Tahoe: Tree Structure-Aware High Performance Inference Engine for Decision Tree Ensemble on GPU

Zhen Xie*, Wenqian Dong*, Jiawen Liu*, Hang Liu#, Dong Li*

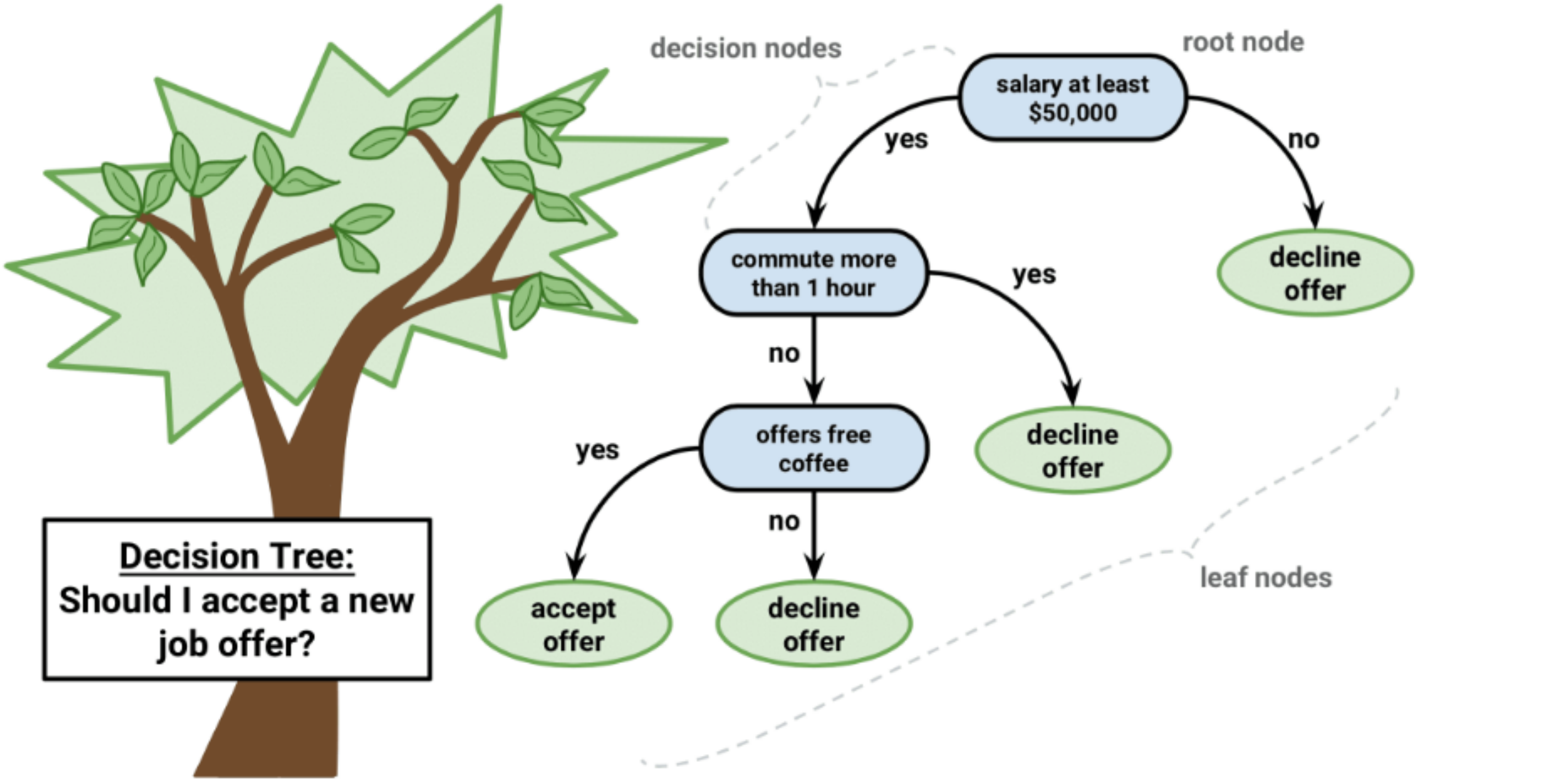
*Parallel Architecture, System, and Algorithm Lab (PASA Lab)

