SeLeP: Learning Based Semantic Prefetching for Exploratory Database Workloads

Farzaneh Zirak, Farhana Choudhury, and Renata Borovica-Gajic
School of Computing and Information Systems, The University of Melbourne

Prefetching for Exploratory workload

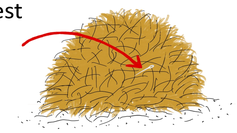
Data Exploration

Search enormous amount of data for interesting information

Not always sure what we are looking for (until we find it)

Data Exploratory tools need to be

- Interactive: additional delay can hurt the result
- Adaptive: Users' workloads and interest may shift



➢ Predict and cache likely query results to reduce query execution time (user wait time)

State-of-the-art Prefetchers

Work with Logical Block Addresses	Not scalable
Not adaptive	Not suitable for SQL-based workloads

➢ Semantic based prefetchers are proven to be more effective

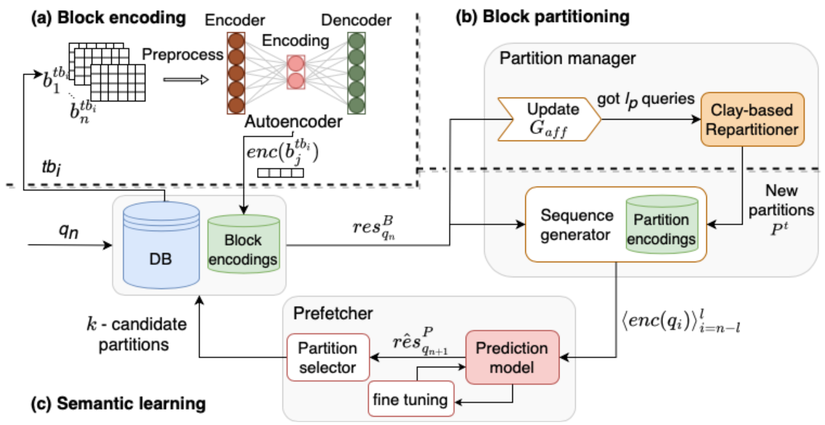
Objectives

Propose a prefetcher:

Semantic based	• Makes prefetch decision considering data values
Support SQL workload	• Is suitable for both navigational and SQL-based workloads
Enhance interactivity	• Predicts accurately and reduces the response time
Adaptive	• Can adapt to the workload shifts

SeLeP in Nutshell

System Overview

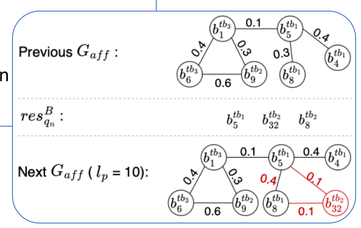


Extract semantics

- Each block is like a matrix of values
- Preprocess and normalize the blocks
- Embed blocks into 32 bit vectors using AutoEncoders

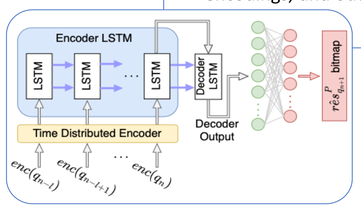
Partition the blocks

- Cluster blocks frequently accessed together in the same partitions → Graph partitioning
- Edges reflect co-accessing rate
- Aggregate block encodings to calculate partition encodings



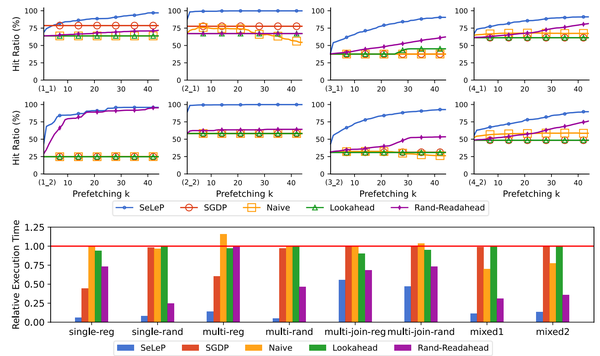
Semantic learning

- Represent queries with aggregation of encoding for partitions they access → Time-series forecasting
- Train the multi-layer LSTM model with sequence of query encodings, and output bitmap of subsequent partition access



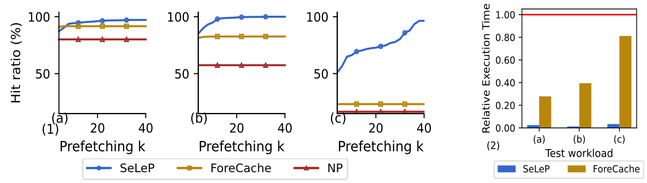
SeLeP in Action

Hit ratio and execution time in different SQL-based workloads



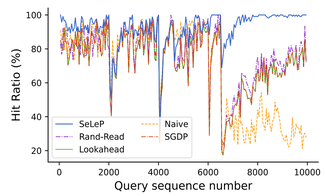
Setting: single table, multi table, multi table with join, and mixed SQL-based workloads from SDSS DR7

Hit ratio and execution time in different navigational workloads

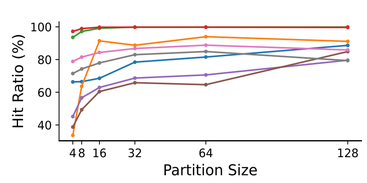


Setting: (a) smooth, (b) jumping, (c) random navigational workloads from SDSS DR7

Adaptivity test



Partitioning impact on hit rate



Summary

- Prefetching can substantially reduce I/O time
  - But the existing prefetchers cannot perform well in SQL-based and jumping navigational workloads
- SeLeP can benefit all types of exploratory workloads
  - improves the hit ratio up to 40% and reduces I/O time up to 45% compared to the state-of-the-art
  - attains 95% hit ratio and 80% I/O reduction on average