BERUFLICHE HOCHSCHULE HAMBURG

Grouping memory-exceeding amounts of CSV records

Comparing PowerShell and MySQL

Mauricé Ricardo Bärisch

Exposé

in course type Computer Science

Supervisor: Prof. Dr. Stefan Schiffner

Submission Date: June 30, 2024

Contents

1 Topic

Performing aggregation and visualization on large datasets has become a major part of Information Technology. Such datasets are often exported by a third party software or service provider, and arrive in a transmittable format like JSON or CSV. Grouping such records can be challenging when exceeding the available memory of the system.

We wrote a naive PowerShell script using the `Group-Object` command. It aimed to group the records coming from a CSV export, but ended up crashing the virtual environment it was running on, due to insufficient memory.

2 Main contributions

Based on this experience, we can define the purpose of this work:

- **Problem:** We need to aggregate CSV data bigger than the available memory
- **Objectives:**
  1. Find a solution that can group large CSV data
  2. Proof that the solution works by running an experiment with data bigger than the available memory
- **Questions:**
  1. Why does the `Group-Object` command not work on limited memory?
  2. What are requirements for a grouping algorithm in order to run on low memory?

As an alternative solution, we will consider MySQL, a Database management system.

3 Background

Databases, especially relational databases, are the gold standard for storing and querying large data. After E. F. Codd presented a mathematical model for relational databases in 1970 [1], manufactors adopted the principles and created SQL databases. Not only are they capable of grouping large data, but there are also a lot of papers and text books about them.

3.1 External Sorting

For example, [2] explains the external sorting strategies applied to large datasets. External sorting algorithms run in multiple so-called ”runs” where

parts of the data is being sorted and written to the disk (sorting phase). After that, the sorted parts are merged together in multiple iterations (merge phase). This way, large data can be sorted by efficiently making use of buffering and writing temporary results to storage.

3.2 Coma-Separated Values (CSV)

Specified in [3], CSV is a file format for storing a table in plain text. It has the following characteristics: 1. each row starts on a new line 2. the values for each column are separated by a delimiter (usually a comma) 3. there may be a header line those values specify the names of the columns

3.3 Docker

Docker is an open platform for containerizing applications, thus making them independent from the underlying infrastructure. We use Docker to build two comparable implementations (PowerShell and MySQL) that group large CSV data.

3.4 Container

As explained on Docker's website [4], a container isolates an application from our operating system and files. It bundles the application source code and required dependencies together, which makes the application very portable.

3.5 Aggregation

Aggregation is the process of summarizing records using numerical functions, e.g. calculate the total sum of a column. Aggregation also supports grouping the records beforehand, allowing us to summarize groups of records.

But personally, I feel like [5] explains it better:

> An extremely common feature of SQL queries is the use of aggregation and grouping. Aggregation allows numerical functions to be applied to entire columns, for example, to find the total salary of all employees in a company. Grouping allows such columns to be split according to a value of some attribute (...)

3.6 Relational Schema

[5] defines the relation schema based on three characteristics:

- **relation name**: the name of the table

- **relation attributes**: a list of uniquely identifiable attributes/columns
- **relation arity**: the number of attributes/columns

[6] defines the relation schema more formally as a database-wide mapping from relation names to either their respective arity (unnamed perspective) or their respective attributes (named perspective). Then, a row in a relation is either defined as a tuple of values $a = (a_1, \ldots, a_n)$ (unnamed perspective) or a set of pairs $a = \{(c_1, a_1), \ldots, (c_n, a_n)\}$, where each pair consists of the column name c_n and the value a_n assigned to it (named perspective).

The named perspective is closer to DBMS practice, but requires columns to have a unique name. We must ensure this in our exported CSV data to be able to make use of relational databases.

3.7 PowerShell

PowerShell is a scripting language coming with Windows. It is mainly used for automation and administration on Windows, but can also be used for general scripting tasks.

Modern PowerShell versions can be installed on other platforms as well. We will use PowerShell within a ubuntu docker container for our experiment.

4 Concept

We use the following methods to find a solution that groups large CSV data:

1. We started off with a **systematic literature research** to explain the bottleneck of our PowerShell implementation and find an alternative solution.
2. We build a **test**[1] to ensure that the new MySQL implementation does indeed aggregate the data in the same way our PowerShell implementation did.
3. We run an **experiment** where both solutions run in Docker containers limited in memory. We expect the MySQL implementation to run slowly but without crashes due to its external sorting strategy.

5 Planned Structure

5.1 Lists and Abstract

Abstract *Performing aggregation and visualization on large datasets has become a major part of Information Technology ...*

[1] For simplification, we just compare the aggregation result from both implementations on the same test data

List of Tables We are going to show some tables representing a CSV file or SQL relation.

Listings For reproducible results, we provide all the code for the experiment in listings or the appendix.

Glossary Because we use a lot of acronyms, e.g. ”CSV”, ”DBMS” ...