*University of California, Merced

#Stevens Institute of Technology

Background for Decision Tree Inference

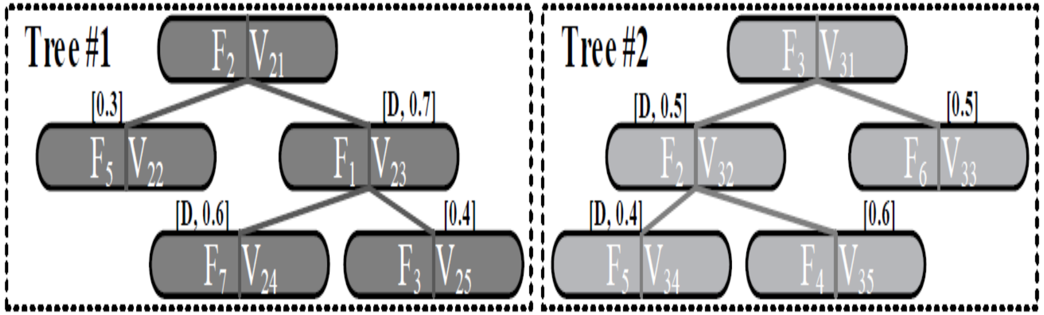
What is the decision tree?



Background for Decision Tree Inference

How is the decision tree inference used?

Forest (that includes multiple trees)



