Towards Practical Latency SLOs on Cloud Data Warehouses

Markos Markakis, Tim Kraska

MIT CSAIL

Key Desired Behavior	Plan resources based on history	Adjust resources based on demand	React to concurrent live load
Timescale	≈ 24 hrs	≈ 5 mins	≈ 1 s
ResTune	✓ *Replay-based knob tuning on replica*	✗	✗
WiseDB	✓ *Tree model for fixed per-template routing*	✗	✗
Das et al.	✗	✓ *Container sizing w/ token bucket*	✗
SLAOrches-trator	✗	✓ *Utilization- or simulation-based pool resizing*	✗
BRAD	✗	✓ *Blueprint beam search*	✗
Auto-WLM	✗	✓ *Queueing-based horizontal scaling*	✗
RAIS	✓ *Multi-forecast parameter tuning*	✓ *Queueing-based horizontal scaling*	✗

SLO-Aware Multi-Cluster Management Requires 3 Key Behaviors

Compute-storage separation unlocks an opportunity

- Modern data warehouses **decouple** compute and storage.
- They face **diverse workloads** with very different latency SLOs (e.g., dashboards vs. ETL)
- Naïve approach: use **dedicated cluster(s)** per workload.
- Must keep scaling/sizing each of them; still **no actual SLO-aware workload management**.

What if we enforce objectives for each query individually?

- Each **individual incoming query** has an SLO; inherited from somewhere, but it doesn't matter.
- We also have a **set of clusters**, of possibly varying sizes.
- **Three key behaviors** to effectively maintain SLOs:
  - **Plan:** Some workload patterns are predictable. Have clusters ready in anticipation.
  - **Adjust:** No forecast is perfect, because of errors or drift. Retain online scaling ability.
  - **React:** Per-cluster load varies. Keep SLOs and cost in mind when routing each query.
- In past work [1-7], **no system offers all 3** key behaviors.

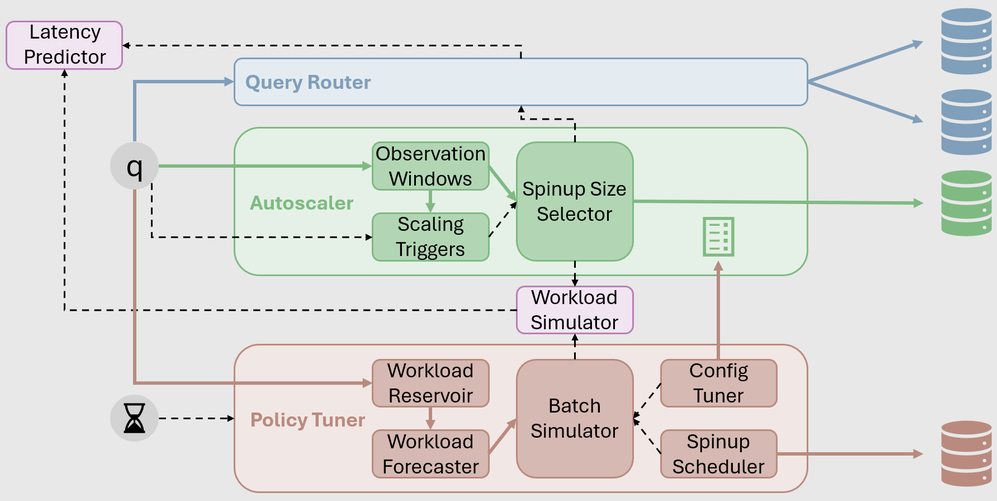
AutoSLO Operates Across Three Timescales

Query Router

- Use an **enhanced version of Iconq** [8], a concurrency-aware latency predictor.
- Estimate how **each routing option affects**:
  - Routed query
  - Running queries
  - Cost
- Pick the **best option**.

Autoscaler

- Maintain an **arrival and a completion window** over recent queries.
- When **SLO violations are high** among recent completions, trigger scaling.
- Hypothesize a cluster request of a given size; **simulate a short forecast** using copies of the arrival window and see how well it does.
- Repeat for different possible sizes **and pick the most promising size.**



Policy Tuner

- Use per-day, per-hour **workload reservoir** to generate multiple forecasted workload for the period (e.g., day) ahead.
- Simulate them to identify **earliest reliable load peak**, where enough workloads have a high enough SLO violation rate.
- Schedule a cluster spinup **shortly before the peak**; identify best size based on downstream helpfulness.
- Find the next peak, given the scheduled cluster spinup, and **repeat this process** for a bounded number of iterations.
- Then, **also tune reactive scaling** by tuning Autoscaler trigger aggressiveness.

