INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING

ICPP/2021/CHICAGO/USA

acm In-Cooperation

sighpc

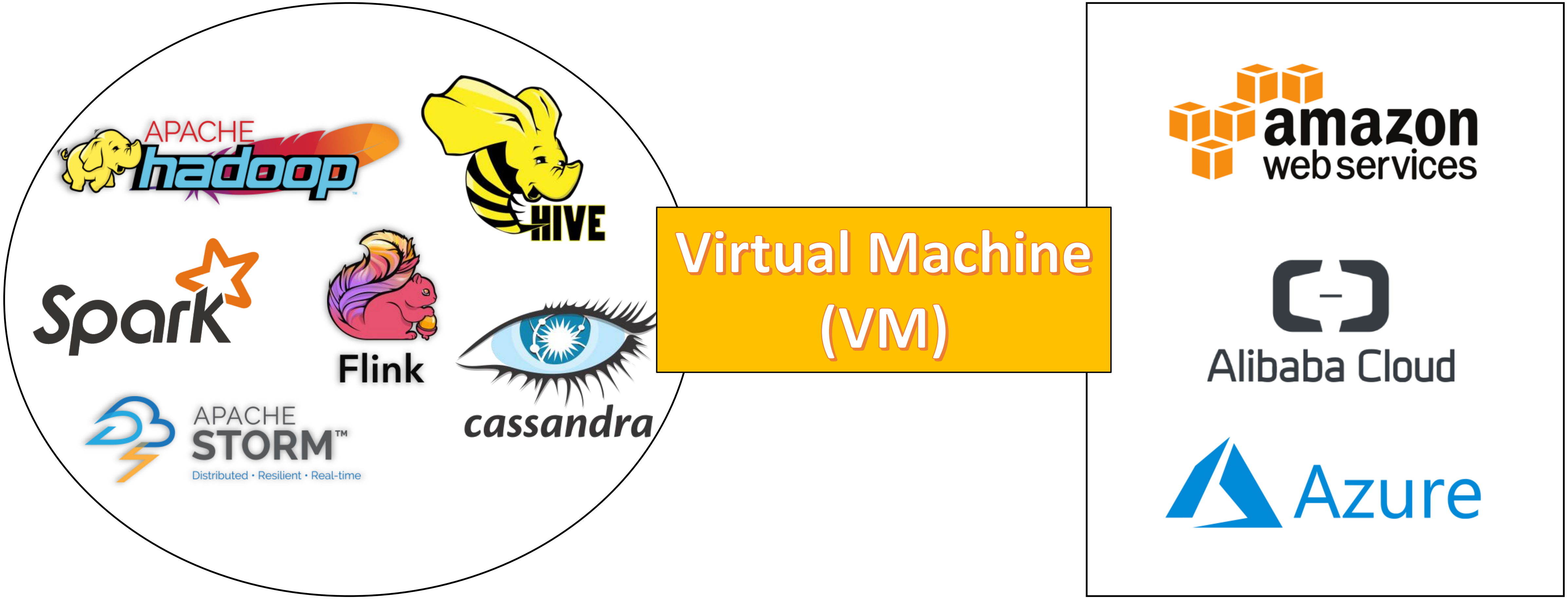
AUGUST 9-12, 2021

Best VM Selection for Big Data Applications across Multiple Frameworks by Transfer Learning

Yuewen Wu[◊], Heng Wu [◊], Yuanjia Xu [◊+], Yi Hu [◊+],
Wenbo Zhang [◊⁂], Hua Zhong [◊⁂], Tao Huang [◊⁂]

