Vulcan: Automatic Query Planning for Live ML Analytics

NSDI 2024

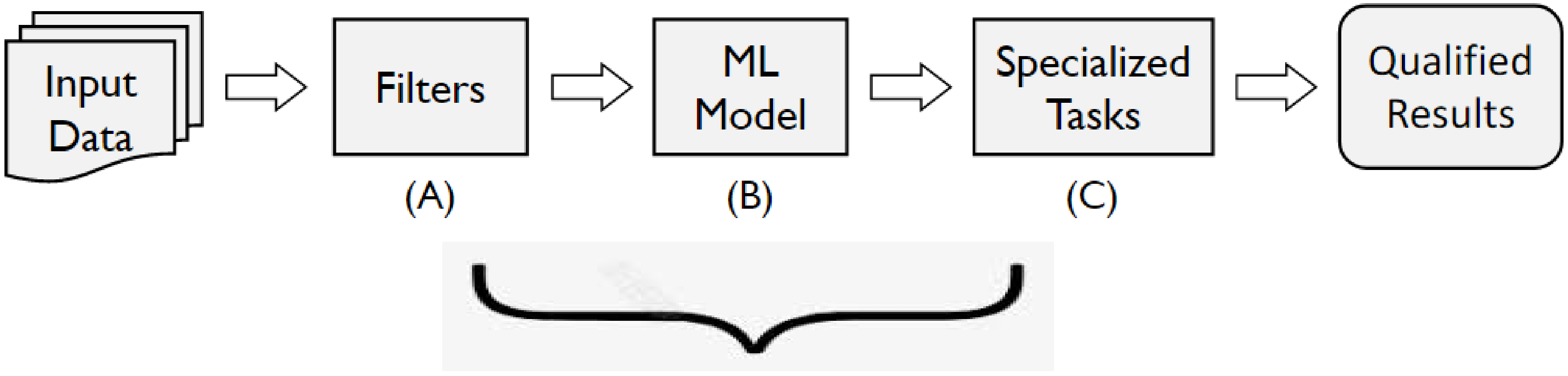
Yiwen Zhang, Xumiao Zhang, Ganesh Ananthanarayanan, Anand Iyer, Yuanchao Shu, Victor Bahl, Z. Morley Mao, Mosharaf Chowdhury University of Michigan, Microsoft, Georgia Institute of Technology, Zhejiang University, Google

汇报人：冯敏远

2024年7月11日

前置知识

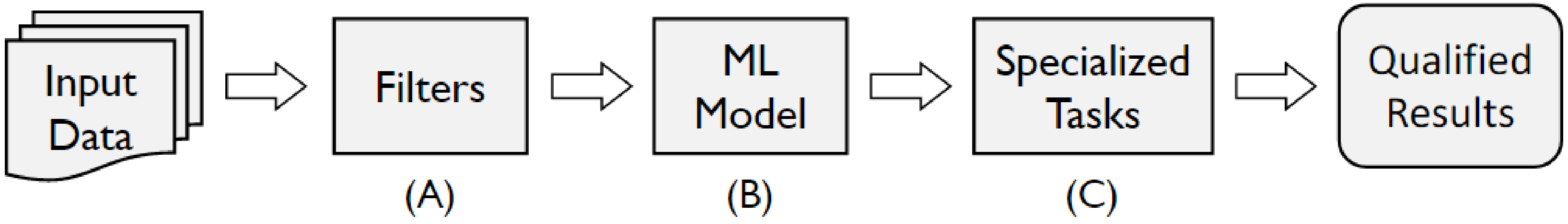
➢ ML PipeLine



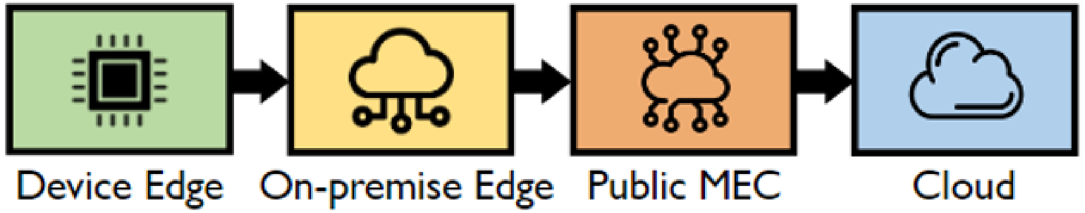
PipeLine

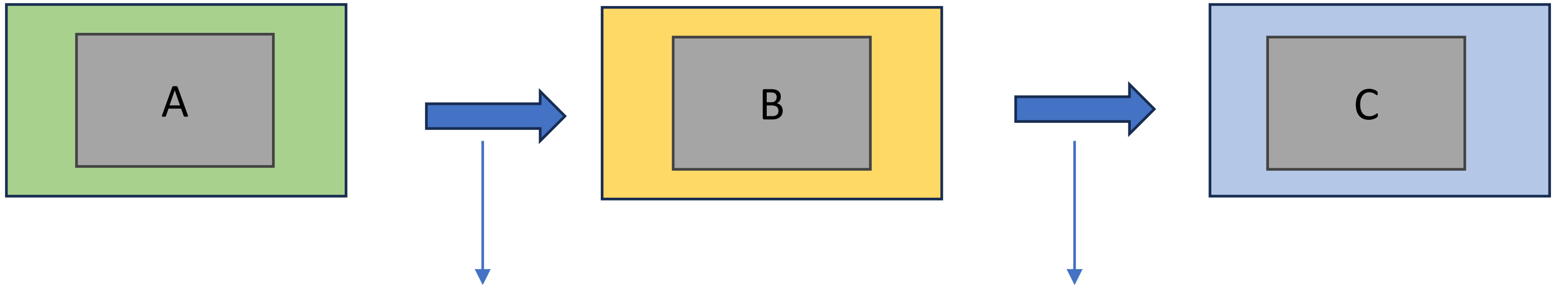
可以理解为：流水线

前置知识



- **Heterogeneous**





Resource Consumption、Latency

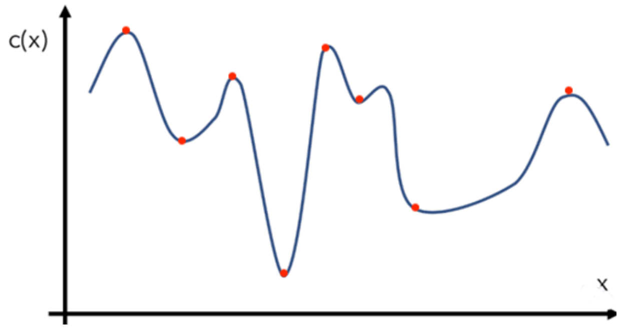
前置知识

➢ Bayesian Optimization(BO)

1. used for global optimization problems(compare with Greedy Algorithm)
2. suitable for expensive or computationally complex black-box functions.

