A Deep Learning Based Approach to Performance Optimization in Big Data Systems

Fei Song, Zhao Cao, Khaled Zaouk, Yanlei Diao

Motivation and Overall Architecture

Motivation

- To run an analytic task on a big data system, the execution requires setting the system parameter (θ).
- The settings of θ have significant impact on user performance goals, such as latency (L), throughput, and cost. Also these goals are often contradict with each other.
- Nowadays, systems let users to set up θ, with no guarantee/prediction on the performance goals.
- Big data systems are very complex. It is hard for the user to set up θ to optimize the system performance.

Our Framework

Cost Model

- For a finite set of well defined operators, one can manually construct a general purpose cost model by careful analysis.
- Big data systems do not have a fixed set of operators. Hence it is too complex to build a general cost model manually.
- To our best knowledge, this is the first effort to have a framework, which is deep-learning based, for automatically building a cost model in-situ.

Multi-Objective Optimization

- Construct approximate Pareto Optimal Skyline efficiently.
- Explore Pareto Optimal plans that capture tradeoffs between various user performance goals.

Cost Model

Cost Model with Approximation

Workload characteristics

- A vector W_i is the description of P_i , denoted as $W_i(t)$.
- It is stable within a time period, but may evolve slowly over time.
- W_i is a hidden variable. So we need to extract it from data (manually constructed in some previous works).

Approximation

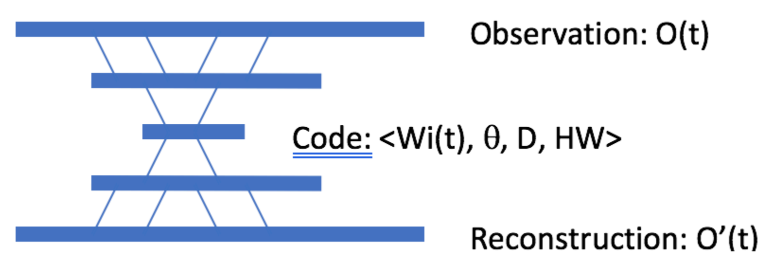
- $F(W_i(t), \theta, D, HW) + X_\varepsilon(t) = O_i(t)$
- X can be viewed as Random Noise, we focus on simulating F.
- when c is a small constant, we can use $W_i(t\text{-}c)$ in place of $W_i(t)$ as an approximation.
- $O_i(t) = F(W_i(t-c), \theta, D, HW)$

Deep Learning Approach

Train a deep auto-encoder I* to

Minimize avg(distance(O(t), O'(t))

From code, we extract W.



With W extracted, the next step:

Train a neural network regression model F* as the cost model.

- For a P_i with enough training data: good result, predicting error is low, less than 10%.
- For new P_i without enough training data: still needs to improve.

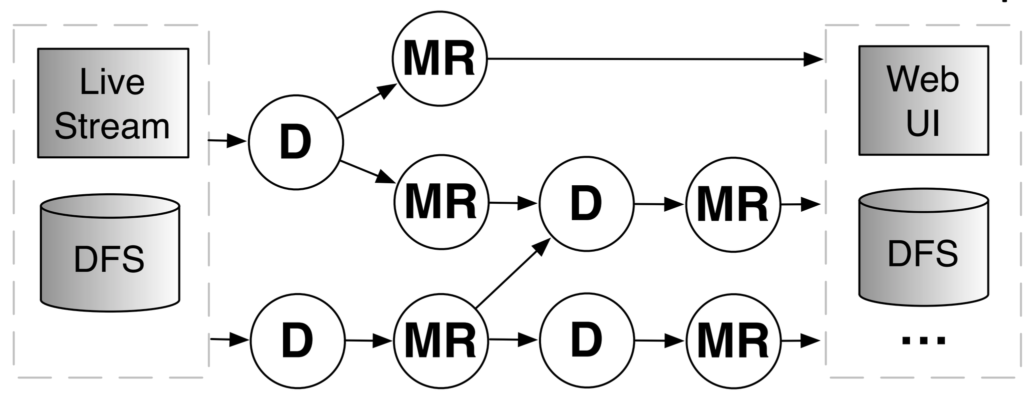
System Model

System parameters

- P_i: logical dataflow program description.
- θ: System configuration, such as the degree of parallelism and granularity of scheduling.
- HW: Hardware description.
- D: Data description.
- $O_i(t)$: Observations taken at time t, includes latency, throughput, CPU, MEM, I/O, Network, etc.