AutoSLO Maintains SLOs Effectively and Efficiently

End-to-end SLOs are met effectively

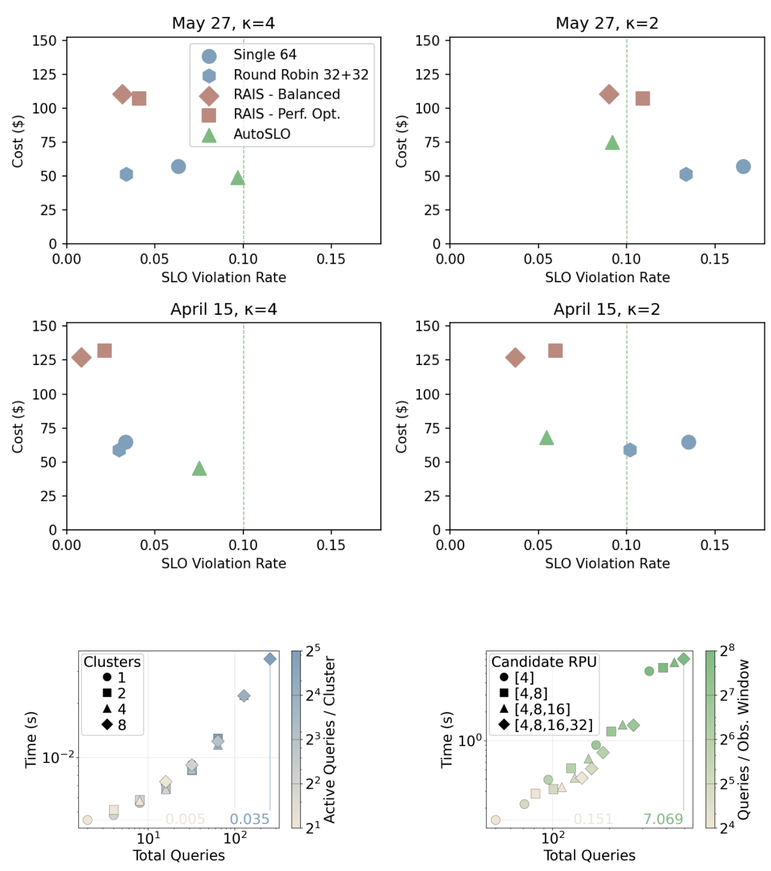
- AutoSLO is always the **cheapest method that meets the SLO target**.
- It reduces cost by a mean of **26.4% vs. per-scenario next-best**.
- Compared to the only baseline that always meets the target, it **reduces cost by a mean of 49.6%.**

Individual components are effective

- Policy Tuner: **one day of workload history** reduces SLO violation rate by a mean of 44.6%.
- Query Router: reduces SLO violation rate by a **mean of 47.8%** while also lowering cost (see full preprint).
- Spinup Size Selector: Reduces SLO violation rate by a **mean of 93.7%** while increasing cost by <12% (see full preprint).

Individual components are efficient

- Each component operates within its **target timescale:**
  - Routing decisions take **no more than 35 ms.**
  - Cluster size decisions take **no more than 7.1 s.**
  - See full preprint for additional efficiency experiments.



[1] Xinyi Zhang, Hong Wu, Zhuo Chang, Shuowei Jin, Jian Tan, Feifei Li, Tieying Zhang, and Bin Cui. 2021. ResTune: Resource Oriented Tuning Boosted by Meta-Learning for Cloud Databases. In Proceedings of the 2021 International Conference on Management of Data (Virtual Event, China) (SIGMOD '21). Association for Computing Machinery, New York, NY, USA, 2102–2114.
[2] Ryan Marcus and Olga Papaemmanouil. 2016. WiSeDB: a learning-based workload management advisor for cloud databases. Proc. VLDB Endow. 9, 10 (Jun 2016), 780–791.
[3] Sudipto Das, Feng Li, Vivek R. Narasayya, and Arnd Christian König. 2016. Automated Demand-driven Resource Scaling in Relational Database-as-a-Service. In Proceedings of the 2016 International Conference on Management of Data (San Francisco, California, USA) (SIGMOD '16). Association for Computing Machinery, New York, NY, USA, 1923–1934.
[4] Jennifer Ortiz, Brendan Lee, Magdalena Balazinska, Johannes Gehrke, and Joseph L Hellerstein. 2018. SLAOrchestrator: Reducing the Cost of Performance SLAs for Cloud Data Analytics. In 2018 USENIX Annual Technical Conference (USENIX ATC 18). USENIX Association, Berkeley, CA, USA, 547–560.
[5] Geoffrey X. Yu, Ziniu Wu, Ferdi Kossmann, Tianyu Li, Markos Markakis, Amadou Ngom, Samuel Madden, and Tim Kraska. 2024. Blueprinting the Cloud: Unifying and Automatically Optimizing Cloud Data Infrastructures with BRAD. Proc. VLDB Endow. 17, 11 (Aug 2024), 3629–3643.
[6] Gaurav Saxena, Mohammad Rahman, Naresh Chainani, Chunbin Lin, George Caragea, Fahim Chowdhury, Ryan Marcus, Tim Kraska, Ippokratis Pandis, and Balakrishnan (Murali) Narayanaswamy. 2023. Auto-WLM: Machine Learning Enhanced Workload Management in Amazon Redshift. In Companion of the 2023 International Conference on Management of Data (Seattle, WA, USA) (SIGMOD '23). Association for Computing Machinery, New York, NY, USA, 225–237.
[7] Vikram Nathan, Vikramank Singh, Zhengchun Liu, Mohammad Rahman, Andreas Kipf, Dominik Horn, Davide Pagano, Gaurav Saxena, Balakrishnan Narayanaswamy, and Tim Kraska. 2024. Intelligent Scaling in Amazon Redshift. In Companion of the 2024 International Conference on Management of Data (Santiago, AA, Chile) (SIGMOD '24). Association for Computing Machinery, New York, NY, USA, 269–279.
[8] Ziniu Wu, Markos Markakis, Chunwei Liu, Peter Baile Chen, Balakrishnan Narayanaswamy, Tim Kraska, and Samuel Madden. 2025. Improving DBMS Scheduling Decisions with Fine-grained Performance Prediction on Concurrent Queries. Proc. VLDB Endow. 18, 11 (Aug 2025), 4185–4198.