Systems Support for Federated Computation

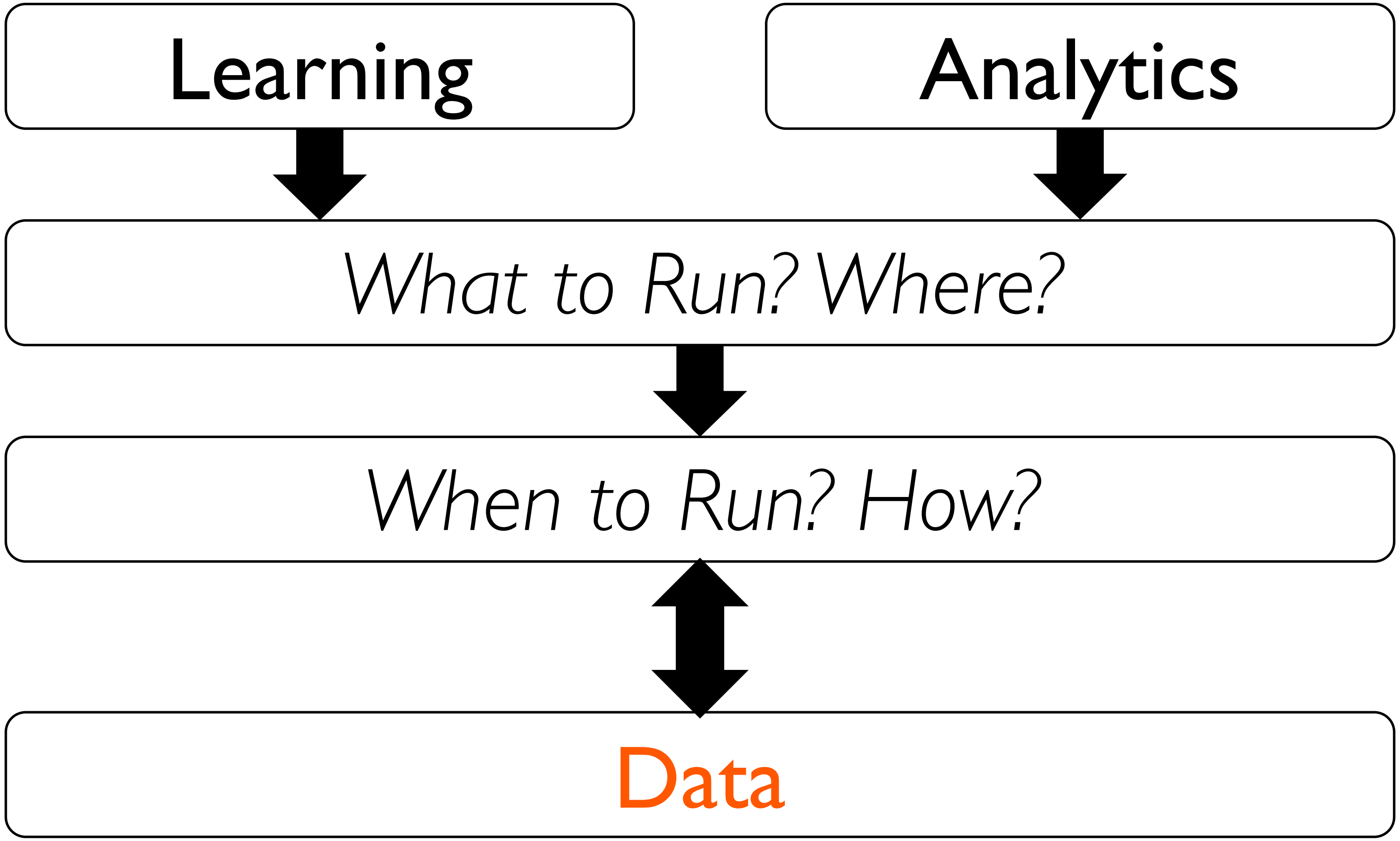
Mosharaf Chowdhury

November 2021

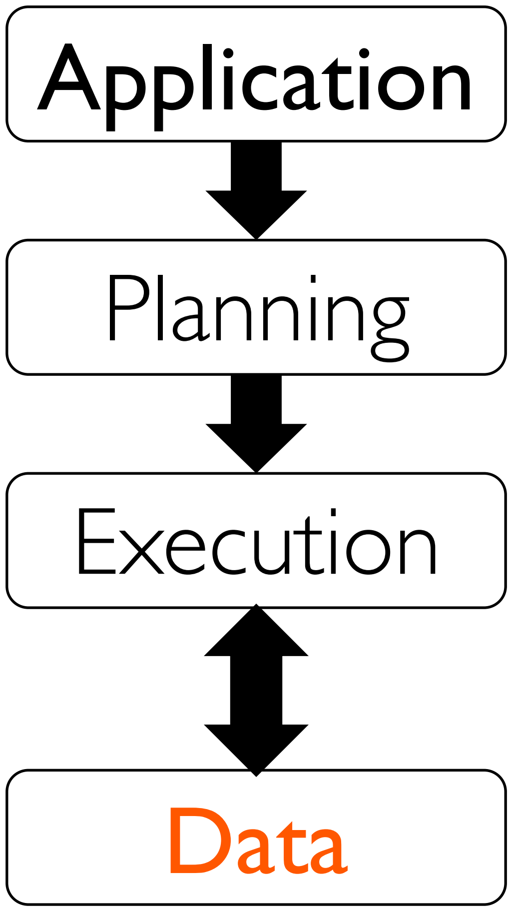
SymbioticLab