[◊] ISCAS, [+]UCAS, [⁂] SKLCS

Dozens of big data frameworks are available in clouds today



Selecting the best VM for multiple frameworks is challenging

Dozens of big data frameworks

Compute Optimized
- C4.large
- C5.xlarge
- C4n.2xlarge

… …

… …

Memory Optim
- R4.xlarge
- R5.4xlarge
- X1.8xlarge

General Purpose
- T3.medium
- M5.4xlarge

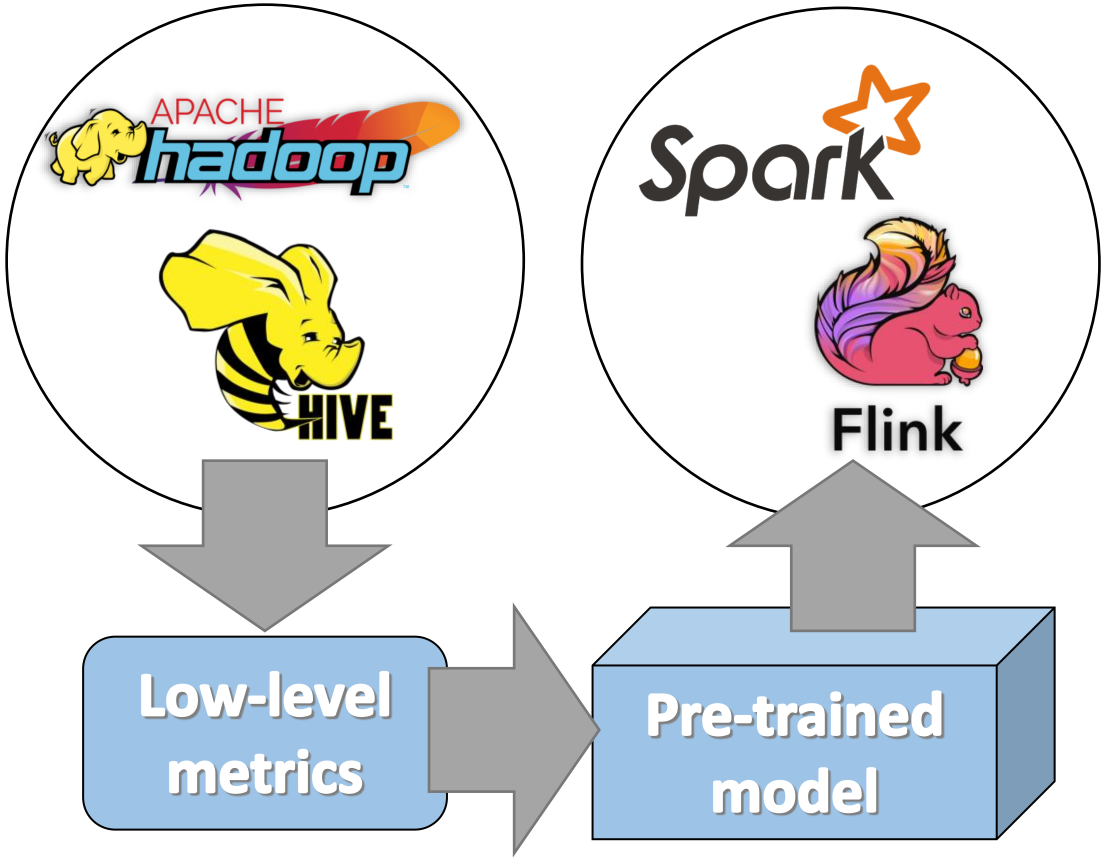
… …

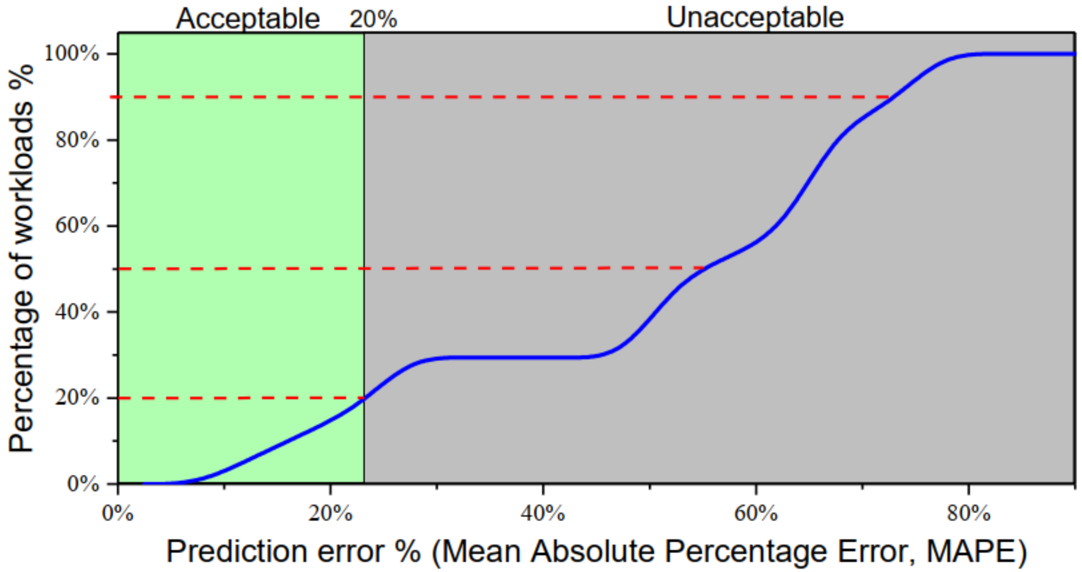
Hundreds of VM types

For a given app and its framework, which VM is the *best* to achieve better performance or lower budget?