High Parallelism

→

High Bandwidth

GPUs

facebook

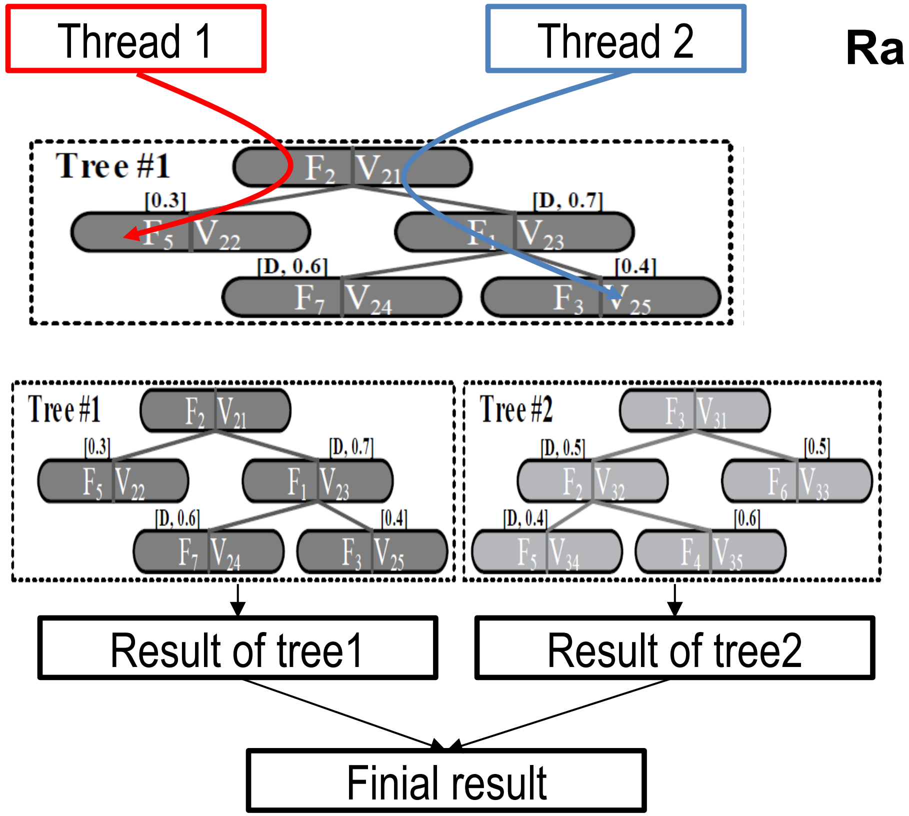
Microsoft News

RAPIDS

Accelerated with NVIDIA

Motivation

We found three performance problems when traversing a forest on GPU



Random memory accesses → **Uncoalesced Memory Accesses**

Load imbalance → **Thread Idling**

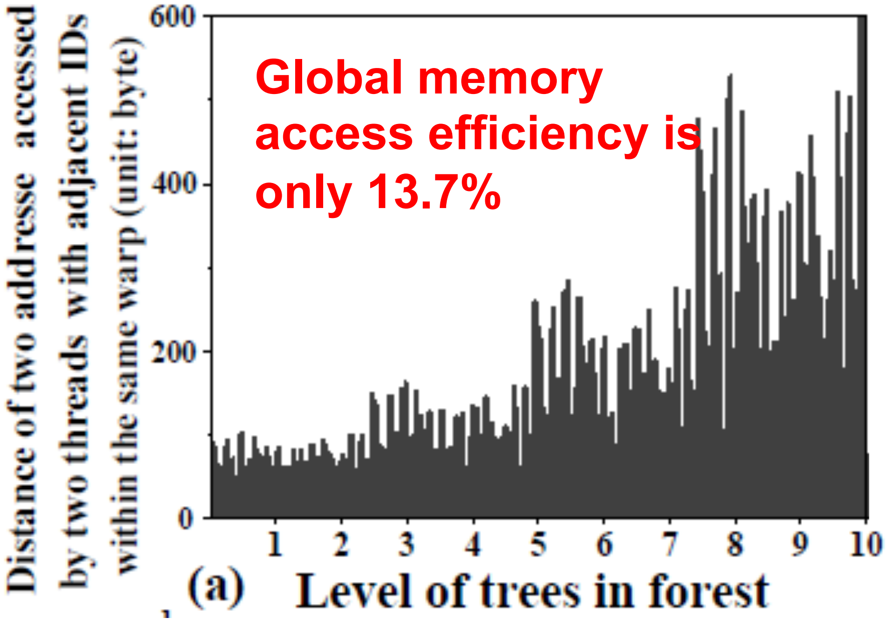
Reduction operation → **High Overhead**

Motivation

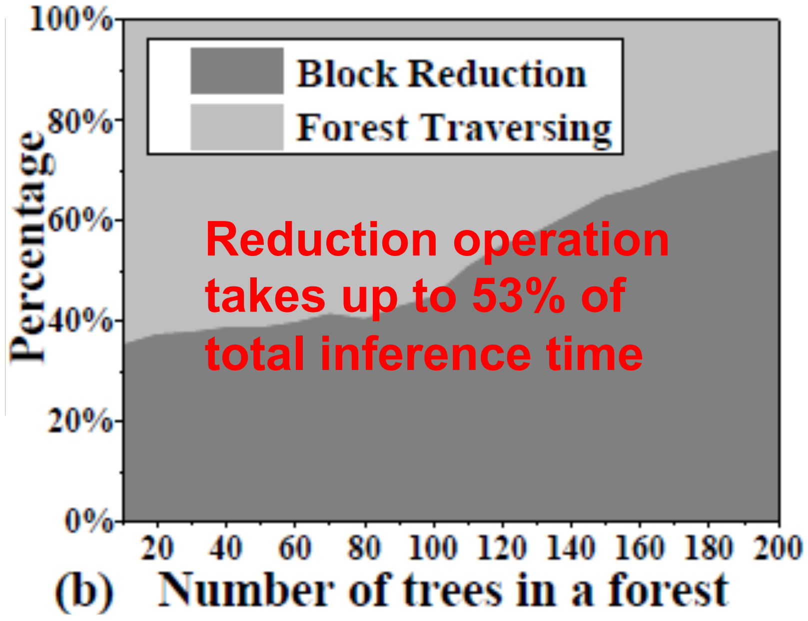
Quantification of Performance Problems

Test decision tree model: a random forest trained by XGBOOST on Higgs dataset.

