Semantic Cache Evaluation

([GitHub - open-infrastructure-labs/caerus-semantic-cache](https://github.com/open-infrastructure-labs/caerus-semantic-cache))

# Setup

* 3 Bare Metal Servers
* Storage
  + HDFS Namenode on first server
  + HDFS Datanode / server
  + 250GiB SSD / server
* Compute
  + Spark Master on first server
  + Spark Worker / server
  + 30 CPU Cores / server
  + 60 GiB memory / server
* Cache
  + HDFS – Disk tier used (64GiB used for Semantic Cache)

# GridPocket Trace

## Type of Queries

* Q1 (ShowMapCons): Compute the per meter aggregated consumption, allowing to display results either in a heatmap or a per state aggregated consumption.
* Q2 (ShowMapMeter): In order to display a cluster map, obtain each meter with its info (city, Id,...).
* Q3 (ShowMapHeatmonth): Get daily data for a given month for a (slider) parametric per day display.
* Q4 (Showgraphcons): Obtain the consumption of meters in Paris for a month.
* Q5 (ShowPiemonth): Obtain consumption for a specific subset of region consumption.
* Q6 (ShowGraphHCHP): Obtain data for drawing peak versus shallow hour consumption.
* Q7 (Showday): Get the data for displaying the consumption of any specified hour of a given month.