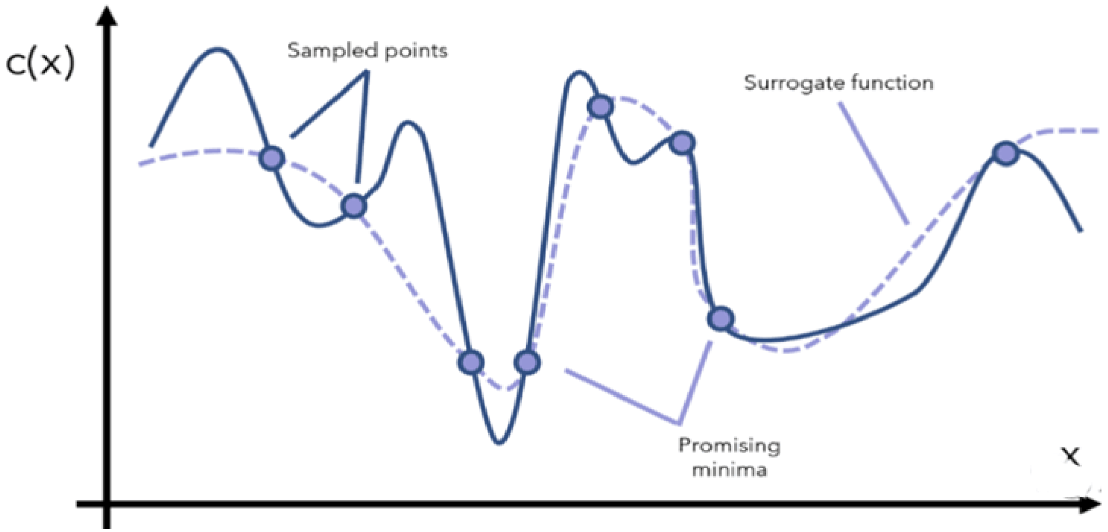
Assume we have an objective function C(x) with a high evaluation cost, and we aim to find the x that minimizes C(x).

1. initialization: Select several initial points x1,x2,..., and evaluate C(x)

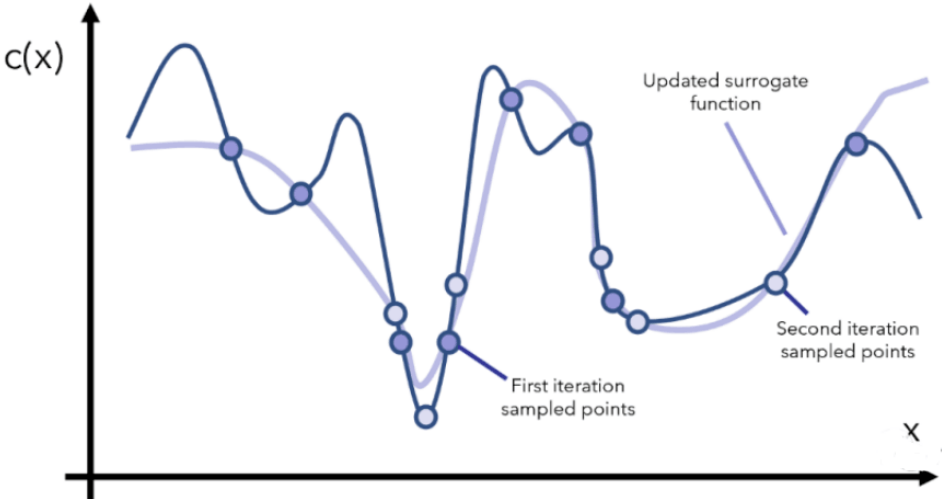


前置知识

2. Select the next sampling point



3. Update surrogate optimization



4. Repeat the above two steps

提纲

- 研究背景
- 研究问题
- 系统设计
- 实验评估
- 工作总结

提纲

- 研究背景
- 研究问题
- 系统设计
- 实验评估
- 工作总结

研究背景

Live Traffic Analysis

Real-time Speech Recognition

Autonomous Driving

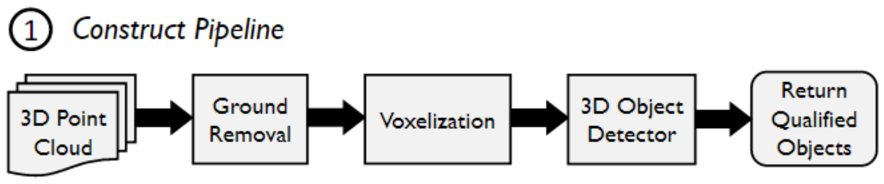
Stock Market Monitoring

Live ML Analytics' Requirement Grows Rapidly

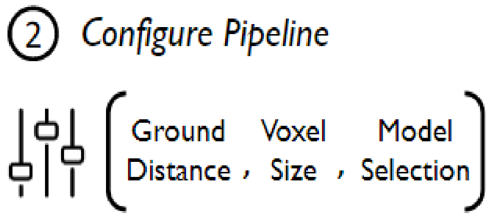
研究背景

➢ the workflow of live ML query processing(offline)

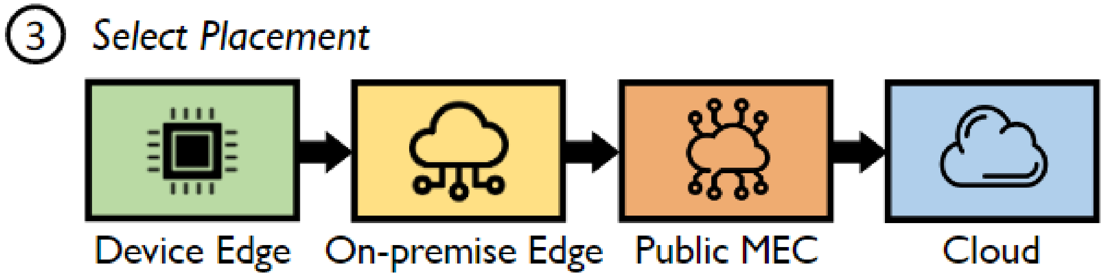
- **constructing the pipeline**



- **selecting configurations of the pipeline operators**



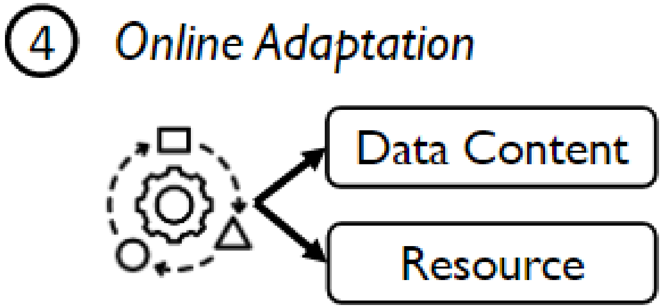
- **determining the physical placement of pipeline operators across infrastructure tiers**



➢ the workflow of live ML query processing(online)

- **Performing online adaptation**

(performance is affected by runtime dynamics due to resource changes and data variations)



An ideal solution should adapt to runtime dynamics by adjusting the pipeline's pipeline, its configurations, and placement.

研究现状

➢ Current Research Shortcomings

- **1. Existing research are mostly piecemeal but lacks systematic approaches.**
  (e.g.,focus on optimizing compute resources while neglecting the impact of other resources such as network)

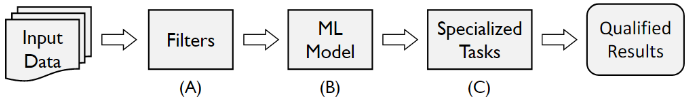
- **2. There is no systematic method to automatically construct pipelines**
  (although there are declarative query languages (e.g., SQL) for ML queries)

- **3. Deployments rely on past experience with simple heuristics**
  (when choosing physical placement of pipeline components, one cannot afford to exhaustively search)

研究现状

➢ Current Research Shortcomings

Single Module



- **4. Assuming that the ML analytics component is a single module rather than a pipeline**

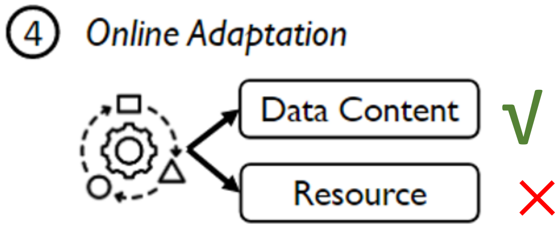
  (Existing research mostly focuses on query configuration selection)

- **5. Heavily rely on domain-specific insights into video content**

  （inapplicable to general ML scenarios beyond video analysis.）

- **6. Fail to effectively adapt to variations in computation and network resources**.

