Big data analysis of data transfers in multi petabyte distributed storage system

Ayat Al Bahadli, Austin Peay State University – OMNI internship

FERMILAB-POSTER-22-089-STUDENT

Introduction

dCache is a distributed storage system providing location independent access to data. The data are stored across multiple data servers as complete files presented to end-user via a single rooted namespace.

dCache has built-in monitoring capabilities which provide an overview of the activity and performance of the installation's doors and pools. Information about each data transfer is stored in a database - billing database.

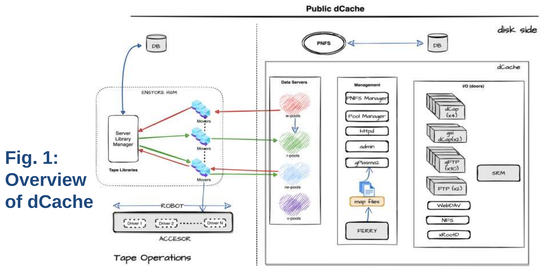

Fig. 1: Overview of dCache

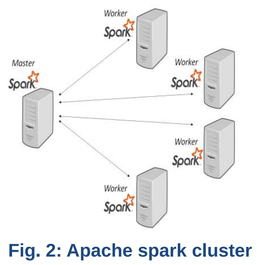

Fig. 2: Apache spark cluster

Purpose of Project

- The goal of this project is to setup and run distributed analysis of the data in the billing database to understand correlations in the access patterns and how the usage of the system develops over time to improve the efficiency of data access for scientists.
- Produce various storage performance plots to analyze tape access efficiency. Starting with simple things like how long it takes to stage data by VO and then move on to determining things like what is a probability of a co-located file to be pre-staged and identifying "popular files". Understanding of this information will help to implement more intelligent staging algorithms.
- We want to explore the data to discover correlations that can be used in the future for some predictions about system behavior based on its state at t0 to predict emerging patterns in the system.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Methods

- Processing data by directly querying the billing database is prohibitively slow due to huge volume of data stored there. We utilized data distribution and process parallelism provided by Apache Spark analysis engine.
- We used 16 former CMS worker nodes to setup Apache Spark Cluster and Jupyter Notebooks to run distributed analysis of the data in the billing database.
- Written code (python 3) to pipeline these data into Resilient Distributed Dataset (RDD).
- Produced various storage performance plots by converting the dataframes from spark to pandas and plot the data using matplotlib.

Results

- We were able to set up an Apache Spark cluster, setup a data pipeline to load the content of billing BD into RDD and we have been able to run a big data analysis on it. We had to overcome a few difficulties having to do with the way query parallelism is achieved when loading the data.
- We have observed that running a parallel query on Apache Spark once the data is loaded is significantly faster compared to direct access to DB making the data analysis practical.
- We have obtained the set of monitoring plots which we have set to achieve at this stage of the project.

Future work:

Analysis and understanding of the billing data using Apache Spark analysis engine is ongoing, We expect to have some understanding of the existing workflow patterns so that we can move on to utilizing data from Kafka stream where dCache also publishes the same billing data. The ultimate goal will be to be able to automatically modify dCache setup to adapt to emergent patterns based on already observed behavior.

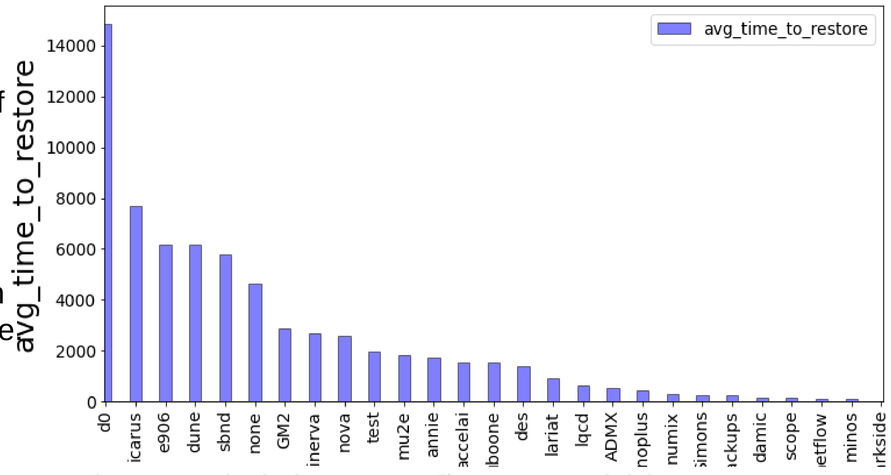

Fig. 3: Average time it takes to restore a file per VO over period since 2022-01-01

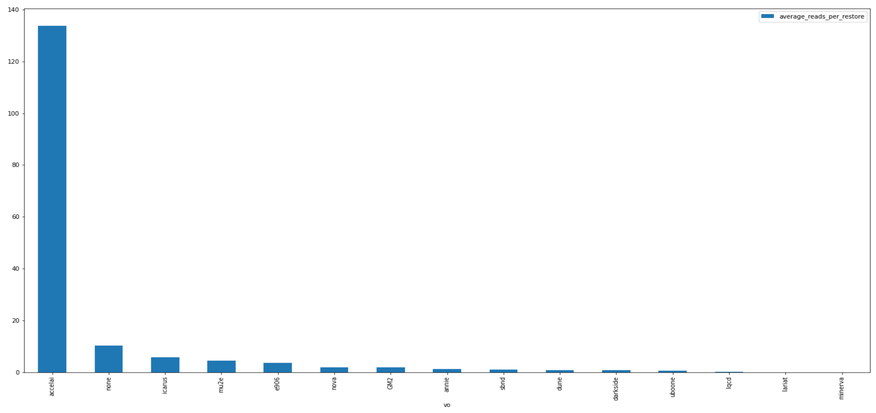

Fig. 4: Average number of reads per restore for each VO for the month of June 2022

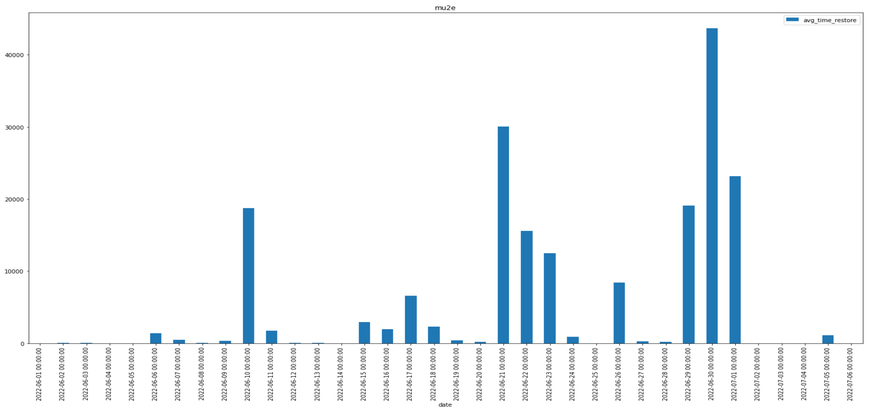

Fig. 5: Average time to restore vs date for mu2e VO over period since 2022-01-01