The forest has 120 trees, and the maximum depth of each tree is 10.



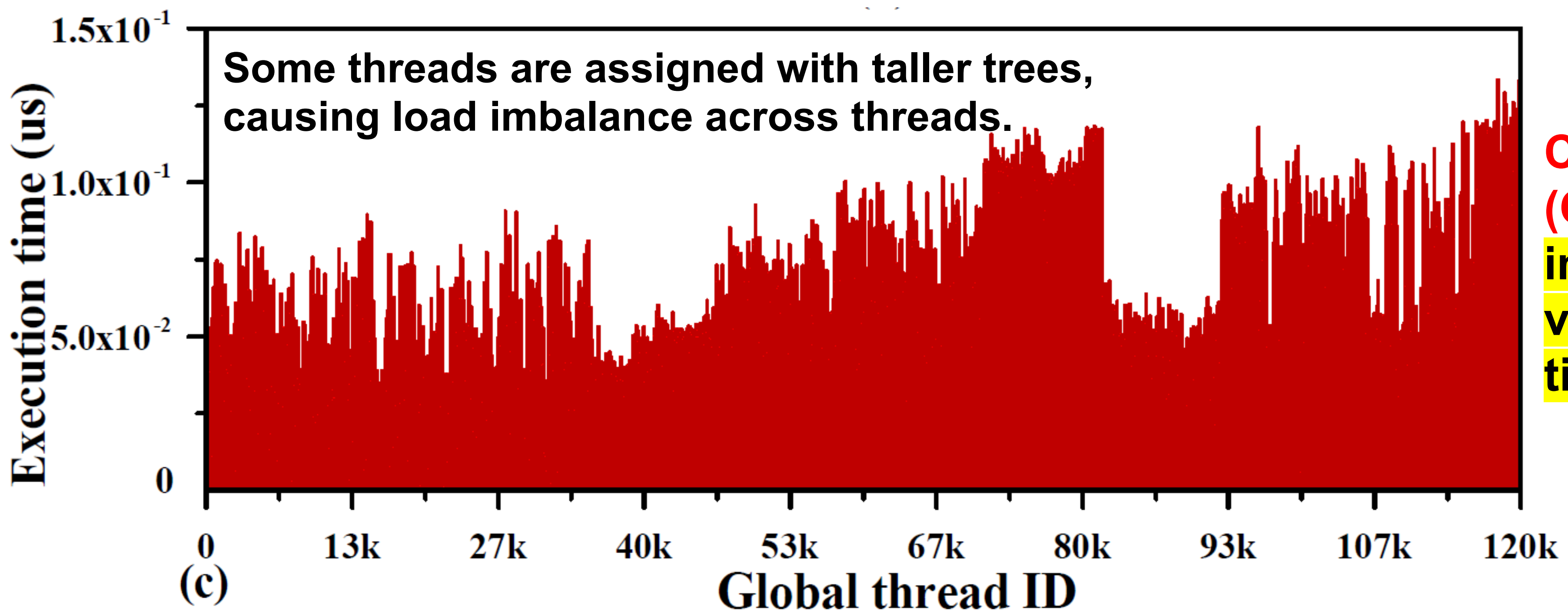
Uncoalesced memory accesses



High reduction overhead

Motivation

Quantification of Performance Problems



Load imbalance across threads.

Coefficient of variation (CV) =49.1%, indicating a large variance in execution time across threads.