(in edge environments, resource fluctuations are common, necessitating a more flexible adaptation mechanism)

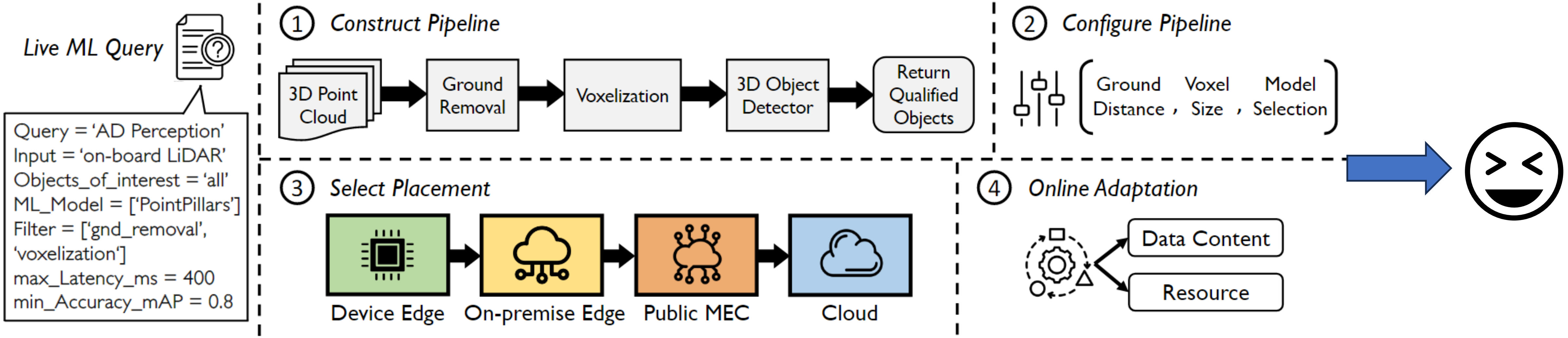


提纲

- 研究背景
- **研究问题**
- 方法设计
- 实验评估
- 工作总结

研究问题

how to perform automatic query planning (four steps) for live ML queries based on user-provided performance requirements ?



1.Construct Pipeline

2.Configure Pipeline

3.Select Placement

4.Online Adaptation

研究问题

➢ Our Goal

optimize latency and accuracy,

while minimizing the network and compute resource consumption

accuracy ↑

latency ↓

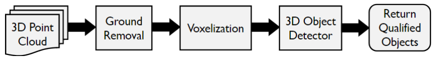
network resource consumption

compute resource consumption ↓

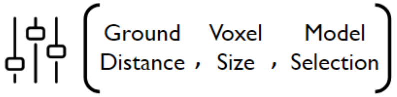
研究问题

- **How to do**

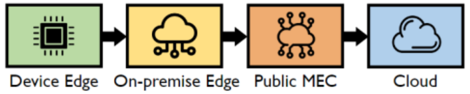
1.Construct Pipeline



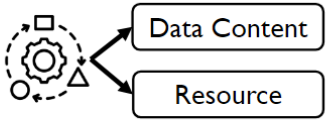
2.Configure Pipeline



3.Select Placement



4.Online Adaptation



place the filters in what order

care both data content and resource changes

reduce the time complexity of search

提纲

- 研究背景
- 研究问题
- **系统设计**
- 实验评估
- 工作总结

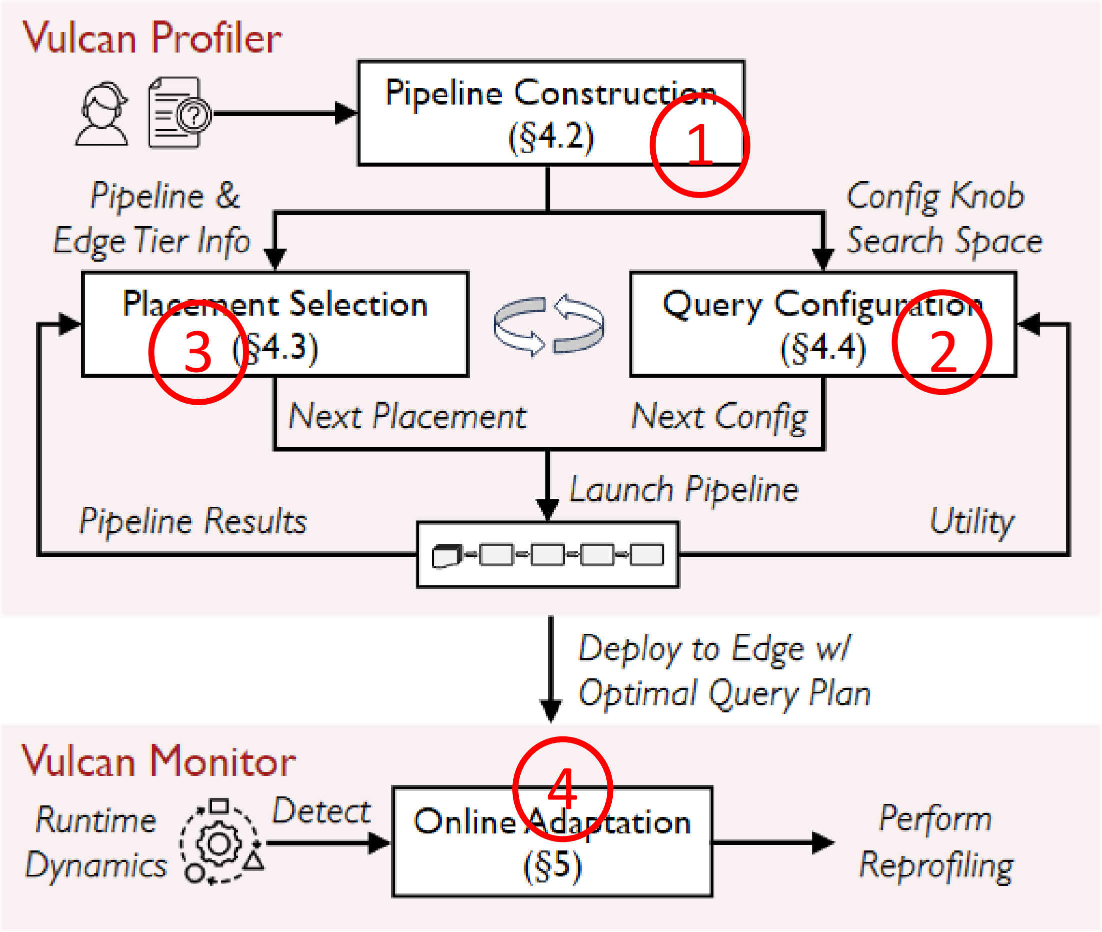
系统设计

- **Vulcan**

An ML analytics system that provides **automatic query planning** for live ML queries.

