Project Proposal: Transformer - Fast Data Converter

Kareem Taha: tkareem@pdx.edu
11/2/26

I propose to opt out of the project theme to instead build a simple data pipeline. The purpose is for me to learn more about applying Rust parallelism to my work as a Data Engineer. I want to explore converting logs or data files.

If this proposal is rejected, my alternative proposal is an advanced cellular automata applier that works on images. It would take an image as input and apply rules on the pixel bits depending on which rule the user specified. Other simulation parameters include numuber of iterations, potentially custom rules.

1. Project Name

Transformer: A Quick Data Conversion Tool

---

2. Topic Area

Data Engineering / Systems Programming with Rust

The project falls under the broad area of data engineering, focusing on efficient file-format conversion, data transformation, and concurrent I/O in Rust.
It emphasizes practical systems-level programming for real-world data pipelines.

---

3. Project Vision

The goal of this project is to build a fast data converter written entirely in Rust. It will read structured data from large JSON/CSV files, optionally enrich or filter the records, and then efficiently serialize the results into Parquet format for use in analytics platforms such as Apache Spark, Polars, or DuckDB.

This project bridges data engineering and systems programming by addressing the performance, safety, and memory-efficiency limitations of Python-based tools while providing similar functionality with Rust's speed and reliability.

By implementing this tool in Rust, I aim to: - Demonstrate memory-safe, parallel data processing without garbage collection overhead. - Explore the Arrow2 and Parquet ecosystems in Rust. - Deliver a reusable, open-source command-line utility for real-world data workflows.

4 Core Functionality

The main components should include:

1. **File Parsing**
2. **Optional Enrichment and Filtering**
3. **Parquet Writing**
4. **Command-Line Interface (CLI)**
5. **Logging and Progress Tracking**

5 Stretch Goals

If time allows: - **Parallel CSV processing**.
- **Partitioned output** by date or key (e.g., write multiple Parquet files by day).
- **Benchmarking** against Python's Pandas workflow.
- **Dockerized build** for reproducibility.

6. Issues of Concern

My primary concern is dealing with the low-level details of file IO and data representation in Rust. Despite good performance the ownership rules make writing complex code sequences challenging. My approach will be to take things one step at a time.

---

6. Expected Outcome

In the end, I hope to have some type of tool that can extract value from large data files efficiently. Either to pipeline them into a data warehouse or perform some type of analysis.