Tahoe Framework Overview

Three solutions for solving the three performance problems

Adaptive Forest Format

---Probability-based node rearrangement → **Uncoalesced Memory Access**

---Similarity-based tree rearrangement → **Load Imbalance**

Multiple Inference Strategies to Adapt to Various Tree Topologies

---Direct strategy

---Shared forest strategy → **High Reduction Overhead**

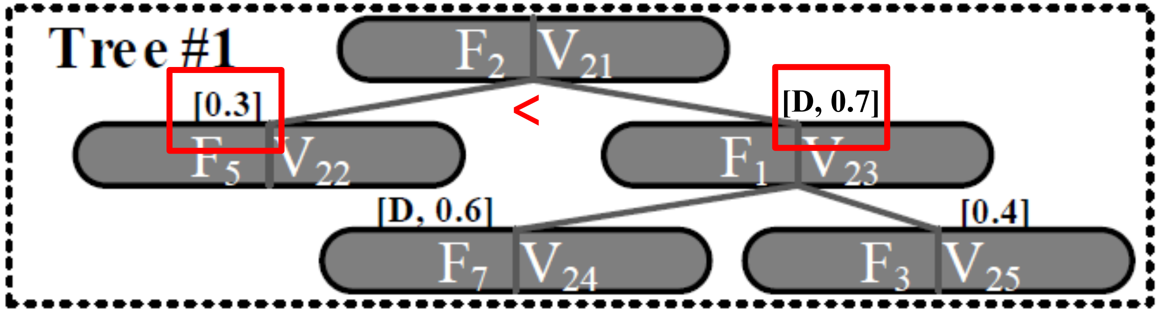
---Splitting shared forest strategy

Performance Modeling to Choose Optimal Inference Strategy → **Human Efforts**

Adaptive Forest Format

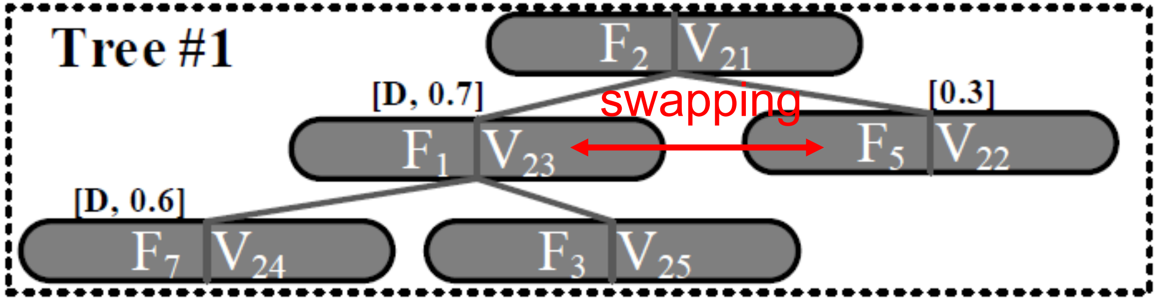
Probability-based node rearrangement

Before node rearrangement



The right child node ($V23$) has higher possibility to be visited than the left child node ($V22$).

After node rearrangement



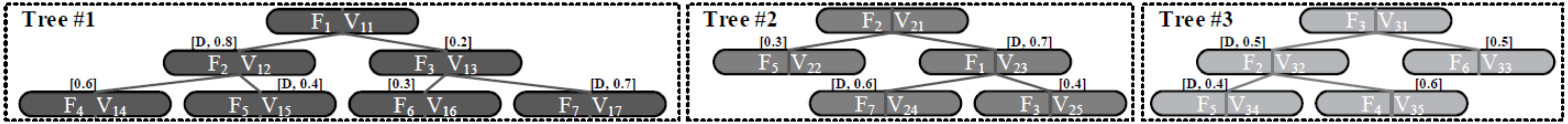
The two children nodes are swapped.

Adaptive Forest Format

Similarity-based tree rearrangement

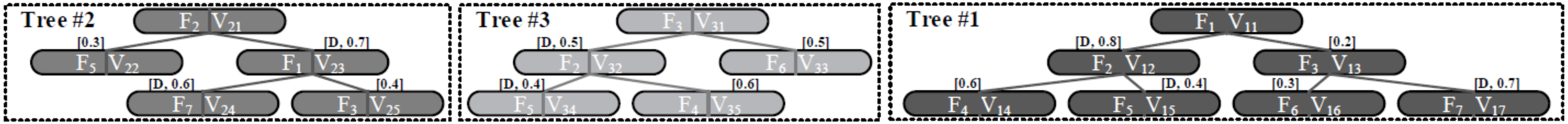
We claim two trees are *similar*, when the two trees tend to be traversed using the similar paths