Limitation of existing machine learning approaches

1. Reusing pre-trained model based on low-level metrics (e.g., CPU utilization)

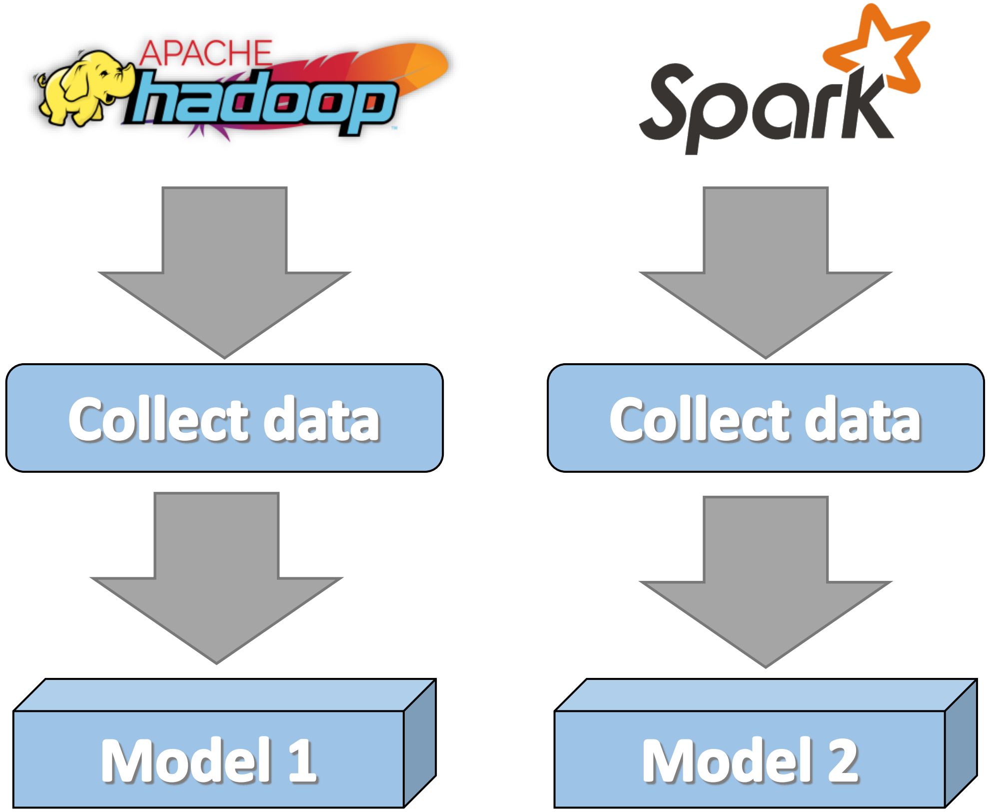


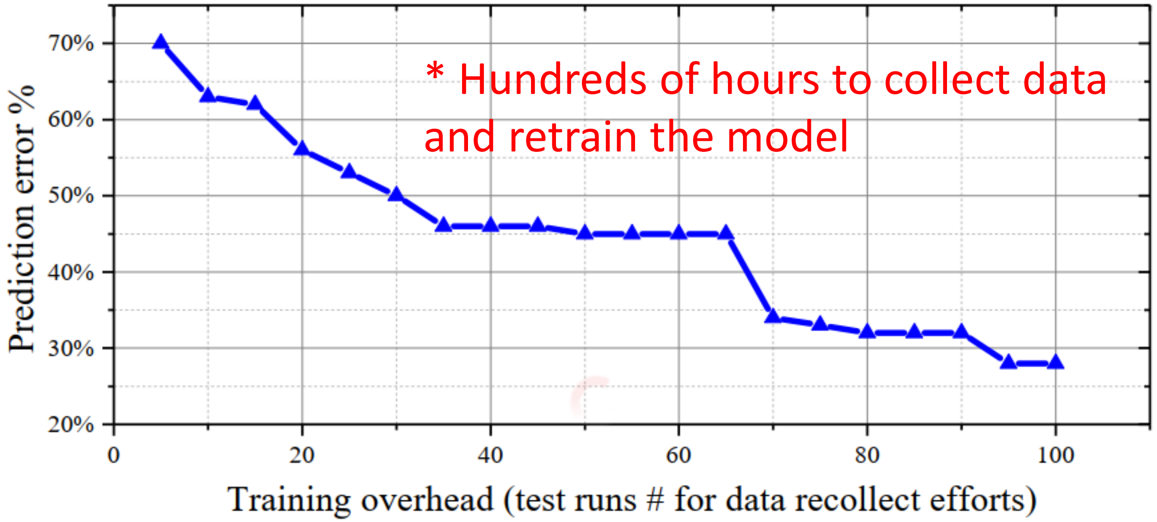


High prediction error for new frameworks

Limitation of existing machine learning approaches

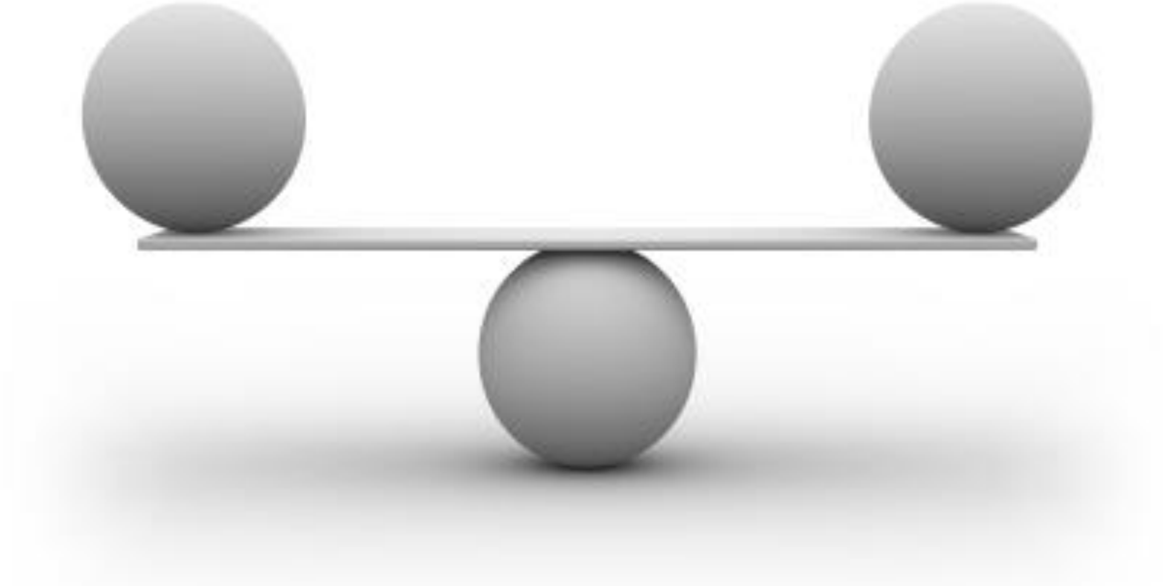
2. Training models for each framework





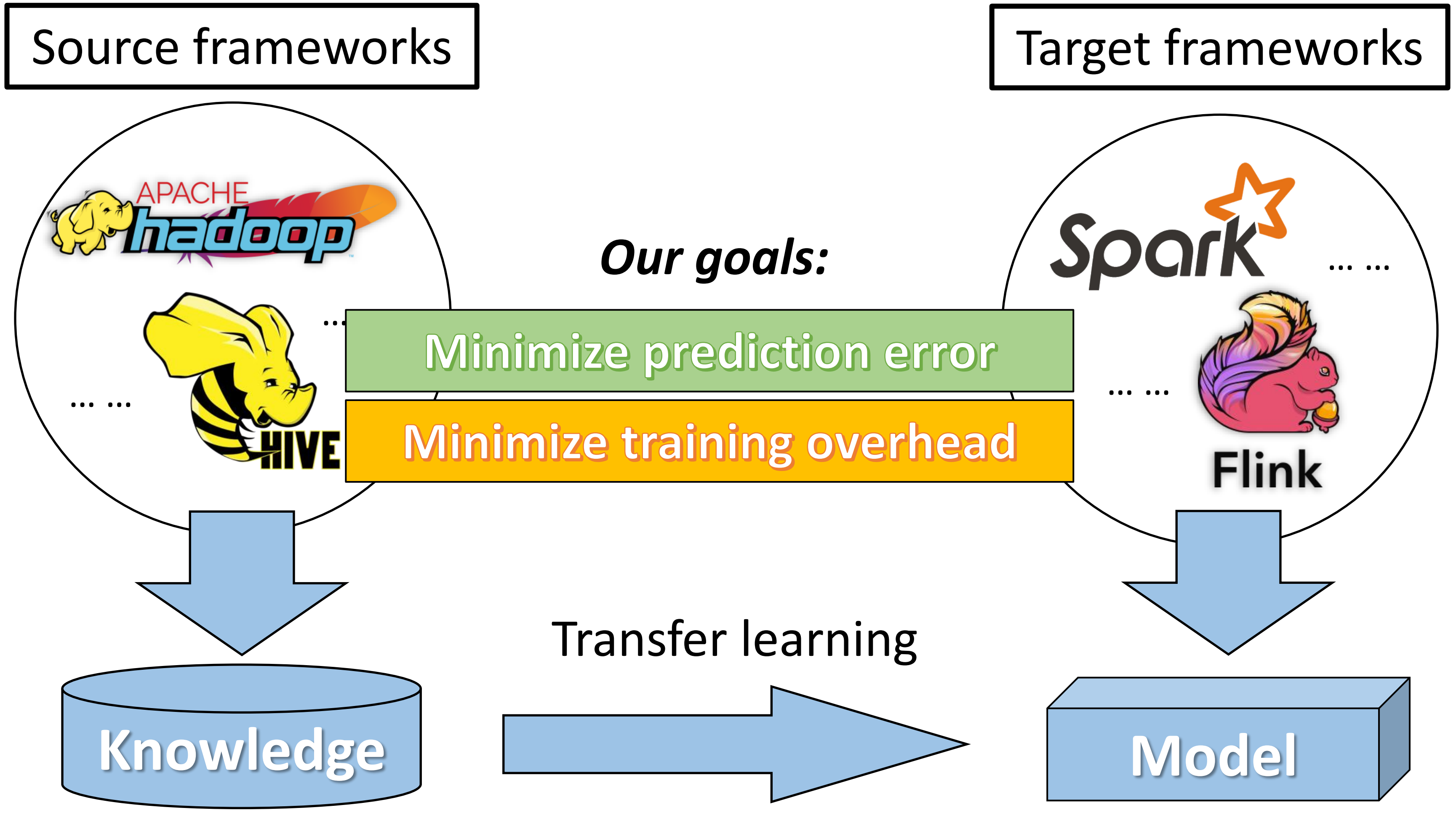
High training overhead for each framework