UNIVERSITY OF MICHIGAN

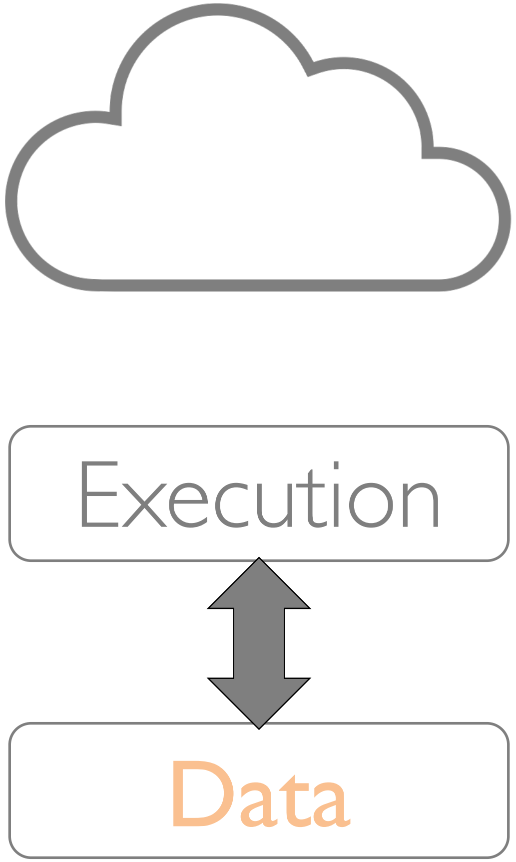
A Systems View of Learning and Analytics



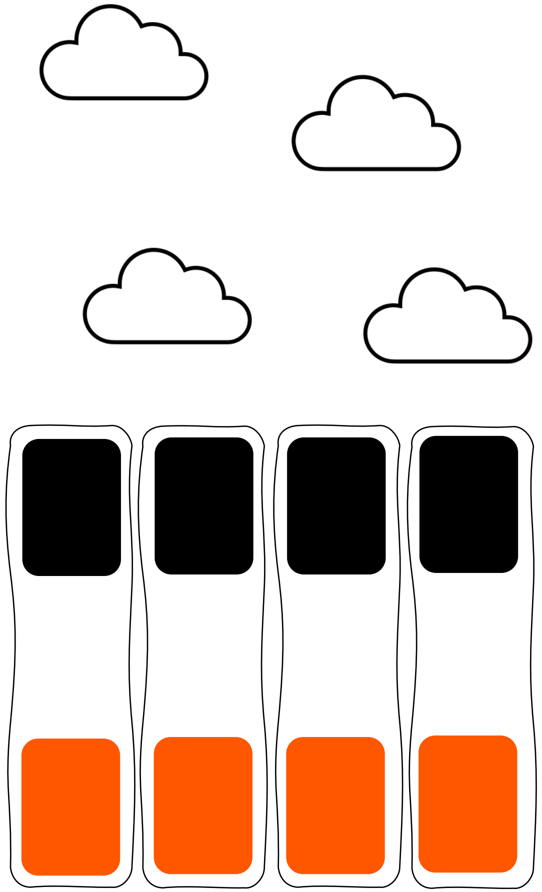
A Systems View of Learning and Analytics



