

Project Plan: Option C - Workload-Aware Tuple Partitioning using Amazon Books Dataset

Step 1: Environment Setup

- Create and activate a conda environment.
- Install required libraries: pandas, numpy, pyarrow, matplotlib, networkx, scikit-learn, tqdm, pyyaml.

Step 2: Project Folder Structure

amazon-optionc/

data/

partitions/

workloads/

results/

figs/

src/

paper/

Step 3: Dataset Acquisition

- Use Amazon Book Ratings/Reviews dataset.
- Place raw dataset files in data/ directory.

Step 4: Preprocessing

- Select fields: user_id, book_id, rating, timestamp.
- Drop missing values and duplicates.
- Convert IDs to dense integer indices (user_idx, book_idx).
- Filter users/books with at least 5 interactions.
- Save processed dataset as data/interactions.parquet.

Step 5: Partitioning Strategies and Workload Generation

Partitioning Methods:

- 1) Round-Robin
- 2) Hash Partitioning (user_id)
- 3) Range Partitioning (book popularity quantiles)
- 4) Group-Based (Schism-like co-access clustering)

Workload Types:

W1: Lookup-heavy (fetch by user)

W2: Range-scan (rating/popularity ranges)

W3: Book bundles (co-access sets)

Save workload descriptions in workloads/ as JSON.

Step 6: Simulation & Metrics Logging

- For each strategy \times workload \times P partitions:
- Simulate query execution.
- Record latency_ms, partitions_touched, rows_scanned, rows_matched.
- Save logs to results/.

Step 7: Evaluation, Plots & Tables

- Compute averages per (strategy, workload, P).
- Plot:
- Latency per workload
- Partitions touched
- Load balance (Gini index)
- Save results/summary.csv and figures in figs/.

Step 8: Adaptive Controller Demonstration

- Implement a small decision module:
- If lookup-heavy \rightarrow hash strategy
- If range-heavy \rightarrow range strategy
- Else \rightarrow group strategy
- Run small test batch and report improvements.

Step 9: Paper Preparation

- Prepare diagrams and tables.
- Insert plots, results, and interpretation in IEEE format.

This document outlines the complete execution plan for implementing Option C.