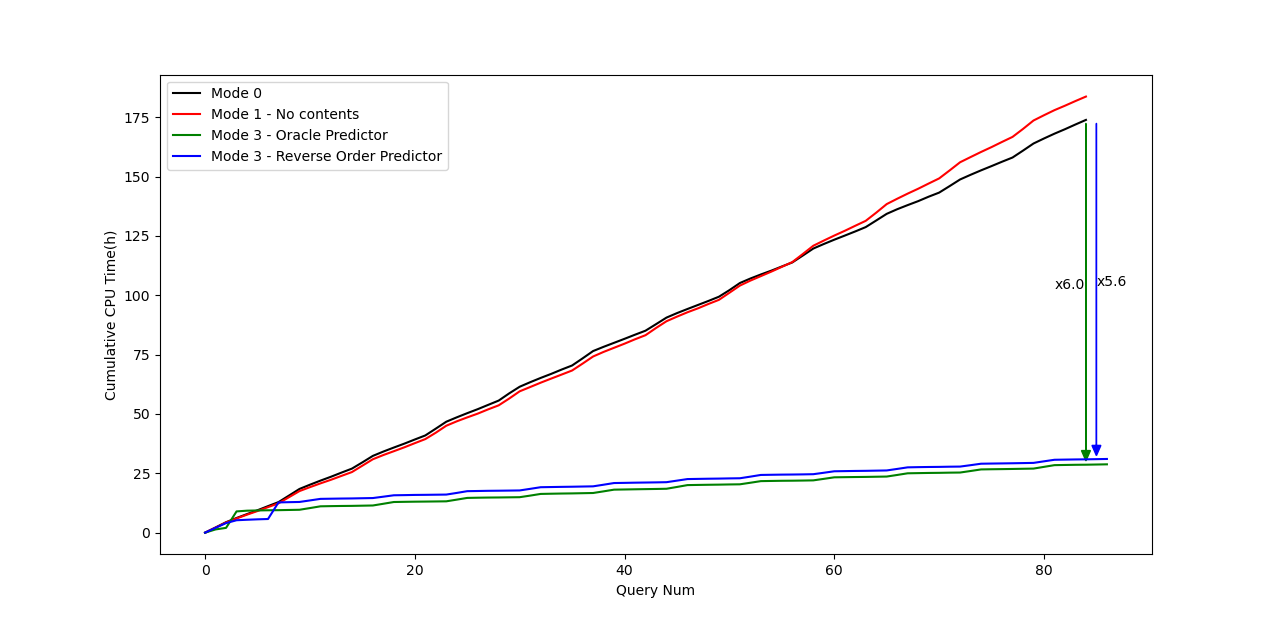
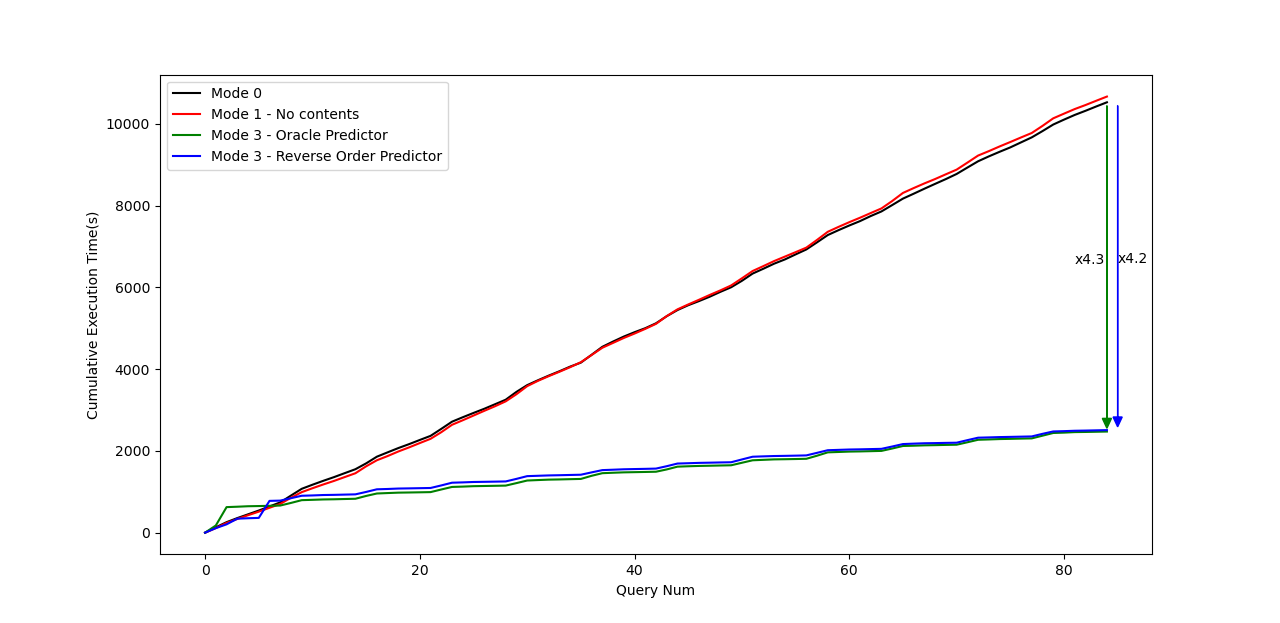