It takes charge of the entire lifecycle of a ML query by

- Construct Pipeline
- Configure Pipeline
- Select Placement
- Perform Online Adaptation



系统设计

➢ Construct

defines a novel metric to quantify each filtering operator

➢ Configuration

explores the best combination of configuration knobs using Bayesian Optimization (BO)

➢ Placement

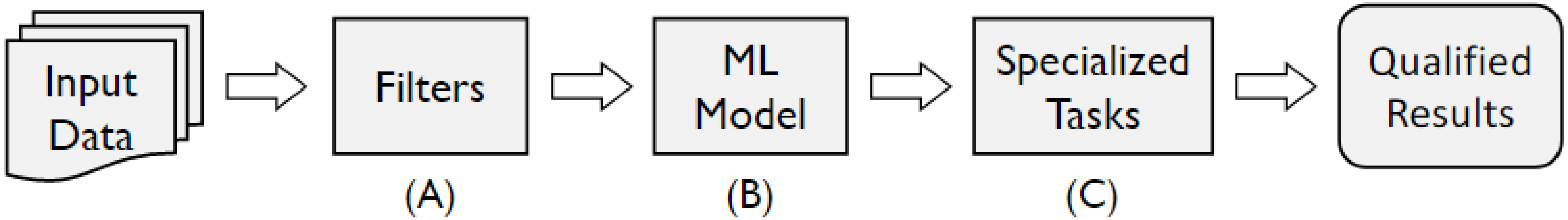
identifies components that are independent of placement

➢ Adaptation

(i) designing programming interfaces that allow for dynamic updates to live pipelines modules without disrupting them
(ii) leveraging prior knowledge to make faster decisions on modifying configurations and placement

系统设计

➢ 1.Construct



Template:

(A) filtering modules
(B) the ML model
(C) specialized modules

系统设计

- **Selecting the Ordering of Filters**
  - Recall： the accuracy of positive predictions
  - Precision： the coverage of actual positives

β： how many times recall is considered more important than precision

$$F_{\gamma} = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

Filter_A → Filter_B → Filter_C → Filter_D

系统设计

- **Challenge**
  - the recall of a filter can change based on its preceding filter

Filter_A → Filter_B → Filter_C

Fγ = 0.8

Filter_A → Filter_C → Filter_B

Fγ = 0.9

  - configuration knob leads to different precision or recall measurements.

config1	**10**

Fγ = 0.8

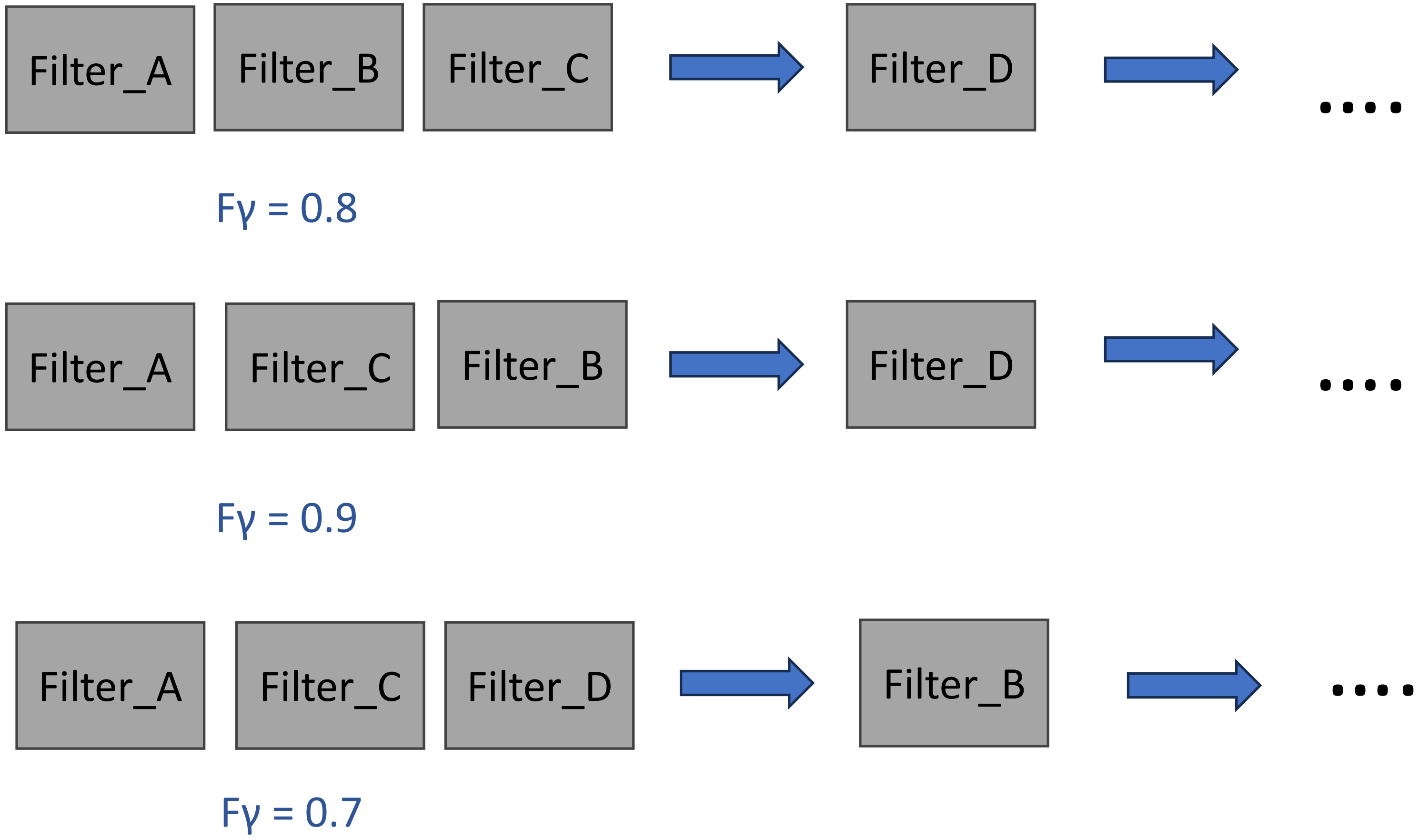
config2	**9**

Fγ = 0.9

Filter_B

系统设计

- **Solution for the first challenge**
  - treating a sequence of filters as a bulk filter



系统设计

- **Solution for the second challenge**
  - choose representative configuration settings （20% 50% 80%）

Filter_B

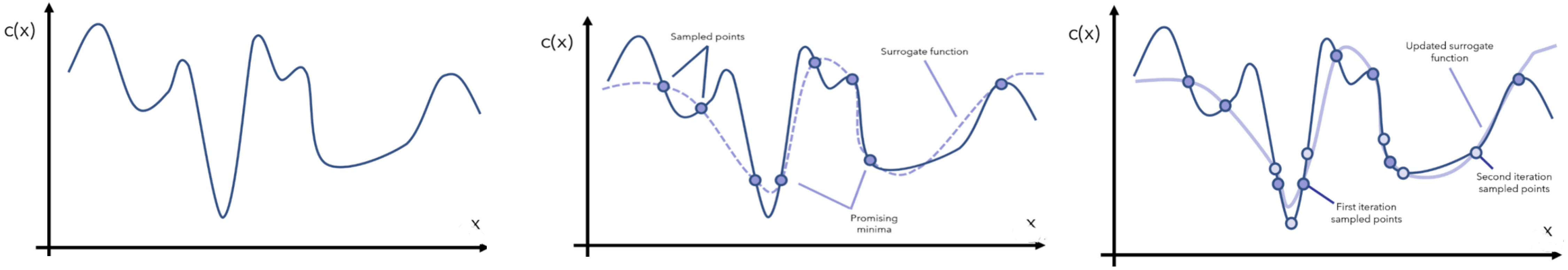
config	**10**	default	Fγ = 0.8
config	**2**	20%	Fγ = x1
config	**5**	50%	Fγ = x2
config	**8**	80%	Fγ = x3

$O(n) \longrightarrow O(3n)$

系统设计

➢ 2.Configuration

- **leverage Bayesian Optimization (BO) to efficiently explore pipeline configurations that achieve the best performance with minimized resource consumption**

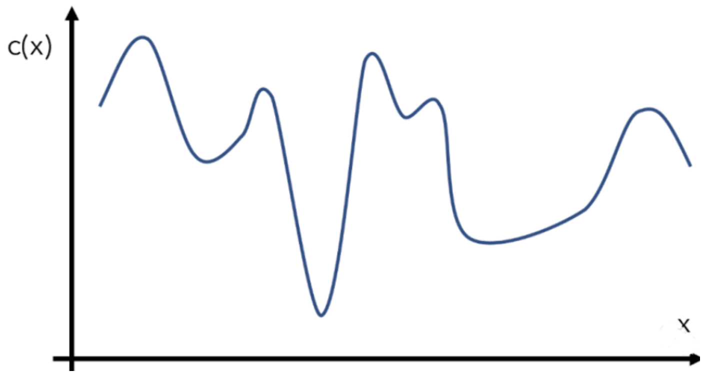


(1) (2) (3)

系统设计

- The input x is **the set of query configuration knobs**
- the output of f is **the utility value, Uq,p,c** (pipeline q 、placement p 、a set of configurations c）.

start with **N random sets of input query configurations** as initial observations for BO to learn the rough shape of the objective function.