Before tree rearrangement



[Tree1, Tree2, Tree3]

After tree rearrangement



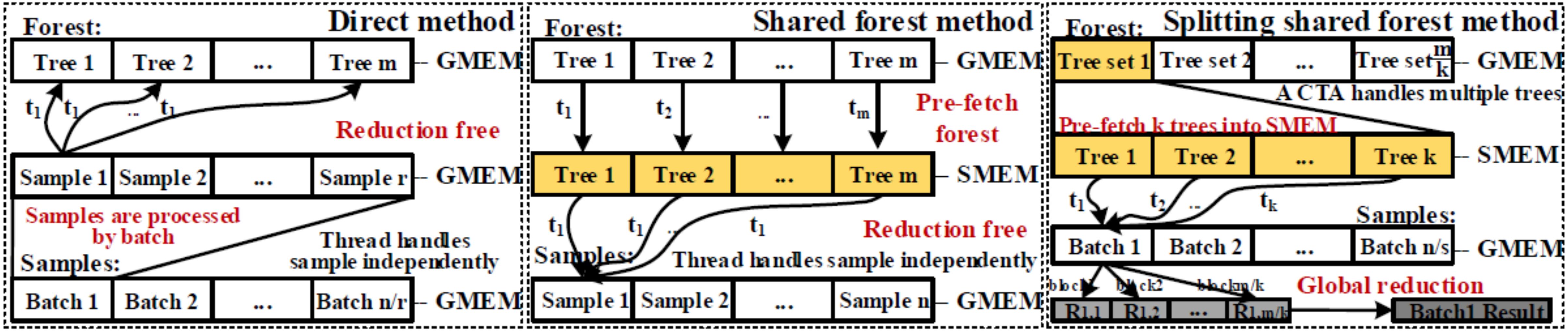
[Tree2, Tree3, Tree1]

Similarity of [T1, T2] is 0.14
Similarity of [T2, T3] is 0.75
Similarity of [T1, T3] is 0.59

Design of Inference Strategies

- Introduce multiple Inference strategies to avoid reduction and make best use of shared memory
- How should we place input samples and trees into shared memory?

GMEM: global memory SMEM: shared memory

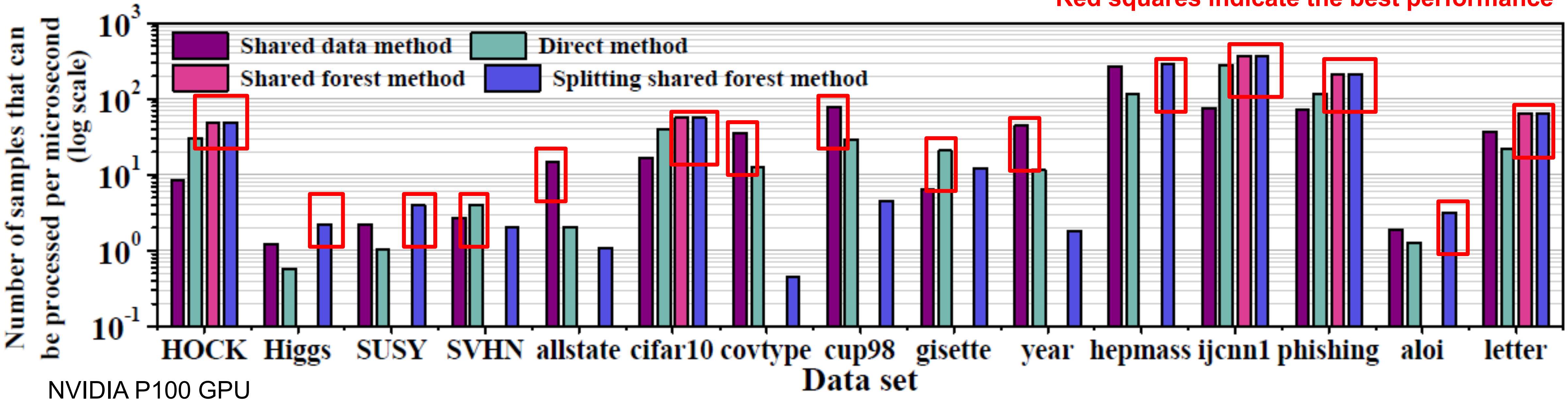


The depiction of different inference strategies. The usage of shared memory is highlighted in yellow.