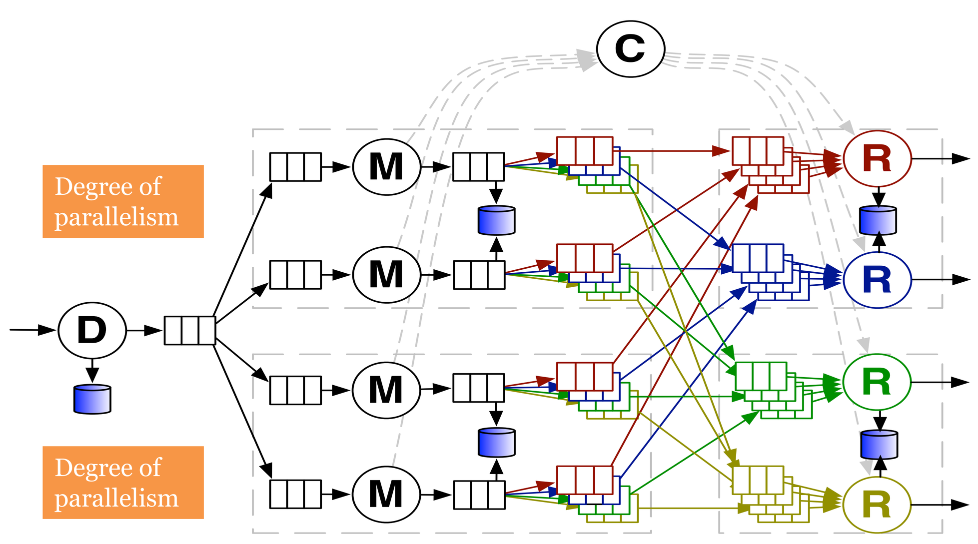
Logical Plan

A complex analytical task (type i) can be modelled as a **logical dataflow program, denote as P_i :**

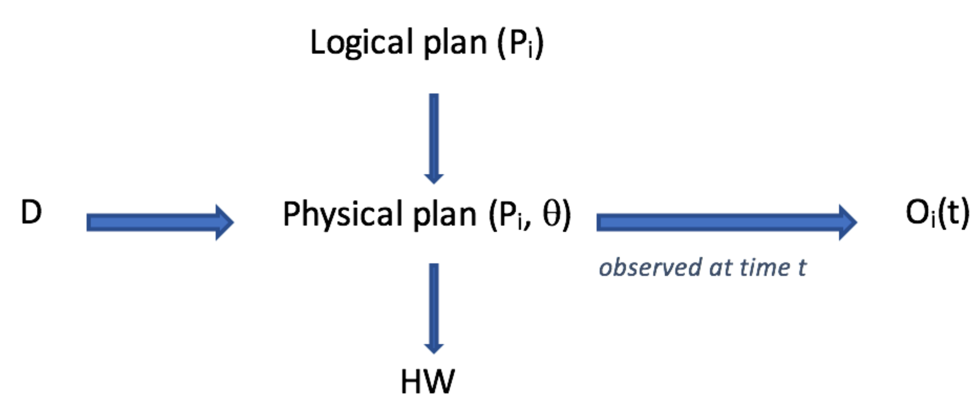


Physical Plan

A close look on **a MR-pair in physical execution plan**, with θ set to a particular value:



