Universitat Pompeu Fabra Barcelona

Boris Sedlak, Víctor Casamayor Pujol, Xiaobo Qu, and Schahram Dustdar

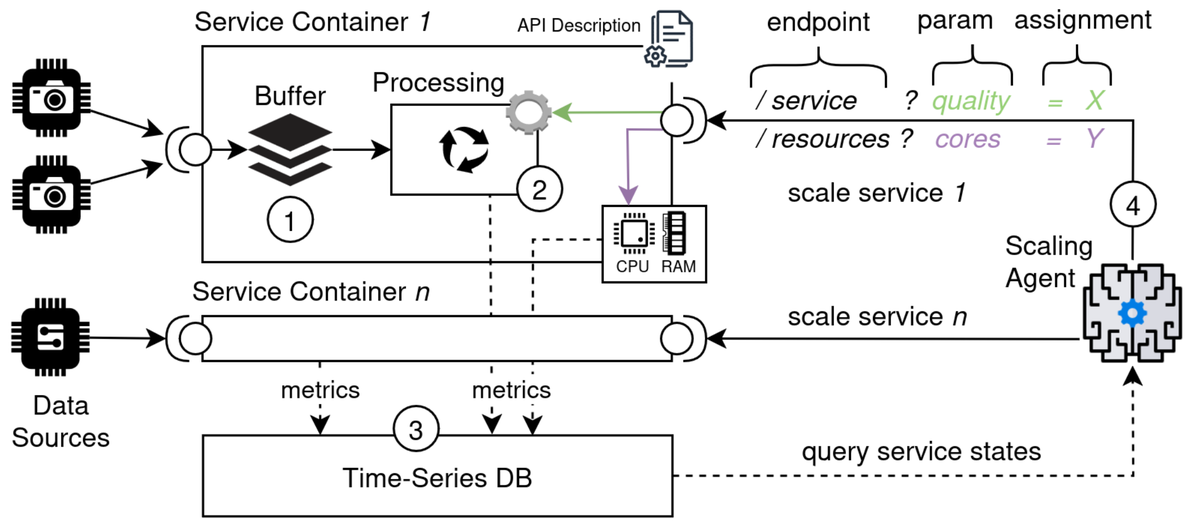
How to Gradually Improve Autoscaling through Analysis of Structural Knowledge

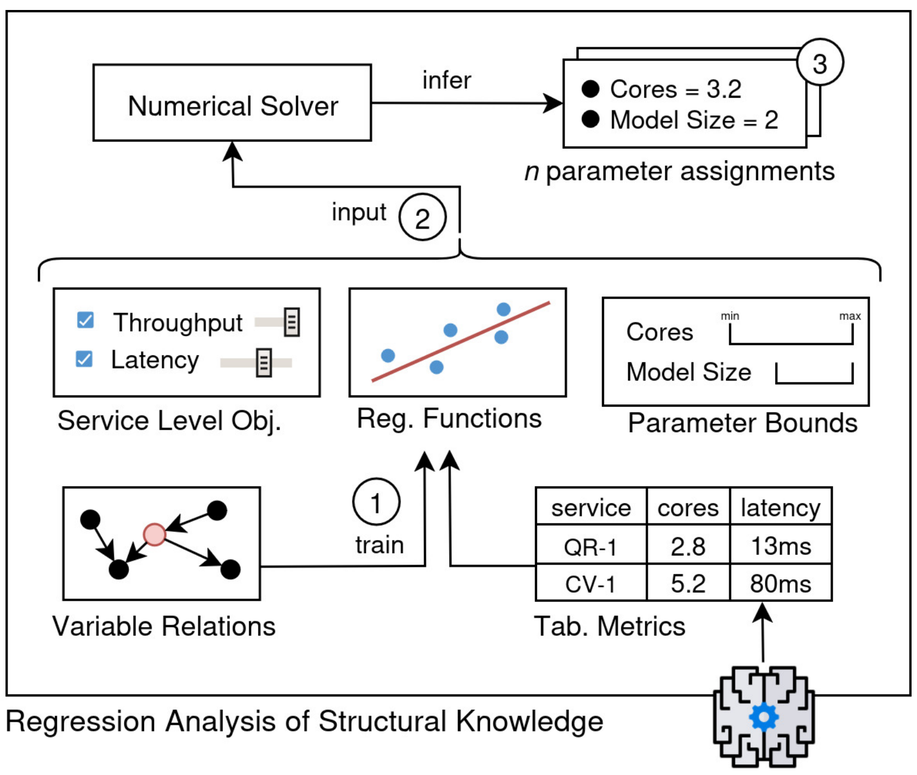
Autoscaling Environment

Containerized service execution on **one** host; scale services through **multiple** service/device params.

Service execution: (1) buffer and (2) process data, and (3) log service and device metrics in a DB.

Scaling agent: (4) develop a scaling policy and acts on service and device through REST API.





Scaling Agent Cycle

Analyzes **structural knowledge** to infer optimal scaling policy under custom application SLOs and parameter bounds.

1) Using metrics and **variable relations**, estimate state transitions through regression functions. Quantifies impact of **interventionable parameters** on global SLO fulfillment.

2) Supply regression functions, SLOs, and parameter bounds to numerical solver; through **gradient descent** find optimal assignment for multiple services under resource limitations.

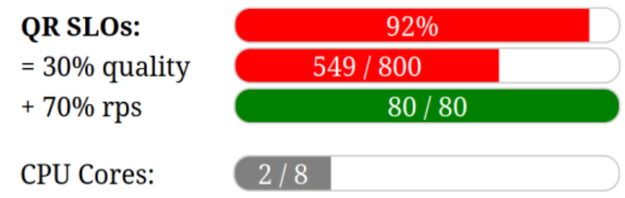
3) Use exposed REST API to scale the services parameters and their containerized resources limitations accordingly.

Demo Contents

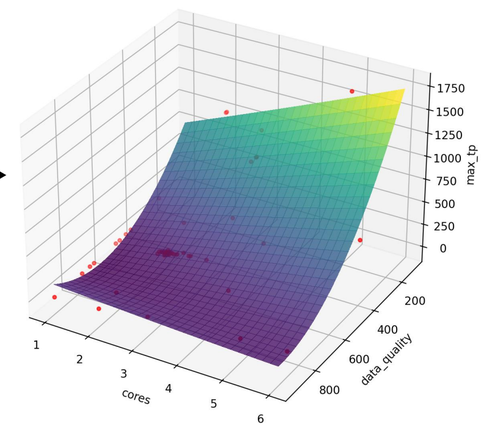
Scaling agent needs to **optimize** the global SLO fulfillment of three processing services that compete for 8 CPU cores. Agent operates in cycles of 10s, continuously executing above cycle. For 30 iterations, agent **explores**, afterwards exploits.

Service output: Current output of three processing services, i.e., detected QR codes, Yolov8 objects, and Lidar mobile map

Service states and SLOs: For interventionable and dependent variables, show SLO fulfillment relative to thresholds; overall SLO fulfillment **weighted** according to preferences. CPU cores are allocated between services within global limitations.



Regression model: Show impact of interventionable parameters on dependent variable (i.e., **expected rps**). Improve over time.



Demo **Video**

Project **Repo**