the teacher too (examples of alternatives: oracle, monetdb, mongodb, cassandra, hbase, hypertable, voltdb, others). You can also start by searching Google for database engines or something like that)

The following items will be evaluated:
1. load time graph (load without keys)
2. query time chart (search without keys), well-identified queries and their times
3. keys times (time of creation of each PK and FK key, well identified each case)
4. query times (keyed searches)
4.b. explain plan comprehension for slowest and fastest queries, try to explain why they take long/short
5. describe/understand/explain what optimizations would be possible according to the manufacturer of each of the engines for searches on a dataset such as TPC-H to run faster, and how these optimizations work.

The following applies to postgres only, except for groups that do postgres + mysql, in which case they also have to apply to mysql and compare with the results in postgres:
6. Now try running 2, 5 and 10 searches at the same time in the engine (these searches must be different or with different values), measure the times and make an evolution graph with the number of simultaneous searches.
7. Try to improve performance using technique(s) from point 5. Get results (for at least 5 searches) and check if/what improved (or not)
8. Make three searches that select (1) less than 5000 lines filtering on the date, (2) only one product filtering on the product, (3) only one supplier filtering on the supplier. Now try creating indexes to make each of these searches faster. It achieved? Show what you did and the results.

For points 1 to 5, comparisons must be made between engines.

Proj2: choose one of the following hypotheses, depending on whether you prefer the topic of data analysis or graph processing algorithms

Proj2a: data analysis of big dataset searched by the group
It is intended that you find one or preferably a set of datasets, if possible large (e.g. >50000 lines in total, and/or several datasets to be related in some way) and an interesting subject for analysis. The reason for having multiple datasets is to discover, through interconnection, interesting aspects, and avoid analysis that is already done by others.

For example, we can compare the weather in Portugal with the weather in England and the rest of Europe if we get data from weather stations here and there. It is intended that groups propose a complete analysis of the topic. They will not be able to use data whose analysis is already done and available (on the web or work of someone in the course chair). Normally, for these works, open data is searched on the web. There are currently a lot of open data on the web, search (because every year more interesting datasets appear) you will find many very interesting examples. Choosing one, or even preferably several, datasets is important because it impacts everything you can do next. If the dataset is something too elementary, the work will be very limited and so will the grade.

From the point of view of learning in the course (and evaluation of the work), the objective will be to apply tools that we learned in the course to process and analyze the data. This means groups can use engines such as Spark, or just use python for the analysis, normally they are supposed to do the analysis as we learned in the course, including loading the data, analyzing it (Exploratory Data Analysis), grouping it in ways that make sense , correlate multiple datasets and merge, get statistics and correlations from the data to explain things, analyze time series, apply algorithms like clustering, classification and/or others as per the problem (what you can use depends on the analysis being done). In addition, the objective and the evaluation will also focus on what we learned relevant from the analysis, why was it important, what does it contribute to the world?
I'll give examples of themes, but of course there could be even more interesting examples:

Air quality data: which countries have the worst and best air quality, etc?
Pollution data in Europe: which countries are most polluted? Etc.
Weather data (there are stations all over the world with data available online)
Social network data (there are open datasets with data of this type for analysis)
Covid data in the world (worldometer, others)
Cancer data around the world
Electricity production and consumption data.

Data on types of electricity production. For example, how green will be the production of what types of production and how does it vary between countries?
Geographic data from googlemaps or other geographic system, such as location and population of cities
NYC taxi cab dataset
Complete characterization of countries in the world in terms of production, GDP, population, various indicators, as well as clustering and classification of these
...
...

How to organize the analysis:

1. Choose the topic that interests you and for which you will obtain data – it requires an active search on your part
2. What do we want to understand about the problem? Find yourself several questions that may be related
3. Formulation of questions and their writing. Normally it should have at least 10 main questions, and each one of them will generate, on average, about three or 4 graphs and studies.
4. Design of the way you will analyze the data, with writing of the algorithms
5. Data analysis, obtaining graphs and conclusions
6. Complete report writing.

Proj2b: complex processing of graph data
Look for a dataset that is interesting to analyze using "graph algorithms" from neo4J and Apache Spark. It can be a georeferenced dataset, a social network or something completely different but for which it is interesting to analyze graphs. For this dataset, first make a description and characterization of the data supported by statistics, maps and graphs (e.g. in the dataset of taxis in New York this will be for example how many services were made in each area of New York, temporally). Then run the algorithms/code from the book "Neo4J Graph Algorithms" over the dataset in Neo4J and Spark, creating examples for each algorithm. Conclude interesting things through analysis. Organize all your analyzes well in a document and track them by dataset and code. Once again, your reviews cannot be a copy of something already done by others.

Examples of dataset search form: google for "graph datasets", also search for "nyc taxi dataset"

Some extra pointers for searching datasets:

Algumas indicações extra para procura de datasets:

"europe air quality open data", "bigdata datasets", "open datasets", etc

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

• 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.

NOTE: alternative to project 2, but only after detailed verification with the teacher, if after the joint analysis the teacher thinks that it would really be interesting and that it would have the following dimension:

If you find any bibliography that studies the detailed use of some storage and/or data processing engine that has a complete example with an interesting dataset, or that can apply to a dataset that you find, and that is similar in size to the project 2, you can analyze with the teacher and it will be a possibility to analyze this dataset following the instructions in the bibliography as project 2. This will only be possible/useful if it is sufficiently large and interesting. The end result will be a complete report and associated code, which will be available to students in future editions of the course.

Example: will it be interesting to explore the spark and neo4J bibliography book? I'm just selling...
Apache Spark and neo4j, starting with the complete examples (last chapters) and then the algorithm examples (the remaining chapters). The code should also generate the search results graphs and show them using the most useful tool. The group will give a presentation with the highlights of the work and make available to present and future colleagues an organized document with each question, the code and the answer, with a visual part whenever useful, together with the exact code updated for these examples (the code already exists in the book, but it is useful to update and organize the detail correctly).

Example: 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.