BERUFLICHE
HOCHSCHULE
HAMBURG

Grouping 10,000 CSV records

comparing PowerShell and MySQL

Mauricé Ricardo Bärisch

Term Paper

in course type Computer Science

Supervisor: Prof. Dr. Stefan Schiffner

Submission Date: May 9, 2024

Disclaimer

I confirm that this thesis type is my own work and I have documented all sources and material used.

Munich, May 9, 2024 Author

Abstract

Performing aggregation and visualization on large datasets has become a major part in 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. This term paper compares the grouping operation on large CSV files between PowerShell's `Group-Object` command and MySQL's `GROUP BY` and `SORT BY` statements in regards of their interface, underlying algorithm and RAM usage. On two identical machines limited in RAM, we simulate how PowerShell's and MySQL's grouping operations perform. As a result, we formulate best practices for aggregating large CSV files.

Keywords: Database, Aggregation, Grouping, CSV

Contents

1 Introduction

Nowadays, many processes handling large amounts of data (not to confuse with Big Data [1]) already exist. Still, they must be maintained, verified and optimized to ensure both quality and efficiency.

1.1 Motivation

At q.beyond, we had to deal with just that: large monthly CSV exports by a third-party service provider that our process was operating on. For verification, we collected the exports from last year, grouped the records by a column and exported each group as an individual Excel file. Each file contained two sheets: one with an aggregated monthly overview and the other with all the individual records of that particular group. As we will explore in 2.2, this result actually requires two different kinds of grouping.

[TODO: Result Figure]

We ended up writing a PowerShell[2] Script, which, in our first draft, crashed our Azure Virtual Desktop (AVD) due to insufficient RAM.

1.2 Main contributions

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

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

As an alternative solution, we will consider MySQL, a Database Management System (DBMS).

[1] Data with high variety, volume and velocity. Cannot be processed by conventional data processing software.

[2] A scripting language coming with Windows. Used for administrative and automation tasks.

2 Background

2.1 Database

2.2 Grouping, Aggregation, Splitting

3 Simulation

3.1 Methods

3.2 Implementation

3.3 Results

4 Conclusion

4.1 Comparison

4.2 Future Work

5 General Addenda

If there are several additions you want to add, but they do not fit into the thesis itself, they belong here.

5.1 Detailed Addition

Even sections are possible, but usually only used for several elements in, e.g. tables, images, etc.

6 Figures

6.1 Example 1

6.2 Example 2

List of Figures

List of Tables

Glossary

AVD Azure Virtual Desktop. 1

DBMS Database Management System. 1

RAM Random Access Memory. The memory accessible for apps to store their variables, functions and other temporary data in. 1

References

ELMASRI, RAMEZ, und SHAM NAVATHE. 1994. *Fundamentals of database systems*. Menlo Park: Benjamin/Cummings Pub.