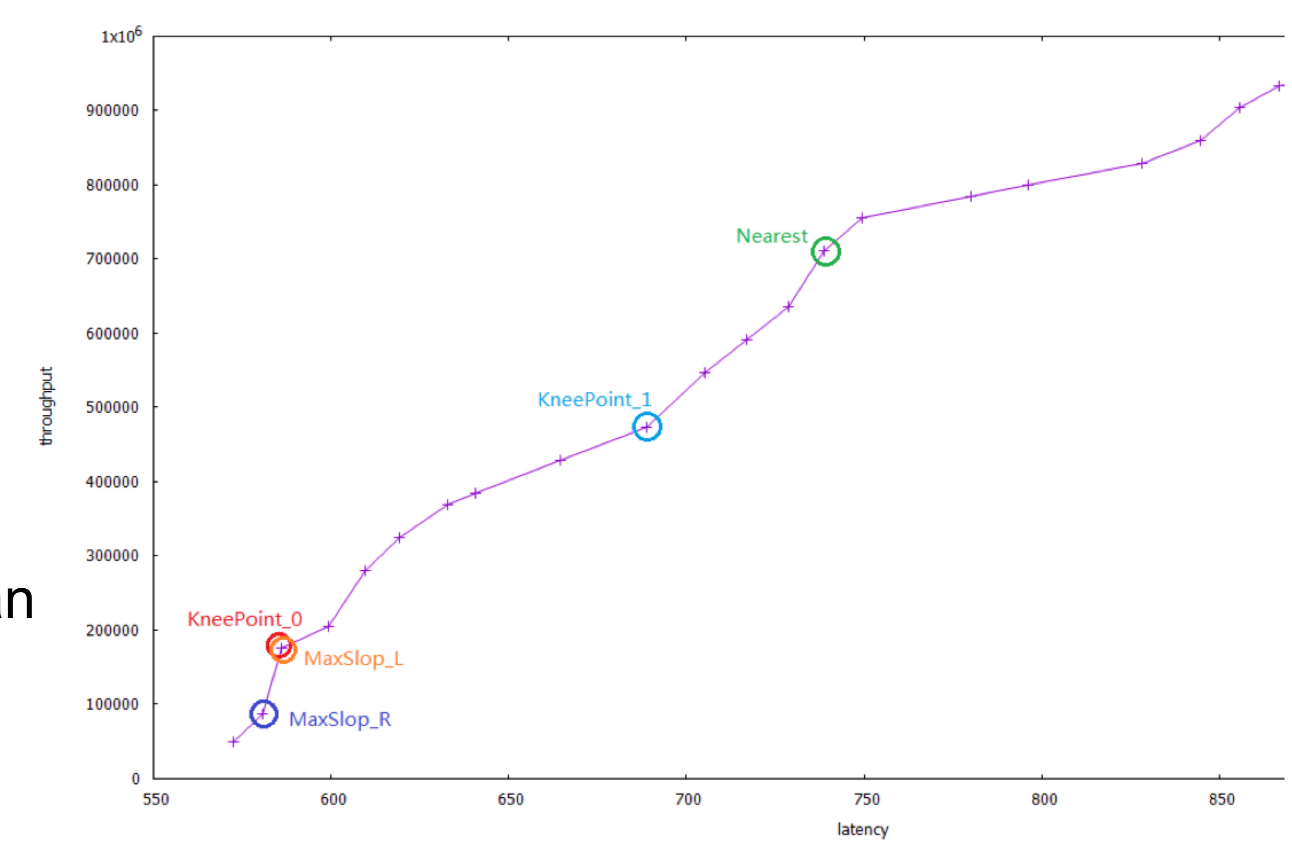
System abstraction



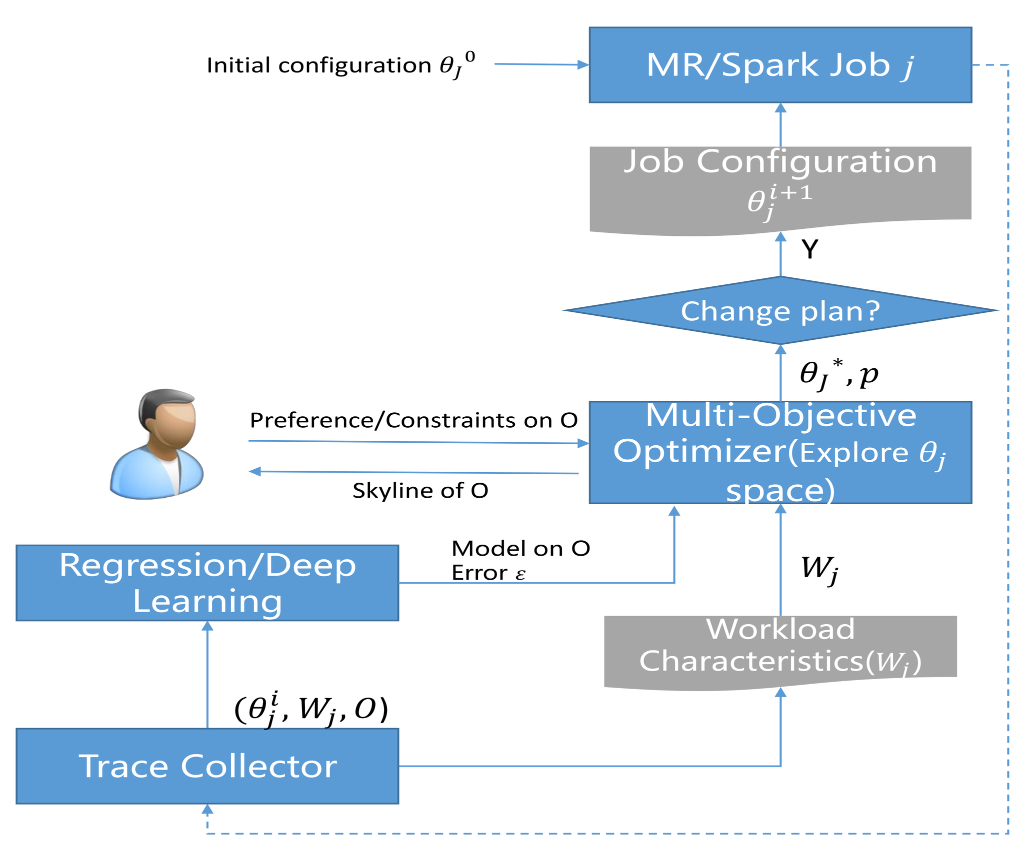
Multi-Objective Optimization

Overall approach

- Pareto Optimal based.
- An algorithm to find the next interesting section to explore.
- Minimize uncertain space. Intuitively, it means where the pareto optimal skyline can be an arbitrary shape.
- When the user is not sure about how to distribute preference, it recommends best plans based on different algorithms.



Pareto optimal skyline
With different optimal plan recommendations



Optimization workflow

CEDAR, INRIA Saclay and LIX (CNRS UMR 7161 and Ecole Polytechnique)