Project_1: Implementation and Experimental Evaluation of Basic DHTs

Professors: S. Sioutas, A. Komninos, G. Vonitsanos (Postdoc Researcher@CEID)

Goal: The major task is the implementation and experimental evaluation of a variety of DHTs in a programming language of your preference (we suggest Python, C++ or Java). You could use artificial synthetic-data sets or real-data sets to evaluate the performance of the following fundamental operations (Queries): Build, Insert key, Delete key, Update key, **Lookup (key), Node Join, Node Leave**.

You could also use a virtual or real distributed environment using threads and sockets. Dockers and Containers using k8s is also a realistic elastic distributed cloud-based environment suitable for the above implementations.

You can download real datasets from the following URLs:

Find Open Datasets and Machine Learning Projects | Kaggle

20 Free Datasets for Data Science Projects | Built In

https://freegisdata.rtwilson.com/

https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

DHTs (Chord && Pastry): Develop Chord and Pastry decentralized (p2p) infrastructures based on DHT methods for storing a set of **(key, value)** pairs, where **value** is a set of attributes related to a specific **key.** Evaluate the performance of the following basic operations: insert key, delete key, node join, node leave and lookup (or exact match) queries. Compare the number of hops each p2p protocol requires. Plot the performance comparison between Chord and Pastry for all the above operations.

You could use the Data Movies Dataset from Kaggle[1] with 14-dimensions (attributes). This dataset contains a cleaned and structured collection of movie metadata sourced from The Movie Database (TMDB), covering films released between 1900 and 2025. It includes over 946,000 movies with detailed information such as genres, production companies, budgets, revenues, popularity, ratings, and more. This dataset is ideal for data science, analytics, and machine learning projects related to the film industry — including trend analysis, box office prediction, and recommendation systems.

Dataset Columns Description

Column	**Description**
id	Unique movie identifier
title	Official movie title

[1] https://www.kaggle.com/datasets/mustafasayed1181/movies-metadata-cleaned-dataset-19002025

adult	Boolean flag indicating adult content
original_language	Original spoken language (ISO 639-1 code)
origin_country	List of production countries
release_date	Movie release date
genre_names	List of genres associated with the movie
production_company_names	Names of involved production companies
budget	Reported production budget (USD)
revenue	Worldwide gross revenue (USD)
runtime	Duration in minutes
popularity	Popularity score (as provided by TMDB)
vote_average	Average user rating

[1] https://www.kaggle.com/datasets/mustafasayed1181/movies-metadata-cleaned-dataset-19002025

vote_count	Number of votes received

The data in this dataset was collected and preprocessed using the TMDB API. All movie information is © TMDB — provided under their Terms of Use. This dataset is not endorsed or certified by TMDB. Users must comply with TMDB's attribution and API usage policies when using this data.

movies_dataset_cleaned(3 files)**:** It includes the main dataset file, documentation, and an exploratory analysis notebook.

- movies_dataset.csv → The main cleaned dataset (946K+ movie records).
- README.md → Full dataset description, column definitions, and usage notes.
- analysis_notebook.ipynb → Exploratory Data Analysis (EDA) and sample visualizations.

Each file is part of a unified effort to provide a well-structured, ready-to-use dataset for movie analytics, machine learning, and data visualization projects.

W*e would like to detect concurrently the popularities of the K-movies with the following titles: title-1, title-2,…, title-k, where k is a user defined parameter (f.e. K=10).*

For the basic distributed lookup query, consider as **key** the movie's **title** attribute. Having Located the correct peers, then you could use local indexes (f.e. B+-trees) for further searching and filtering. Obviously, we must equip each peer with the appropriate local indexing scheme.

Background Knowledge: Data Structures, Multi-Dimensional Data Structures, Algorithms and Complexity, Databases, Object Oriented Programming (C++, JAVA), Functional Programming (Python, Scala).

(***) Deliverables: Zip or Rar file with executable files. Deadline: ~ 1st WEEK of February, 2024.

[1] https://www.kaggle.com/datasets/mustafasayed1181/movies-metadata-cleaned-dataset-19002025