Questions?

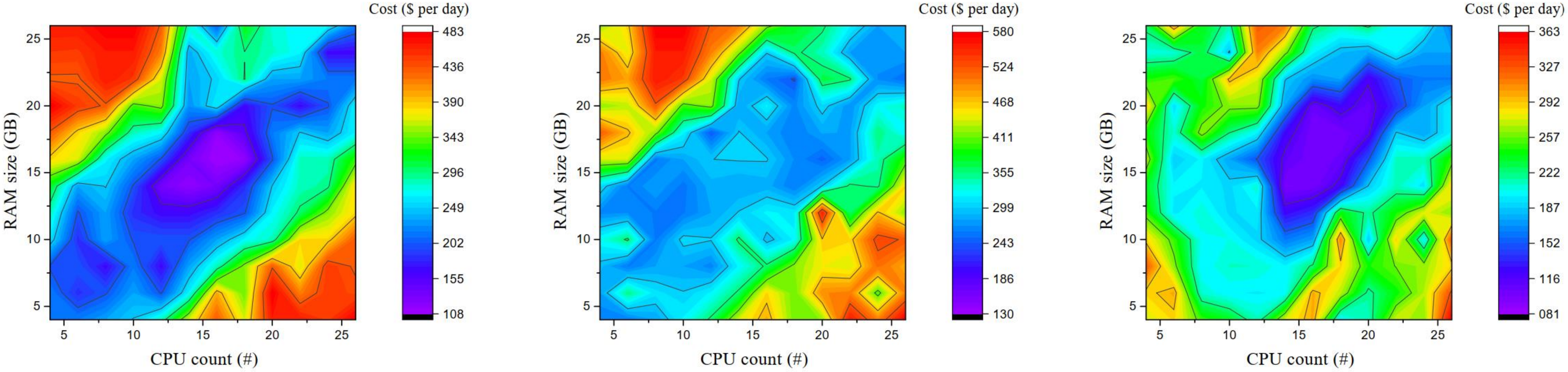


How to balance prediction error and training overhead for multiple frameworks?

Vesta: reusing knowledge by transfer learning



Our core finding: knowledge across frameworks

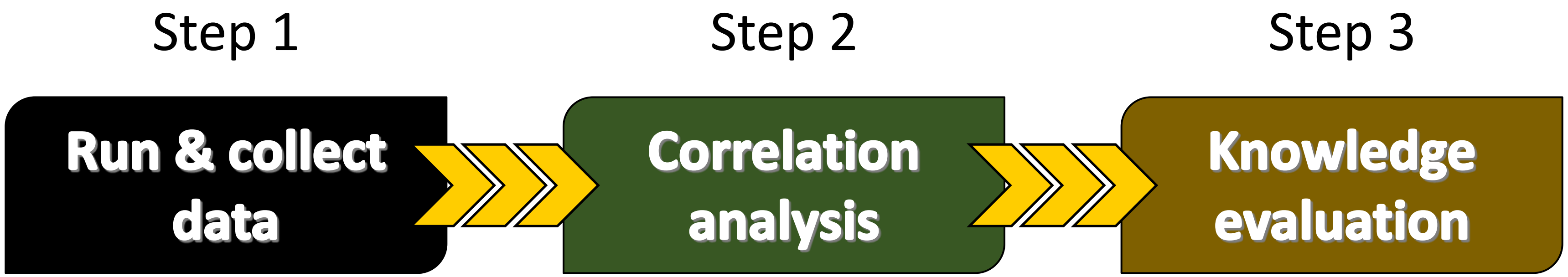


(a) The result of Hadoop TeraSort. **(b) The result of Hive Aggregation.** **(c) The result of Spark PageRank.**

Low-level metrics have **high-level similarities** (aka ***knowledge***) across frameworks (the blue areas in heat maps)

Vesta: abstract knowledge

- Abstracting knowledge by a large-scale evaluation, it contains:
  - 3 widely used big data frameworks: Hadoop, Hive and Spark
  - 30 benchmark workloads (HiBench@Intel and BigDataBench@ICT)
  - 120 VM types (x86-arch) on Amazon EC2
  - 20 resource and execution metrics (e.g., CPU rate, number of tasks)



Vesta: abstract knowledge

- Knowledge evaluation: abstract most valuable knowledge
  - Top 10 knowledge after evaluation (valid for current dataset)