5.2 Introduction

Motivation Explaining the circumstances in q.beyond why we had to group large CSV data.

Main contributions See 2

5.3 Background

Using 3 and future research to explain the two different kinds of grouping (split into groups vs. aggregation on groups), pointing out that PowerShell's `Group-Object` command uses the first approach. Explaining how SQL databases like MySQL could help us solve the problem (by using external sorting strategies). Making sure that we can use relational databases for the data provided by the CSV export.

5.4 Experiment

Methods Explaining the methods used for this term paper (see 4).

Generating records Describing how we use an example CSV and bloat it up by repeatingly adding the same rows to it, ending up with a 400MB big CSV file we can use for our experiment.

Using Docker Explaining how and why docker is being used to create a reproducible experiment and limiting the memory for it.

PowerShell Implementation Going into detail about the PowerShell implementation using `Group-Object` to group the CSV records.

MySQL Implementation Going into detail about the MySQL implementation. Also providing a script for initializing the MySQL database with data corresponding to the contents of the CSV export.

Running the experiment Giving details about how to run the experiment.

5.5 Discussion

Evaluation Our experiment has a lot of quirks! We mention them here.

Findings Presenting our results after running the experiment a few times.

Future Work Given the motivation of this paper and the flaws of our experiment, we hint to possible future work.

5.6 Ending

Appendix Scripts Providing scripts not discussed in the term paper but required to run the experiment.

Bibliography

6 Schedule

1. Gather Topic Ideas (January 1 - January 7): Spending the first week brainstorming and gathering potential topics for the term paper.
2. Decide on a Topic (January 8 - January 14): Narrowing down the list of ideas and selecting a final topic. Getting approval by my trainer.
3. In-depth Research (January 15 - February 4): Conducting research, collecting relevant sources and information.
4. Learn LaTeX (February 5 - February 18): Learning the basics of LaTeX.
5. Learn Docker (February 19 - February 25): Spending time learning docker, focusing on how it can be used for the experiment.
6. Outline the Paper (February 26 - March 3): Creating a detailed outline for the term paper.
7. Implementing the Experiment (March 4 - April 7) Designing, implementing and running the experiment.
8. Writing the Paper (April 8 - April 28)
9. Proof-reading and Revisions (April 29 - May 12)
10. Finalize and Format the Document (May 13 - May 17)
11. Send in Finished Document (May 18)

References

[1] E. F. Codd, "A relational model of data for large shared data banks," *Commun. ACM*, vol. 13, no. 6, p. 377–387, jun 1970. [Online]. Available: https://doi.org/10.1145/362384.362685

[2] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*. Benjamin/Cummings, 1989.

[3] Y. Shafranovich, "Common Format and MIME Type for Comma-Separated Values (CSV) Files," RFC 4180, Oct. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4180

[4] Docker, "What is a container," Docker, accessed: 2023-05-17. [Online]. Available: https://docs.docker.com/guides/walkthroughs/what-is-a-container/

[5] M. Arenas, P. Barcel´o, L. Libkin, W. Martens, and A. Pieris, *Database Theory Querying Data*, preliminary ed. San Francisco, CA, USA: Santiago Paris Bayreuth Edinburgh, 2022.

[6] N. Schweikardt, T. Schwentick, and L. Segoufin, *Database theory: query languages*, 2nd ed. Chapman & Hall/CRC, 2010, p. 19.

[7] T. Halpin and T. Morgan, *Information Modeling and Relational Databases*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[8] J. Edgar. Algorithms for sql query operators. Simon Fraser University. [Online]. Available: https://www2.cs.sfu.ca/CourseCentral/454/johnwill/cmpt454_04queries.pdf

[9] Microsoft, "Group-object," Microsoft. [Online]. Available: https://learn.microsoft.com/de-de/powershell/module/microsoft.powershell.utility/group-object?view=powershell-7.4

[10] ——, "Groupinfo class," Microsoft. [Online]. Available: https://learn.microsoft.com/de-de/dotnet/api/microsoft.powershell.commands.groupinfo?view=powershellsdk-7.4.0

[11] ——, "Hashtable class," Microsoft. [Online]. Available: https://learn.microsoft.com/de-de/dotnet/api/system.collections.hashtable?view=net-8.0

[12] D. Microsoft, "microsoft-powershell - official image — docker hub," Docker, accessed: 2023-05-17. [Online]. Available: https://hub.docker.com/_/microsoft-powershell

[13] D. Canonical, "ubuntu/mysql - docker image — docker hub," Docker, accessed: 2023-05-17. [Online]. Available: https://hub.docker.com/r/ubuntu/mysql

[14] Docker, "Runtime options with memory, cpus, and gpus," Docker, accessed: 2023-05-17. [Online]. Available: https://docs.docker.com/config/containers/resource_constraints/#limit-a-containers-access-to-memory

[15] Microsoft, "about ref - powershell — microsoft learn," Microsoft, accessed: 2023-05-17. [Online]. Available: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_ref?view=powershell-7.4