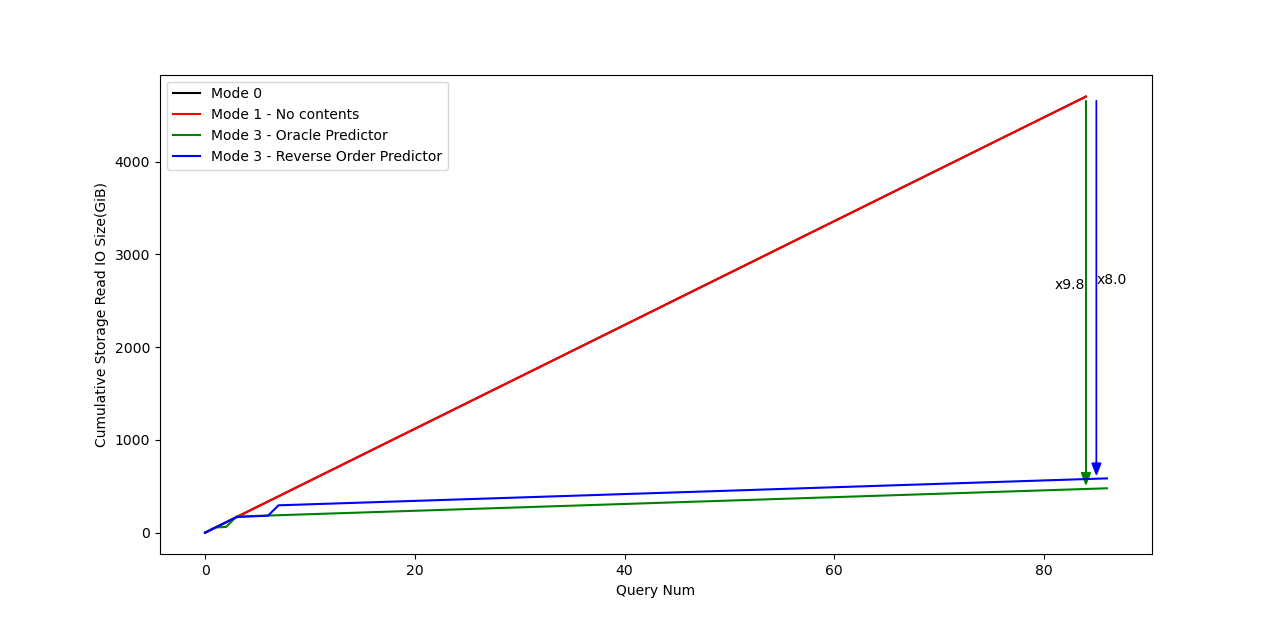
## Trace Characteristics