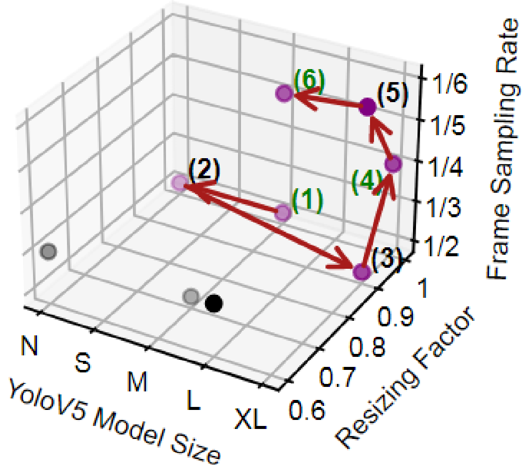


N = 3

系统设计

- **Vulcan stops BO when the improvement of the utility value is less than a threshold for a few consecutive runs**

  (i.e., 10% for 5 consecutive runs, 😆).



a pipeline **q** with placement **p** and pipeline configurations **c**

- $U_{q,p,c} = P_{q,p,c}/R_{q,p,c}$
- $P_{q,p,c}(A, L) = \gamma \cdot \alpha_A \cdot (A - A_m) + (1 - \gamma) \cdot \alpha_L \cdot (L_m - L)$
- $R_{q,p,c} = \alpha_{gpu} \cdot R_{gpu} + \alpha_{net} \cdot R_{net}$

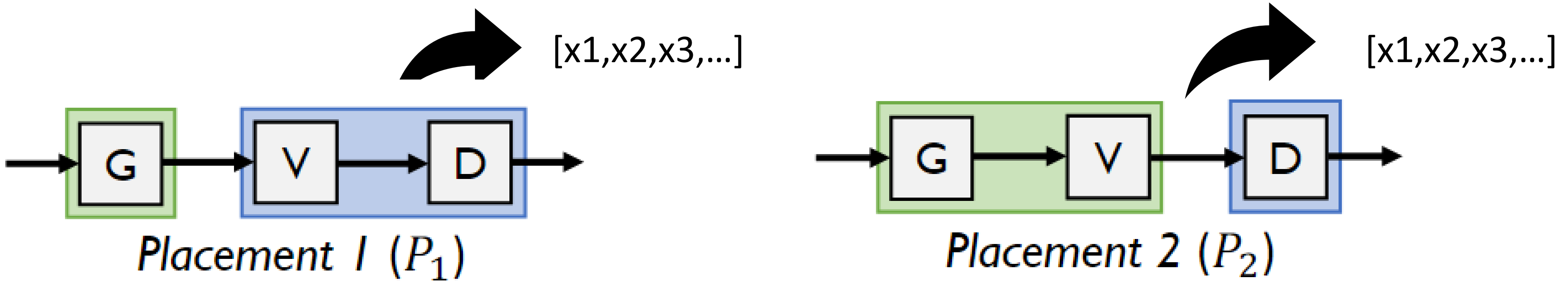
系统设计

➢ 3.Placement

- **Two Observations:**

  1) Query accuracy is independent to placement.

  2) Amount of data generated after each pipeline operator does not change with placement.

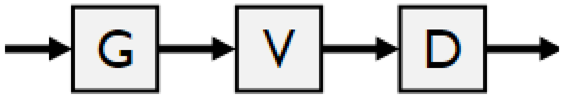


Accuracy = 0.9

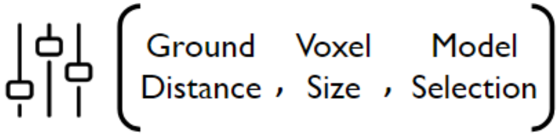
Accuracy = 0.9

系统设计

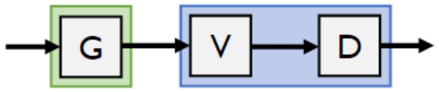
- **deploy the pipeline offline in the datacenter only once per selected query configuration**



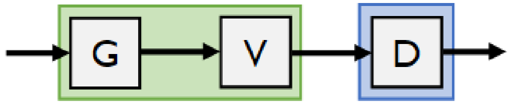
- **Get the configuration knob from the second step**



- **calculating additional latency and resource consumption components introduced by the placement, while reuse the same query accuracy result**

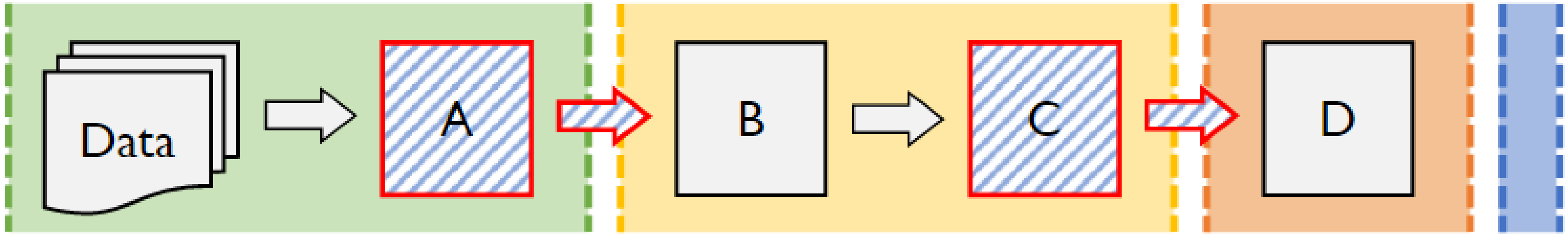


Placement 1 (P_1)



Placement 2 (P_2)

系统设计



device edge → on-premise edge → public MEC →cloud.

shaded operators are used to calculate the additional latency introduced by the placement.

Given a total of M placement choices and N combinations of pipeline configurations

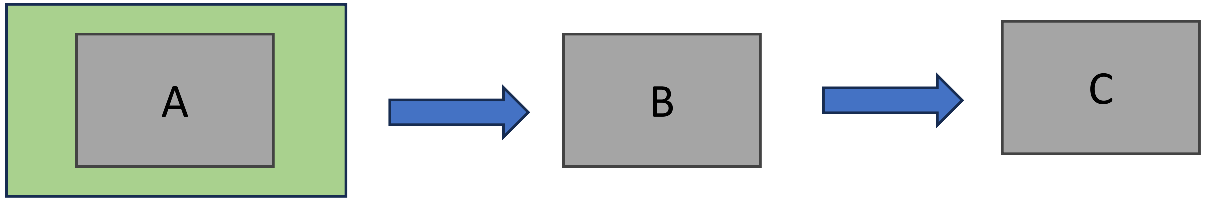
O(MN) → O(N)

系统设计

- exploring all feasible placement choices may **still incur large search cost** as the profiler strives to find a good configuration for an unpromising placement.