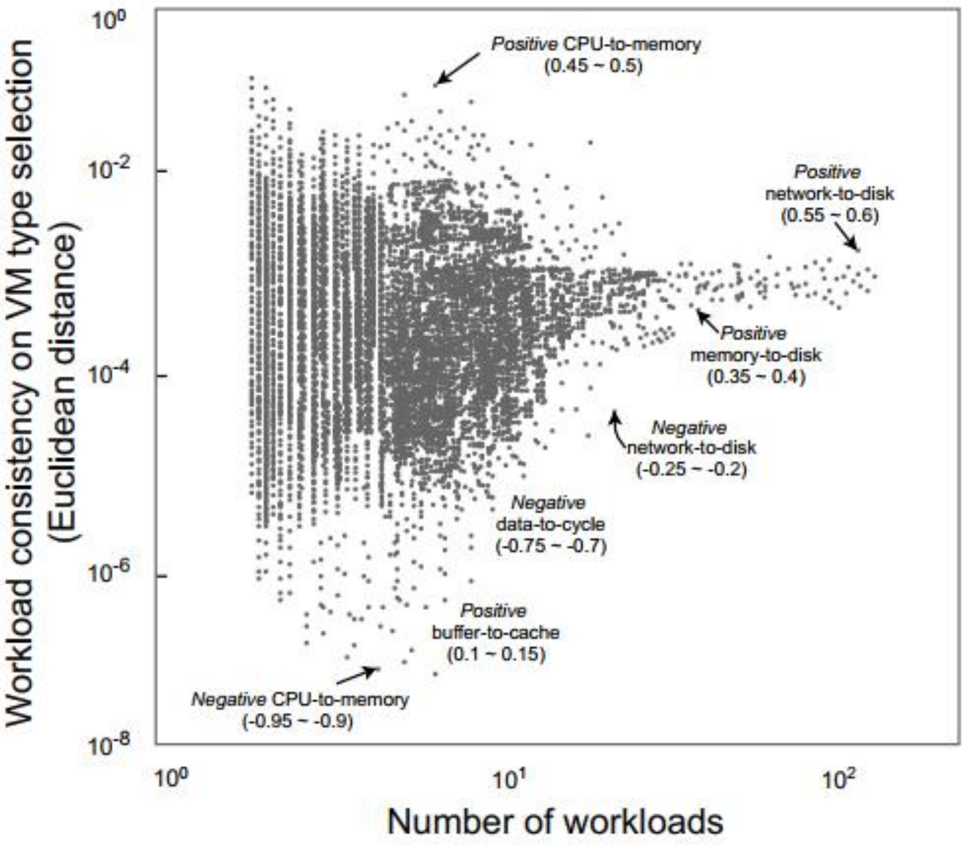
Table 1: High-level similarities (correlations) across frameworks.

Correlations	Description
Resource metrics	
CPU-to-memory	A positive [★] correlation probably denotes a heavy computational workload, so it can infer to larger CPU and memory sizes in VM types. A negative correlation means the opposite side.
memory-to-disk	A negative [≀] correlation can represent relatively small data size, and can infer to lower VM memory size and disk bandwidth in VM types. A positive correlation represents the opposite side.
disk-to-network	A positive correlation reveals that the workload exchange data frequently to facilitate remote data storage capabilities, and can infer to higher disk and network bandwidths in VM types. A negative correlation means the opposite side.
buffer-to-cache	A positive correlation reveals that *buffer cache* and *page cache* are two critical memory caches in this workload, and can infer to larger buffer and cache capabilities. A negative correlation means the opposite side.
CPU-to-network	A negative correlation probably means that there are lots of data synchronizations in the workload, and can infer to higher network bandwidths. A negative correlation means the opposite side.
Execution metrics	
iteration-to-parallelism	A positive correlation means that the workload prefers running in a "thin" cluster (more iterations), and a negative correlation means that it prefers running in a "fat" cluster (more parallelism). It can infer to the choice of the number of VMs.
data-to-computation	A positive correlation reveals that the workload has lots of *computation* phases. A negative correlation means the opposite side. It can infer to the choice of CPU cores and CPU rate.
data-to-cycle	A positive correlation means that it may be a data-intensive workload or a compute-intensive workload. A negative correlation means the opposite side. It can infer to the choice of RAM size and RAM type.
disk-to-synchronization	A positive correlation reveals that the workload exchanges data frequently. A negative correlation means the opposite side. It can infer to the choice of disk bandwidth and disk size.
network-to-synchronization	A positive correlation means that the workload transfers data frequently. A negative correlation means the opposite side. It can infer to the choice of network bandwidth.

Note: [★] The positive correlation reveals the relationship between two variables in which both variables move in tandem — that is, in the same direction. [≀] The negative correlation reveals one variable decreases as the other variable increases.

Vesta: abstract knowledge

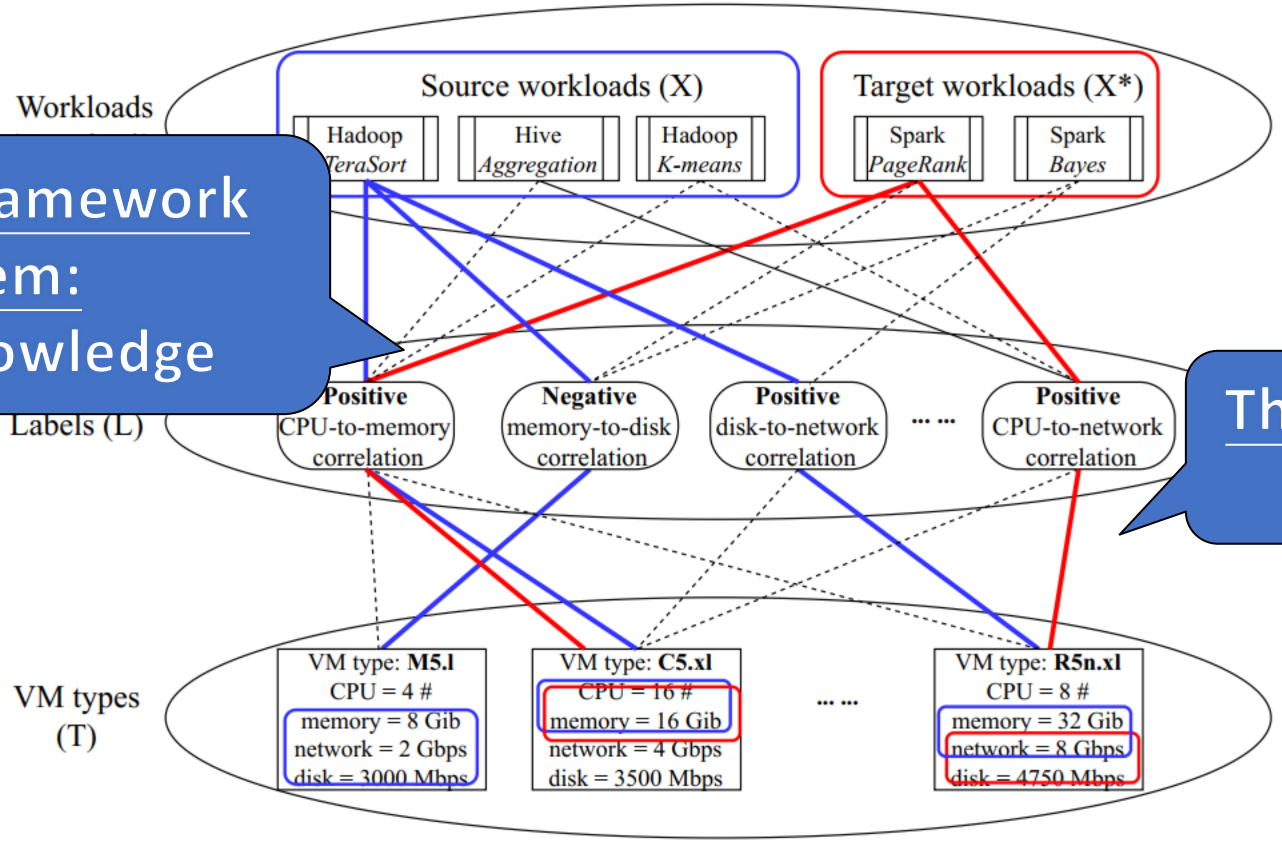
- Knowledge evaluation: data distribution analysis
  - We use scoring mechanism (X-axis for *popularity*, Y-axis for *consistency*) to evaluate the importance of knowledge