* 1 year worth of data (56 GiB CSV)
* Run each query type for each month ( total queries)

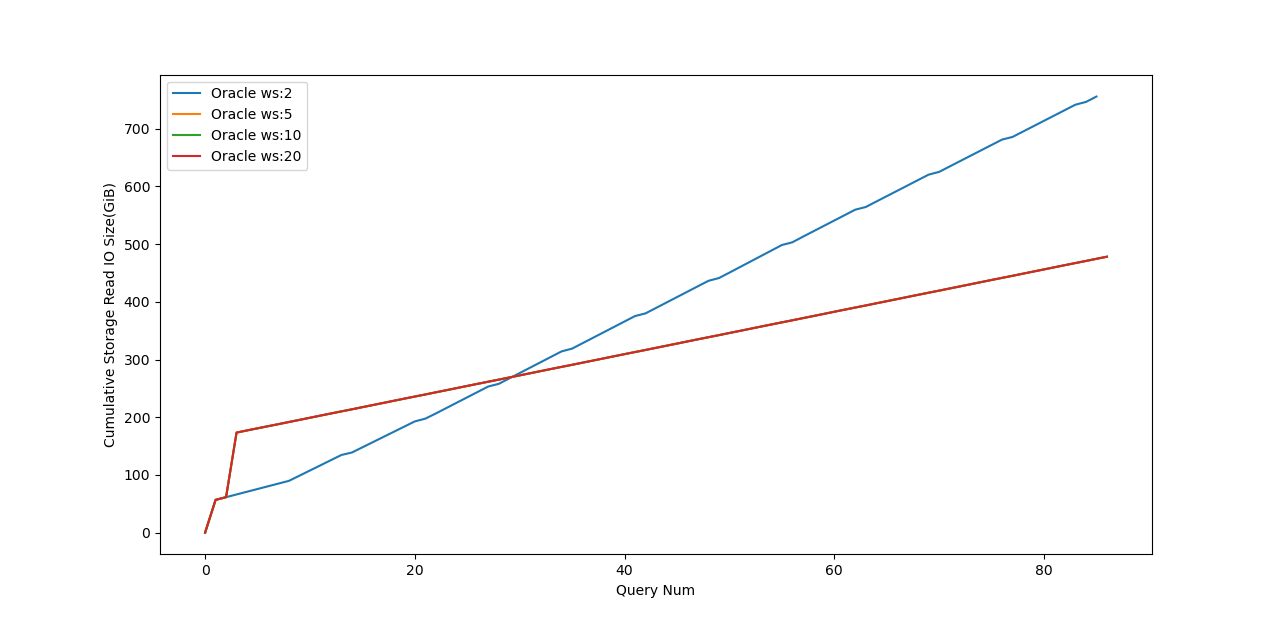