➢ **early pruning**

- the utility value,
- obtain utility values below the threshold, we early prune the current placement choice



If U < threshold, pass

系统设计

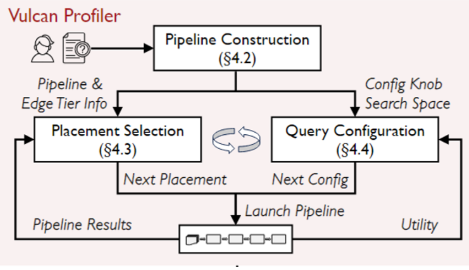
➢ 4.Adaptation

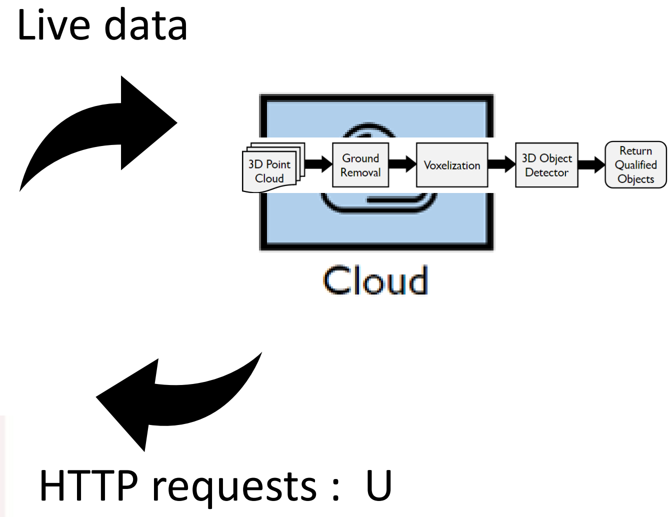
- **Two ideas:**

 1) monitor utility(U) change to detect runtime dynamics
 2) leverage prior knowledge during reprofiling

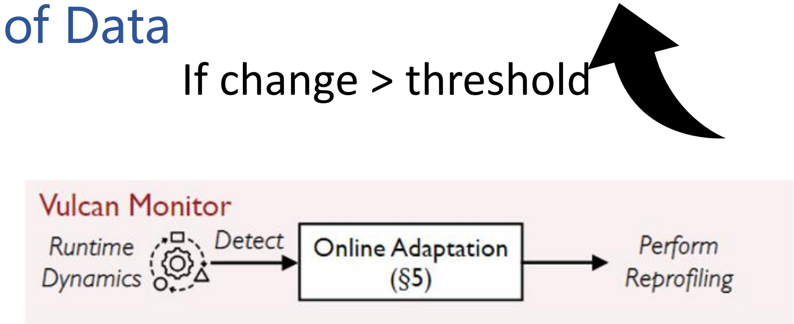
 the threshold of utility change (10%)

 1. Starting the Replication Pipeline
 2. Periodic Reception and Evaluation of Data
 3. Updating Utility Values
 4. Triggering Reprofiling
 5. Reprofiling and Optimization

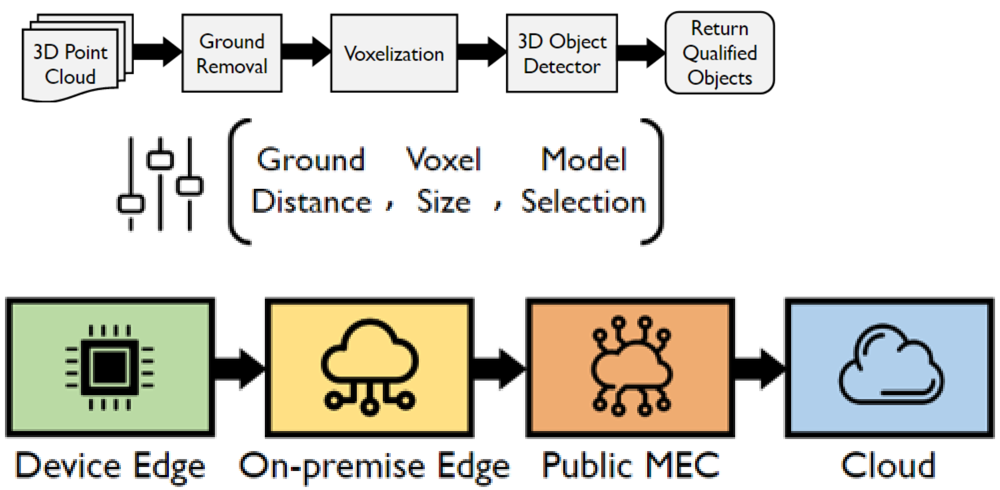




If change > threshold

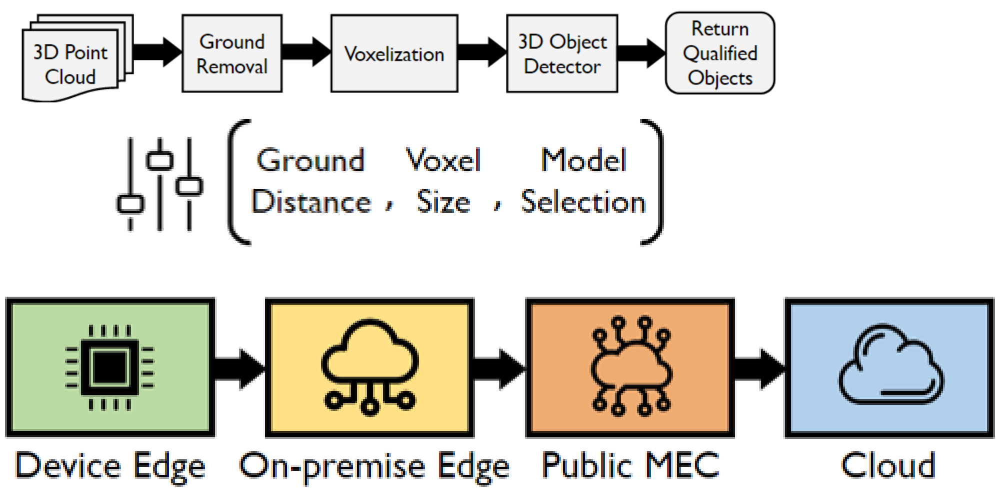


系统设计



A

reprofile



B

the most recent top-K and worst-K configuration per placement choices (K = 3)as initial data points in BO

提纲

- 研究背景
- 研究问题
- 系统设计
- **实验评估**
- 工作总结

实验设置

Live ML Applications：
- Video Monitoring
- Autonomous Driving Perception
- Automatic Speech Recognition

Data Set：
- videos captured by traffic cameras in Bellevue and Washington
- LiDAR sensor data from nuScenes
- Automatic Speech Recognition:

Evaluation metrics：
- profiling time
- latency
- accuracy
- query resource consumption

Offline Comparison：
- Exhaustive Search
- VideoStorm
  - Original
  - Plus
- JellyBean
- Vulcan

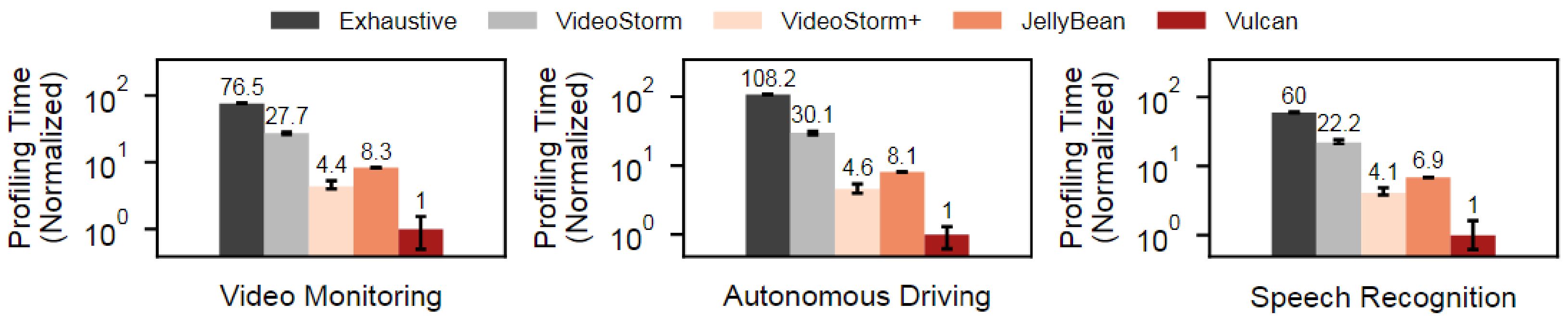
Online Comparison：
- Chameleon
- LLAMA
- Vulcan