We train a K-means model for source frameworks

Vesta: represent knowledge

- Representing knowledge in a two-layer bipartite graph
  - Blue boxes and edges: source workloads and frameworks
  - Red boxes and edges: target workloads and frameworks

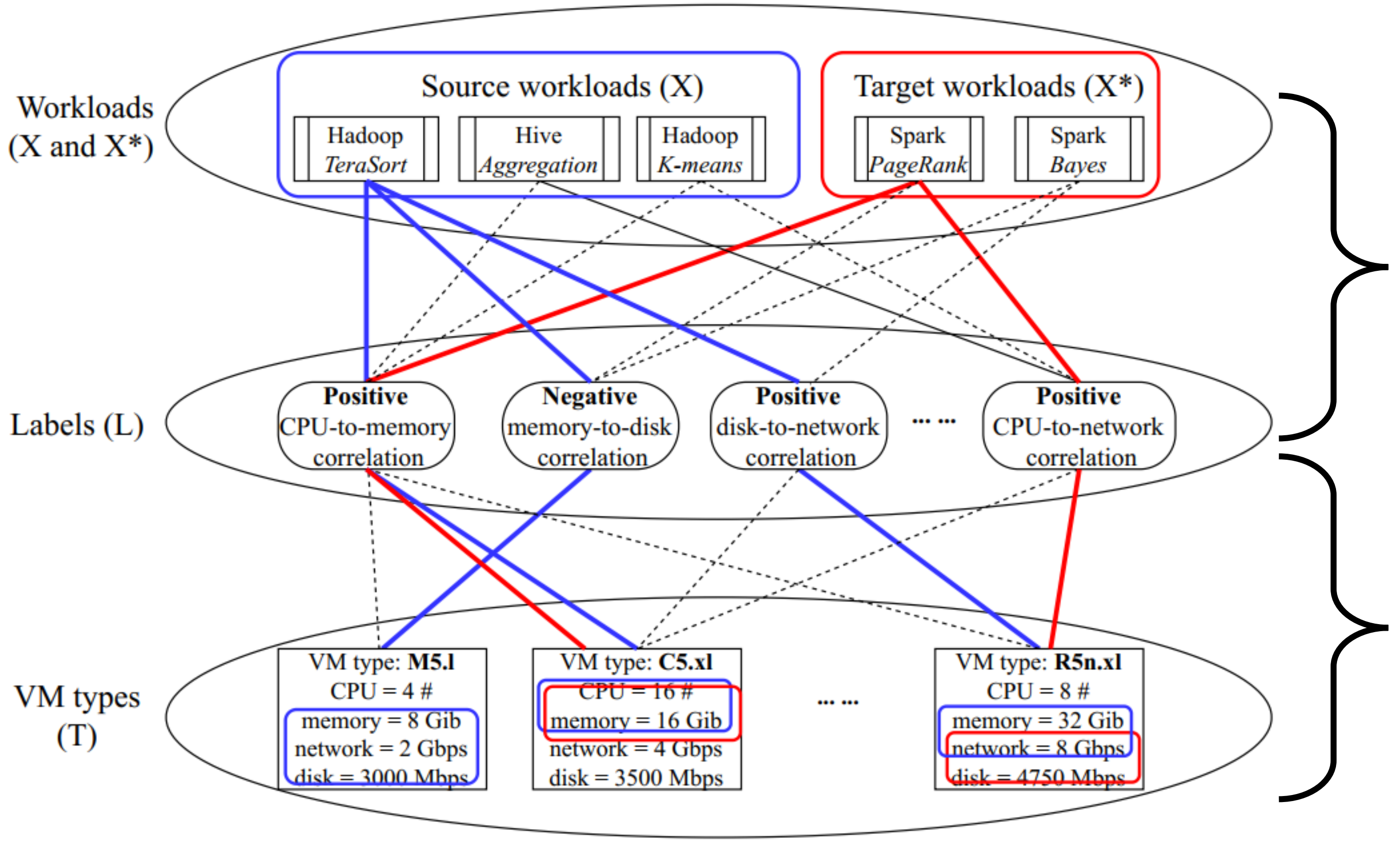


<u>The multi-framework problem:</u>
sharing knowledge

<u>The VM selection problem:</u>
reusing knowledge

Vesta: reuse knowledge

- Step 1: Decomposing the two-layer bipartite graph into matrices



Workload-label relationship

source

target

$U = X^* L_1^T$

$U^* = X^* L_1^T$

Label-VM relationship

$V = T L_2^T$

Vesta: reuse knowledge

- Step 2: Solving the data sparsity problem for target matrix *U**
  - Applying *Collective Matrix Factorization* (CMF) algorithm to select data

$$\min_{U,F,U^*} \lambda \| \boxed{U^*} - \boxed{U} \|_F^2 + (1-\lambda) \| \boxed{U^*} - V \|_F^2 + R(U,V,U^*)$$