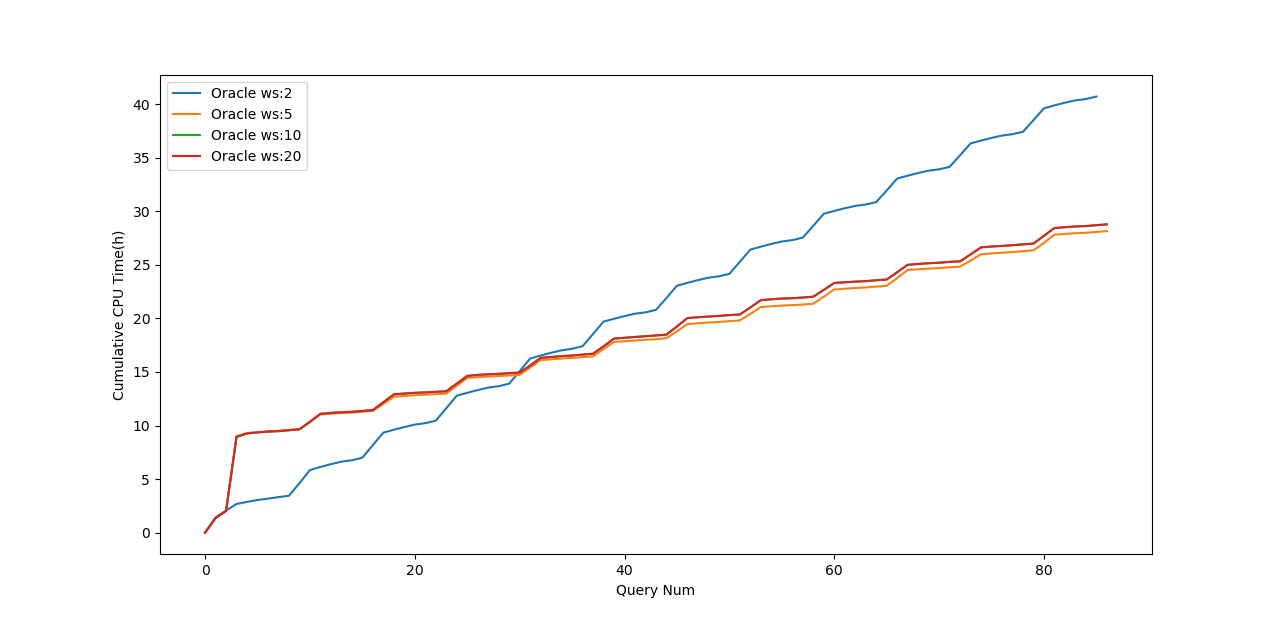
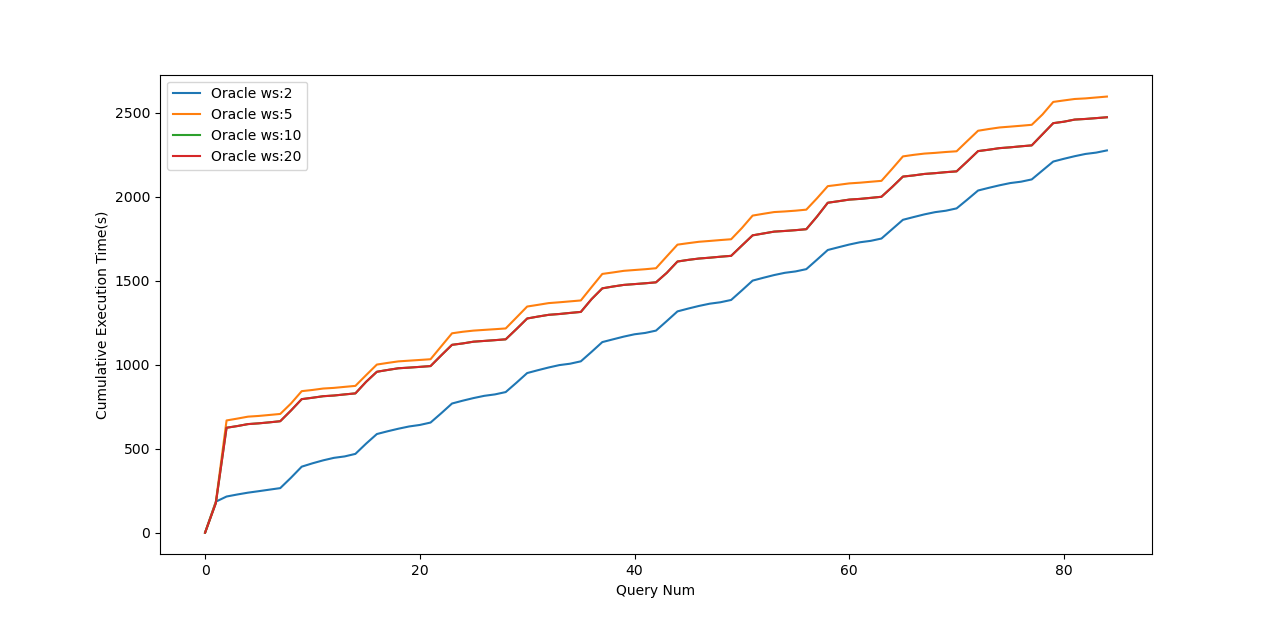
## Evaluation