Design of Inference Strategies

We study the performance of the three inference strategies proposed by us and one existing inference strategy

Red squares indicate the best performance



NVIDIA P100 GPU

Performance comparison of the four inference strategies using 15 datasets

Conclusion: No single strategies can perform best in all datasets with different batch sizes, datasets, and forests.

Performance Modeling for Tree Inference

Performance modeling is used to decide which inference strategy should be used for best performance.

Shared data

$$T_{SMEM} = \frac{S_{sample}}{BW\_W_{SMEM}} + \frac{D_{tree} * N_{trees} * S_{att}}{BW\_R_{SMEM}}$$

$$T_{GMEM} = \frac{S_{sample}}{BW\_R^{COA}_{GMEM}} + \frac{D_{tree} * N_{trees} * S_{node}}{(BW\_R^{COA}_{GMEM})/2}$$

Direct method

$$T_{GMEM} = \frac{D_{tree} * N_{trees} * S_{node}}{(BW\_R^{COA}_{GMEM})/2} + \frac{D_{tree} * N_{trees} * S_{att}}{BW\_R^{NCOA}_{GMEM}}$$

Shared forest

$$T_{SMEM} = \frac{D_{tree} * N_{trees} * S_{node}}{BW\_R_{SMEM}}$$

$$T_{GMEM} = \frac{D_{tree} * N_{trees} * S_{att}}{BW\_R^{NCOA}_{GMEM}}$$

Splitting shared forest

$$T_{SMEM} = \frac{N_{nodes} * N_{trees} * S_{node}}{BW\_W_{SMEM} * N\_batch} + \frac{D_{tree} * N_{trees} * S_{node}}{BW\_R_{SMEM}}$$

$$T_{GMEM} = \frac{N_{nodes} * N_{trees} * S_{node}}{BW\_R^{COA}_{GMEM} * N\_batch} + \frac{D_{tree} * N_{trees} * S_{att}}{B^{NCOA}_{GMEM}}$$

More details can be found in our paper.

- Performance modeling is used only once at each batch
- Performance modeling is highly lightweight
  - It takes up to 3% of an inference time

Evaluation

Platform

- A high-end server with 24 Intel Xeon E6-2760 v3 CPU cores running at 2.30GHz;
- Three generations of GPU
  - Tesla K80 (Kepler), Tesla P100 (Pascal) and Tesla V100 (Volta).