Cloud L&A



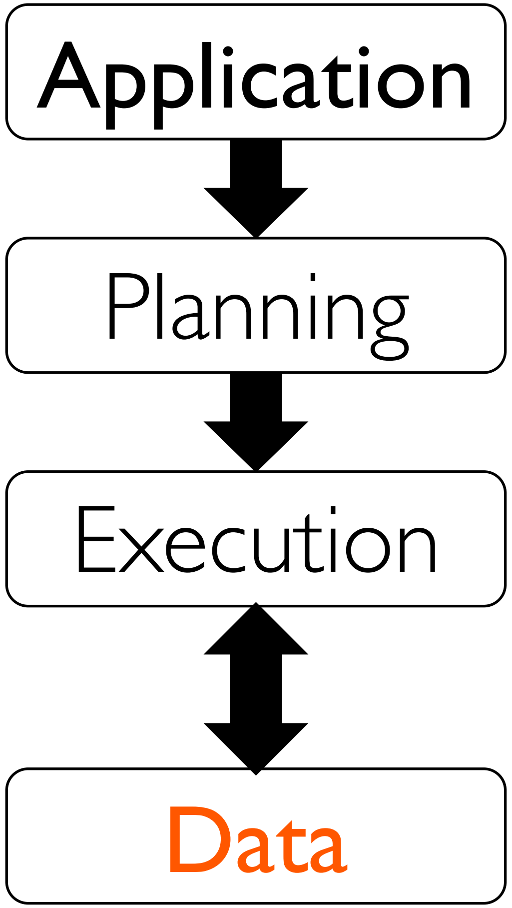
Cross-Silo FL&A



Cross-Device FL&A

Network is King!

1. Low bandwidth
2. High latency
3. Asymmetric topology
4. Dynamic variations



CellScope@MobiCom'18

Fed-ensemble@arXiv'21

Auxo

QOOP@OSDI'18

Oort@OSDI'21

NOCS@SPAA'19

Terra@arXiv'19

Sol@NSDI'20

Flamingo

FedScale@arXiv'21

Pando@NSDI'20

Application

Planning

Execution

Data

Oort: Cross-Device FL&A

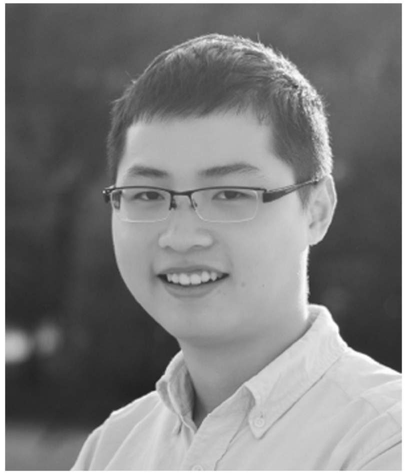
Sol: Cross-Silo FL&A

FedScale.ai

https://github.com/ SymbioticLab

Sol

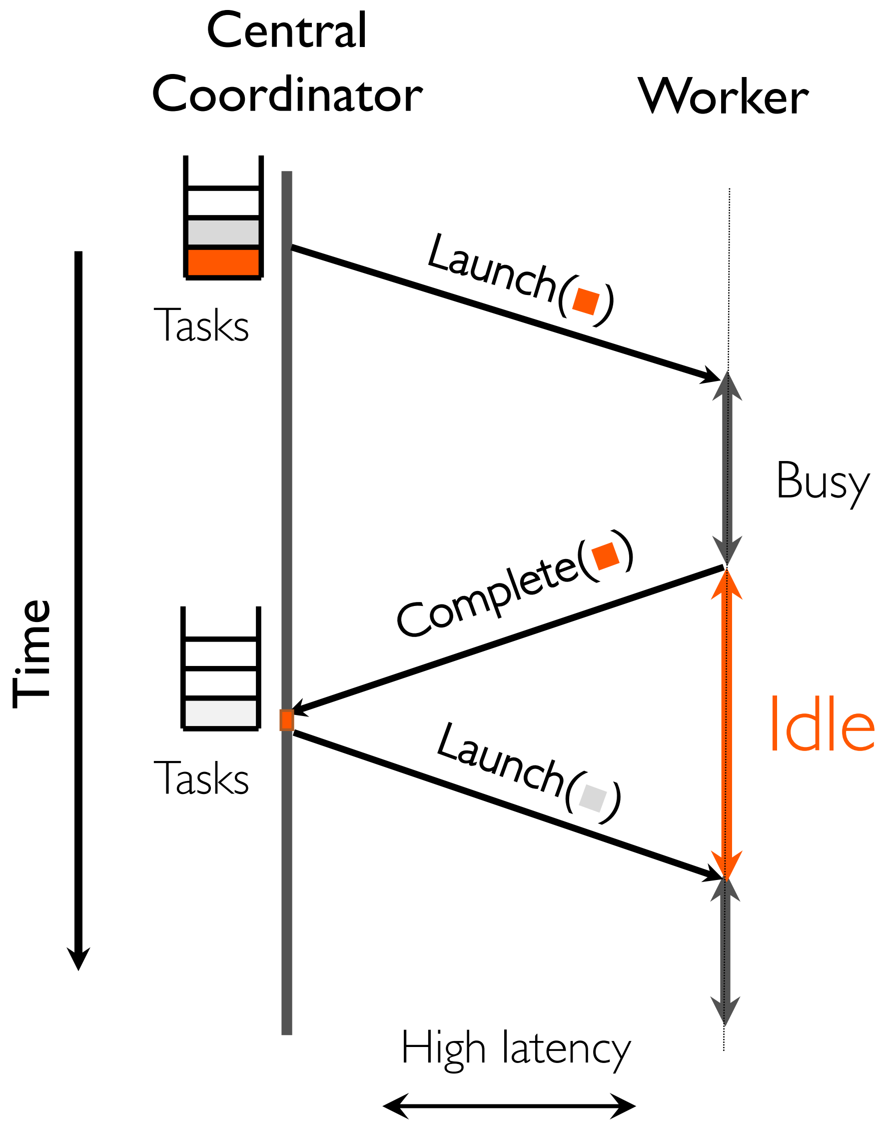
Fast Distributed Computation Over Slow Networks