### Methods Comparison

Compare four different methods (baseline Spark, Semantic Cache w/o contents, Semantic Cache with Oracle Predictor, Semantic Cache with Reverse Order Predictor).

### Window Size Comparison

Compare methods using different window size predictions with the Oracle Predictor.

### Optimization time overhead (in Spark Driver) when using different prediction window sizes.

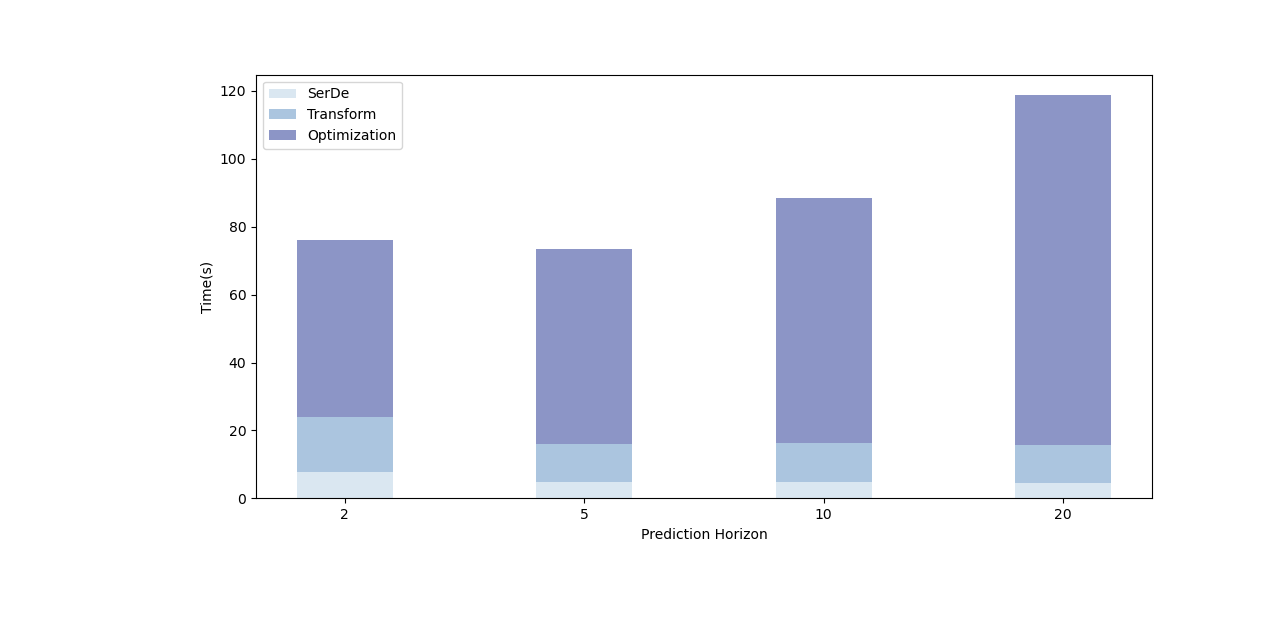


Figure 1: Oracle Predictor

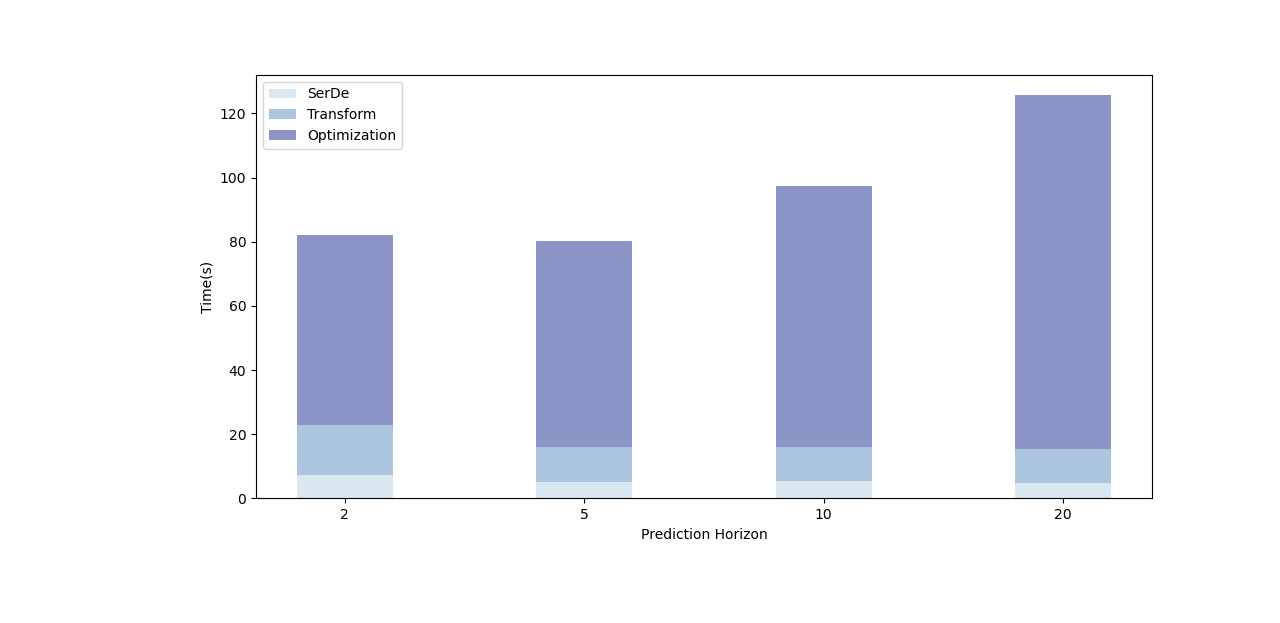


Figure 2: Reverse Order Predictor