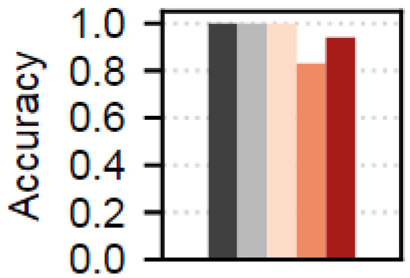
Emulation Setup:
- device edge
- on-premise edge
- public MEC
- cloud

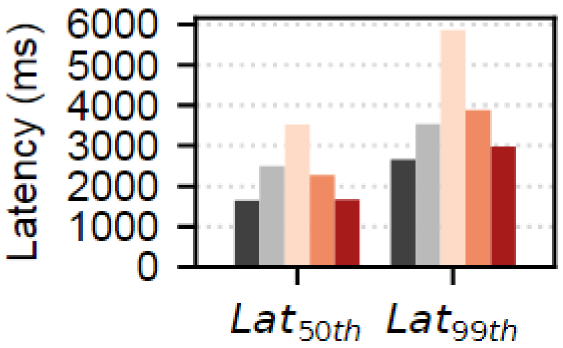
实验评估

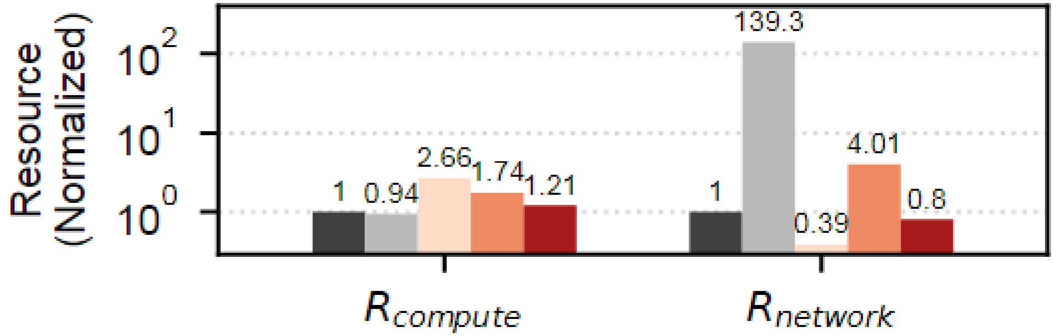
- **PERFORMANCE EVALUATION**

End-to-End Improvement









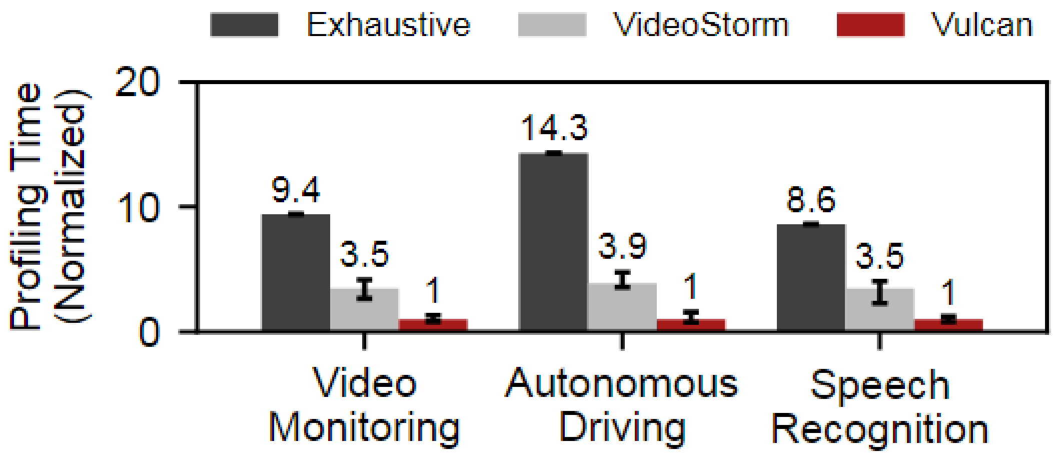
Profiling Time: Far Ahead

others: Not Bad

实验评估

- **PERFORMANCE EVALUATION**

the same pipeline and the best placement choice (Better Query Configurations)

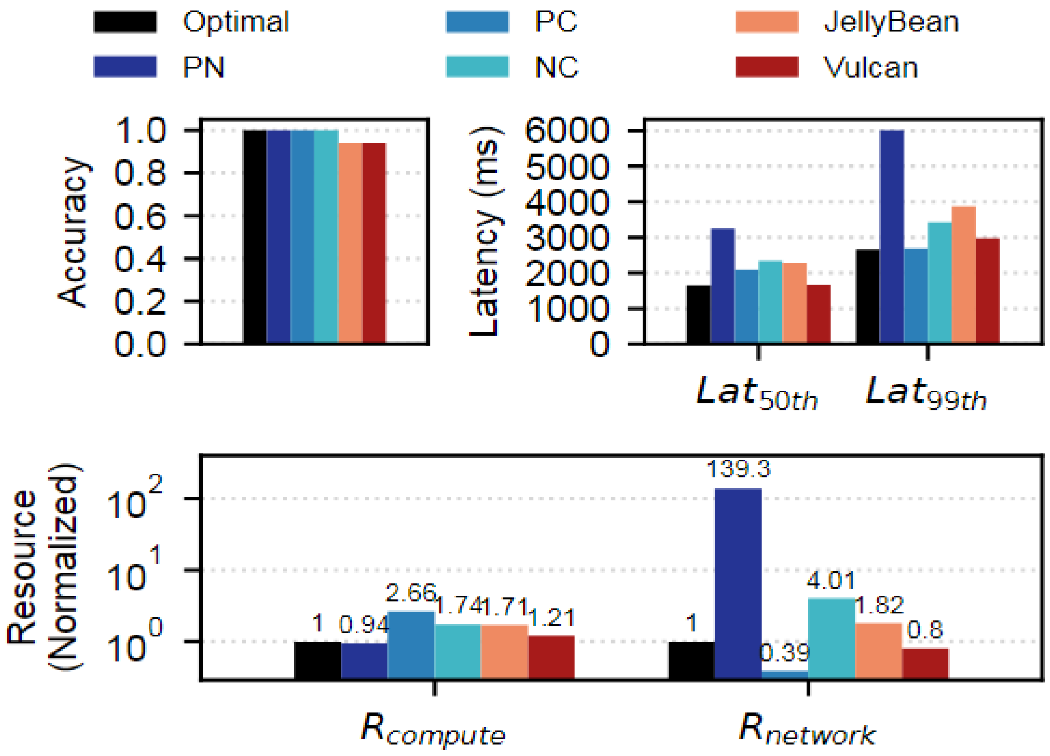


The Advantage Of BO

实验评估

- **PERFORMANCE EVALUATION**

optimal pipeline configuration and the same pipeline（Selecting Better Placement)