Input Datasets

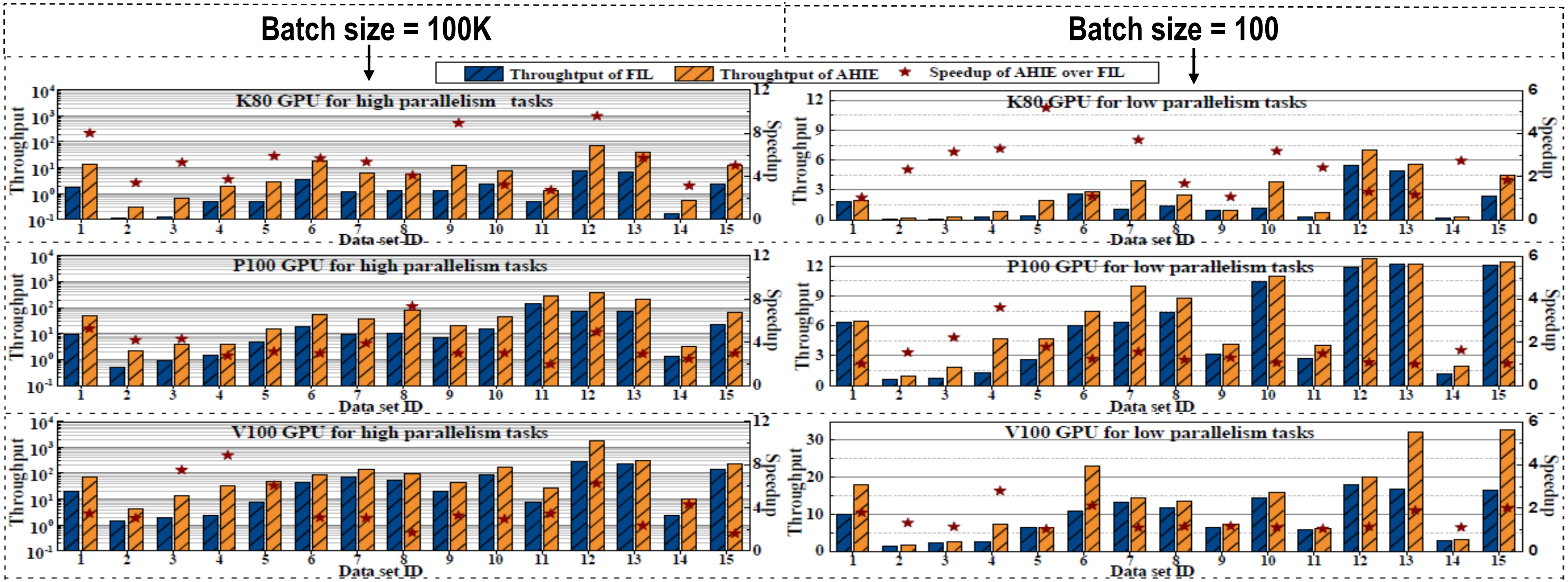
- 15 datasets from UCI repository and LIBSVM
- 70% of each dataset is used for training and 30% is used for inference.

Baseline (state-of-the-art an industry quality)

- A high-throughput tree library (FIL) in NVIDIA RAPIDS suite.

Evaluation

Overall Performance



- For the high parallelism task, Tahoe introduces **5.31x**, **3.67x** and **4.05x** speedup on average on three GPUs;
- For the low parallelism task, Tahoe introduces **2.34x**, **1.52x** and **1.45x** speedup on average on three GPUs.

Evaluation

Quantifying memory coalescence.

With Tahoe, the global memory read throughput is improved from **62.4 GB/s to 174.7 GB/s** on K80, **98.8 GB/s to 314.0 GB/s** on P100, and **112.4 GB/s to 378.5 GB/s** on V100.

Quantifying load imbalance.

GPUs	High parallelism tasks		Low parallelism tasks	
	A.C.V. of FIL	A.C.V. of Tahoe	A.C.V. of FIL	A.C.V. of Tahoe
K80	47.2%	13.1%	36.4%	10.8%
P100	51.3%	16.2%	42.9%	13.5%
V100	54.6%	15.9%	44.7%	12.5%

A bigger forest gets more performance benefit from load balancing.

Quantifying effectiveness of removing blockwise reduction.

Tahoe removes blockwise reduction for **27 cases from 45 cases**.

Conclusions

- We reveal three common performance problems in decision tree inferences
- We introduce Tahoe, an inference engine on GPU that considers the common paths of tree traversal and the similarity of tree topologies to enable high performance decision tree inference
- Tahoe largely outperforms an industry-quality inference engine
  - More than 3x speedup for high parallelism tasks on three generations of GPUs
  - More than 1.4x speedup for low parallelism tasks on three generations of GPUs

Thank you and questions?

Tahoe: Tree Structure-Aware High Performance Inference Engine for Decision Tree Ensemble on GPU
http://zhen-xie.com/papers/EuroSys'21.pdf

Zhen Xie zxie10@ucmerced.edu
Dong Li dli35@ucmerced.edu