- Step 3: Searching the *best* VM for target workloads and frameworks
  - Employing *Stochastic Gradient Descent* (SGD) algorithm to search the best VM
  - Reusing knowledge (data) from source workloads and frameworks
  - Training the model incrementally for target frameworks

Evaluation setup

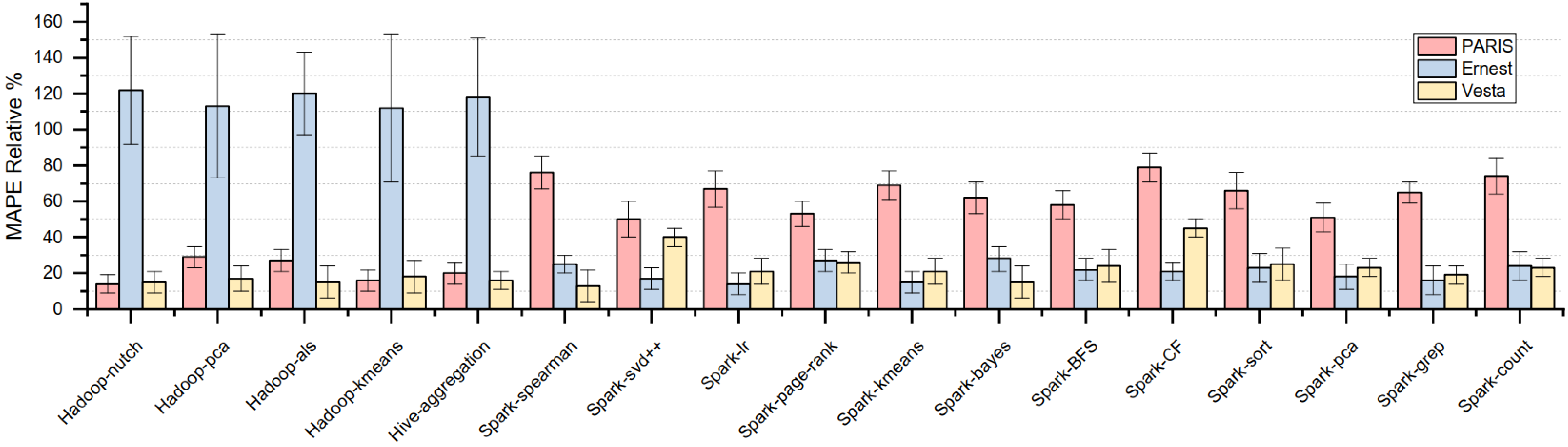
3 frameworks & 30 benchmark workloads
(Source: *Hadoop, Hive*) (Target: *Spark*)

120 VM types from Amazon EC2

Alternative solutions:
- PARIS@SoCC' 17
- Ernest@NSDI' 16

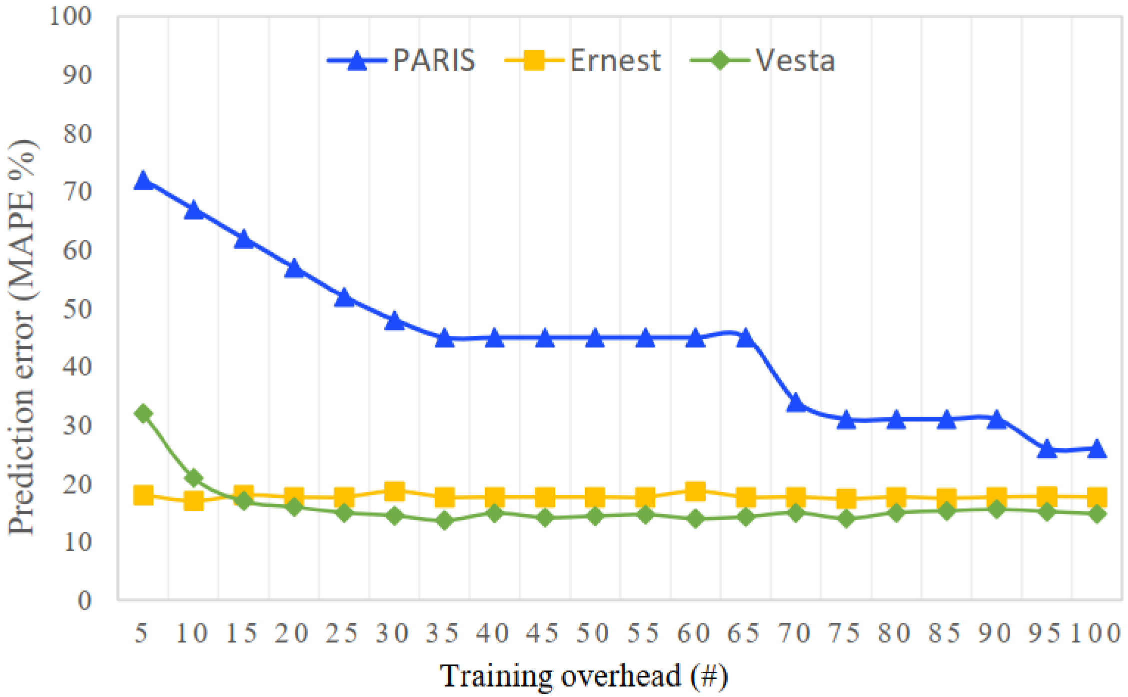
Experiment metrics:
- Performance improvement: prediction error
- Training overhead: number of runs
- Practical metrics: execution time & budget