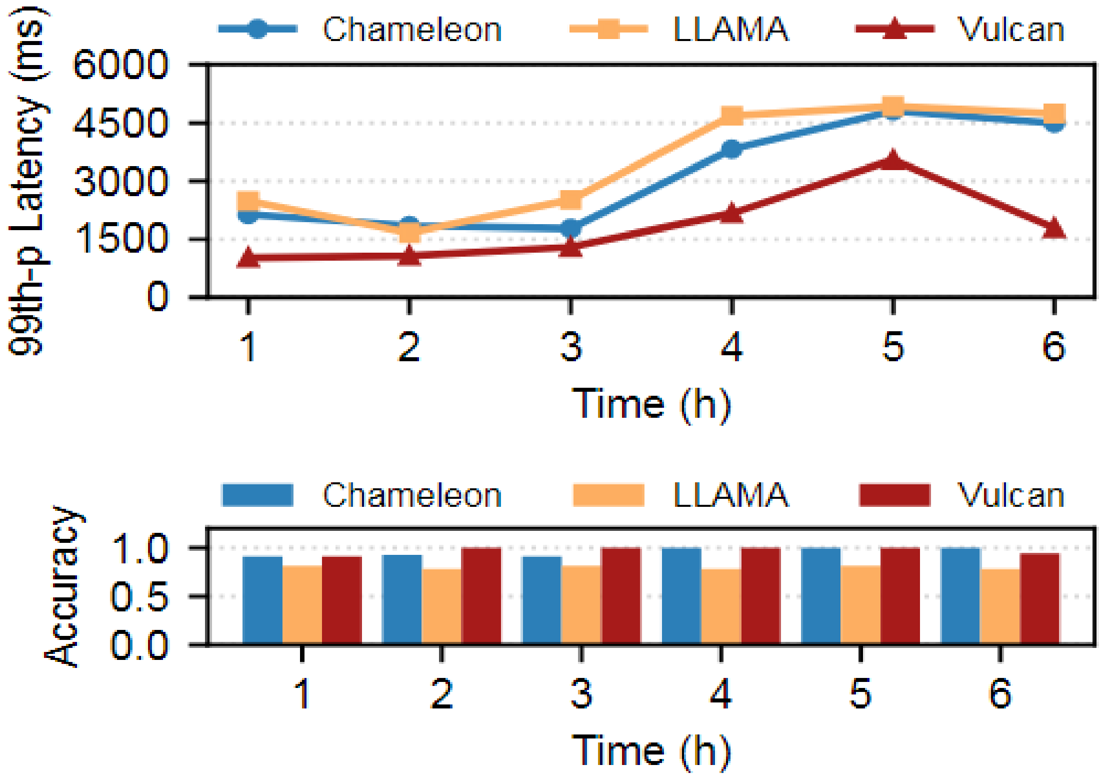
video monitoring

Excellent Average Performance

实验评估

- **PERFORMANCE EVALUATION**

Handling Runtime Dynamics



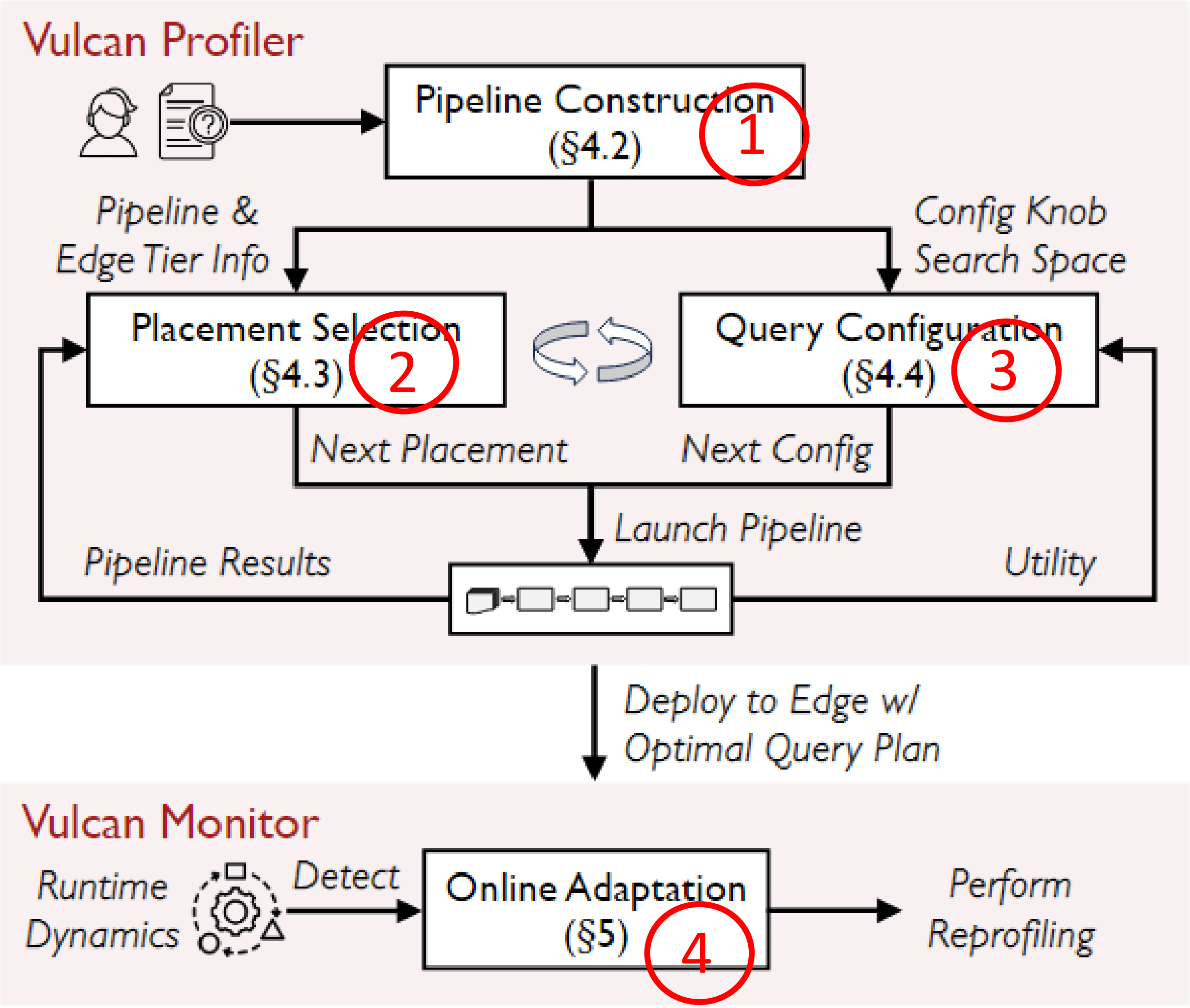
Low Latency & High Accuracy

提纲

- 研究背景
- 研究问题
- 系统设计
- 实验评估
- **工作总结**

实验评估

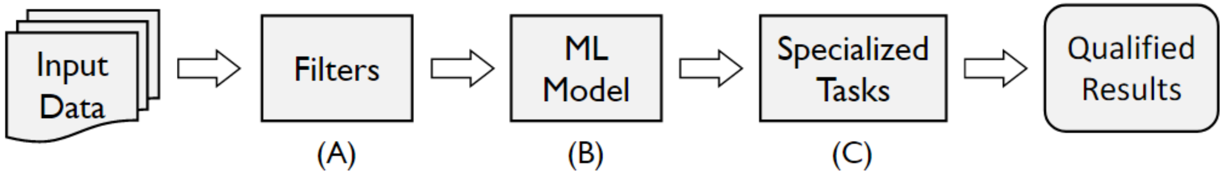
- **Summary**



- **Automatic Query Planning**
  - **Construct Pipeline**
    - Filters sorting
  - **Select Placement**
    - Reuse the result
  - **Configure Pipeline**
  - **Perform Online Adaptation**
    - BO

一些想法

1. Can the idea of filter sorting be applied to the latter two



2. Can the idea of filter sorting be applied to classifiers in video analytics

3. The accuracy variation caused by transient data changes was not considered

➢ Some optimization techniques

- leveraging prior knowledge
- Early pruning.
- Analyzing data irrelevance to reuse resources and reduce complexity
- Quantifying the problem

Q&A

2024年7月11日

冯敏远

Southeast University Intelligent Internet of Things Laboratory