Performance improvement for multiple frameworks



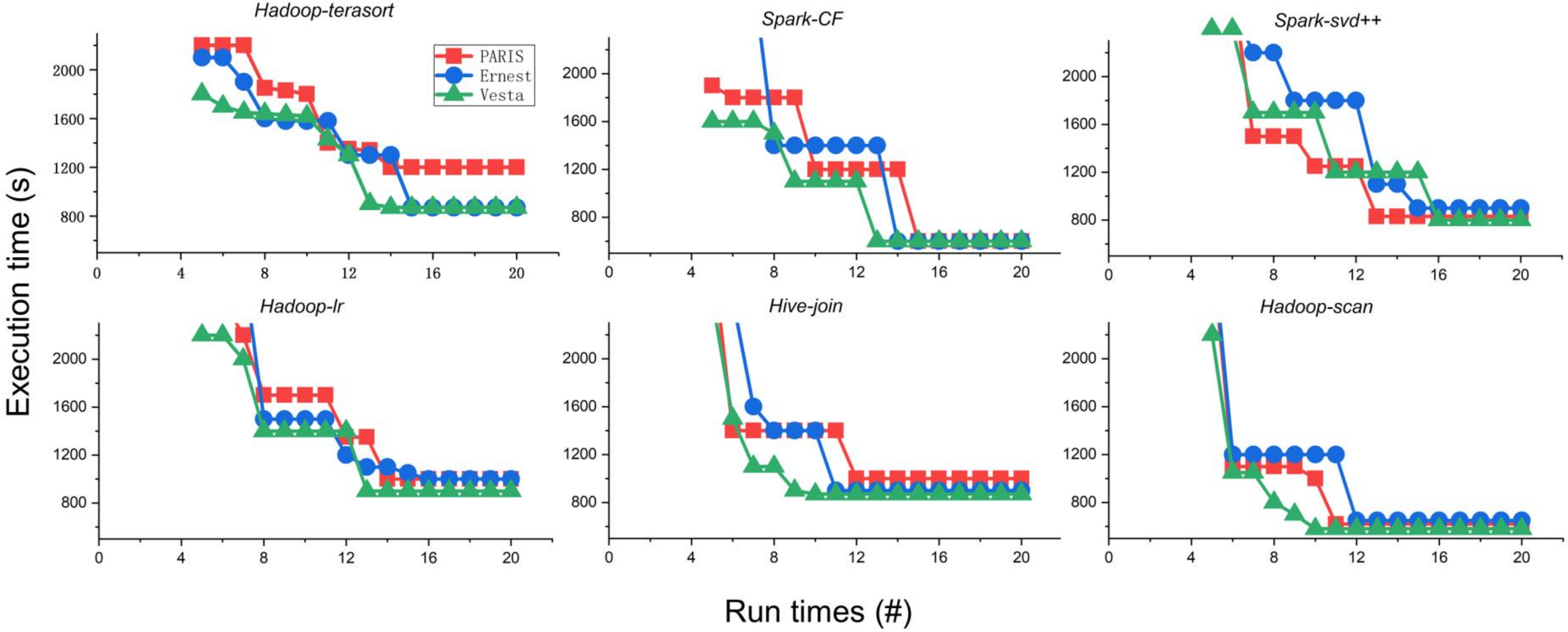
Vesta can reduce up to 51% prediction error, that is –
can improve up to 51% performance due to reuse knowledge

Training overhead improvement for multiple frameworks



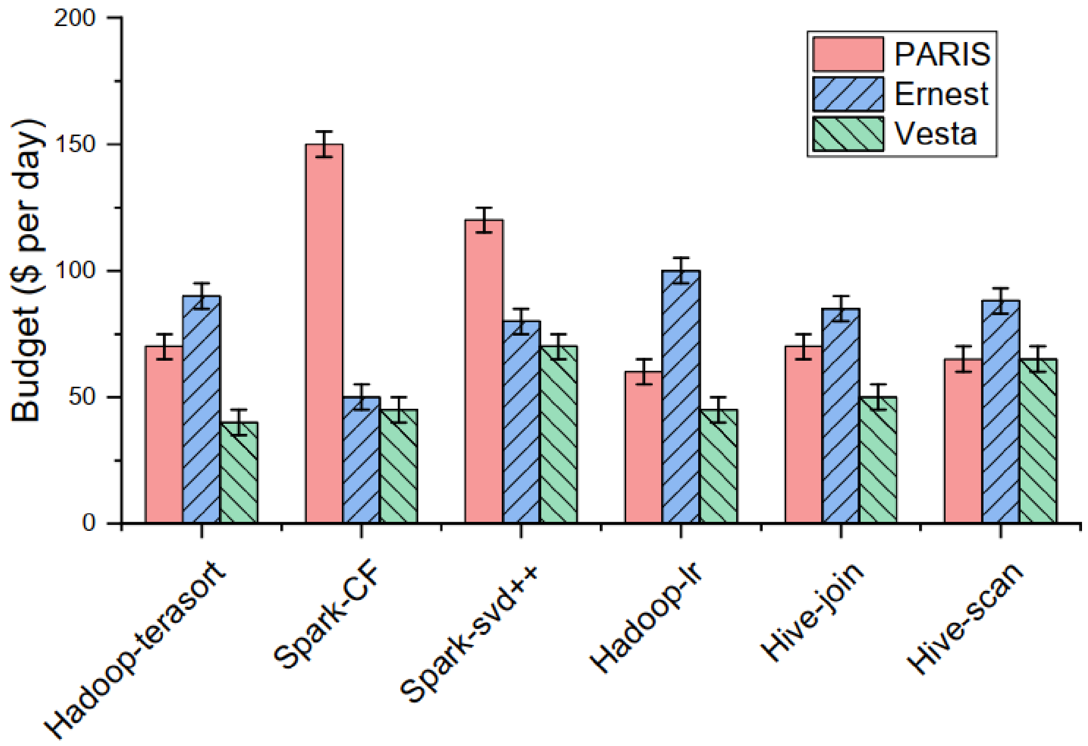
Vesta can reduce up to 85% training overhead due to transfer learning

Optimizing the execution time of running application



Vesta can find VMs with shorter execution time in 5 of 6 applications

Optimizing the budget of renting VMs



Vesta performs better or at least comparable budgets

Summary

- Vesta selects the *best* VM effectively for multi-framework applications by transfer learning.
- Vesta **observes knowledge** (high-level similarities) across frameworks.
- Vesta **abstracts knowledge** by a large scale evaluation on Amazon EC2.
- Vesta **represents and reuses knowledge** through a combination of technologies.
- Vesta can **improve application performance** up to 51% while **reducing** 85% **training overhead.**
- Vesta can easily adapt to big data frameworks that follow the *Bulk Synchronous Parallelism* (BSP) design. In the future, we want to extend Vesta to support deep learning applications, such as TensorFlow and PyTorch.

Take a look of our paper for more details

Thanks!

*Any **questions** ?*

You can find me at

- wuyuewen@otcaix.iscas.ac.cn