w/ Fan Lai, Jie You, and others

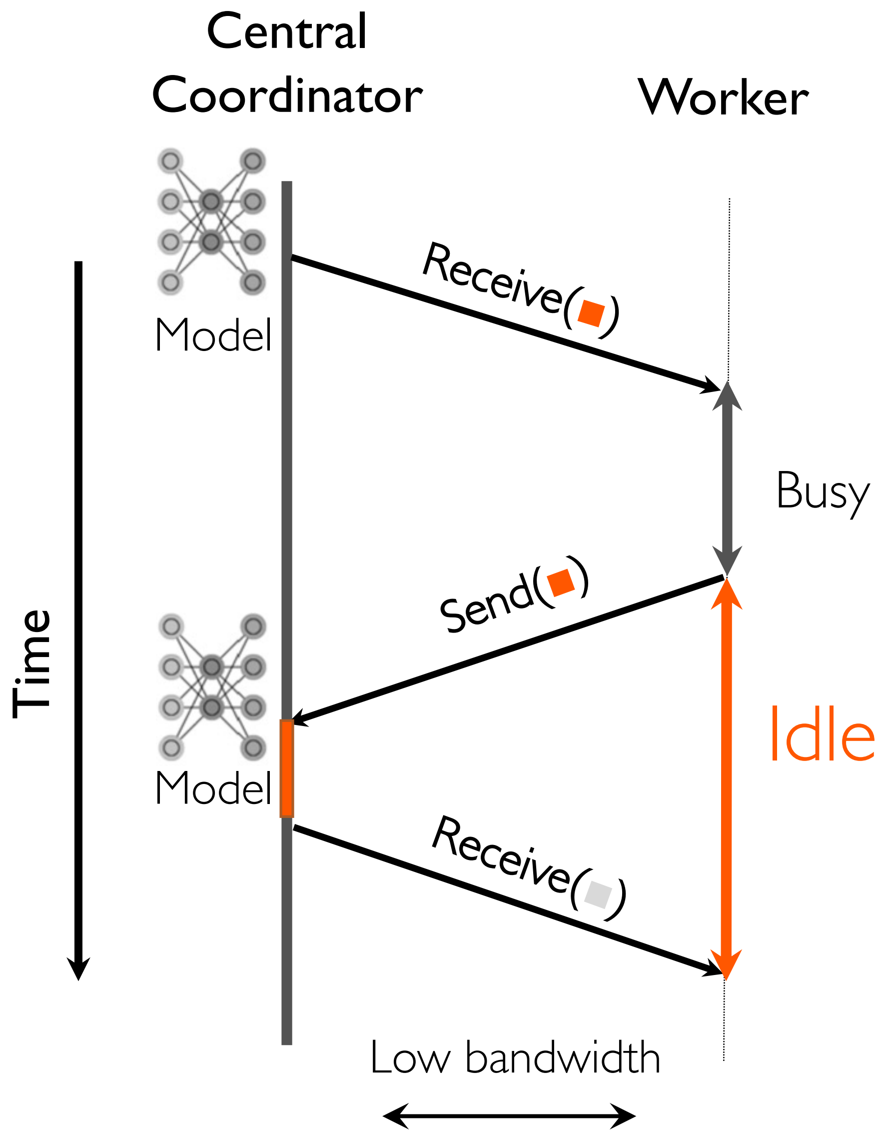
NSDI'20

Latency Impact on Short Computations



5X worse completion times for <u>interactive analytics</u> when running on 1ms vs 100ms networks

Bandwidth Impact on Long Computations



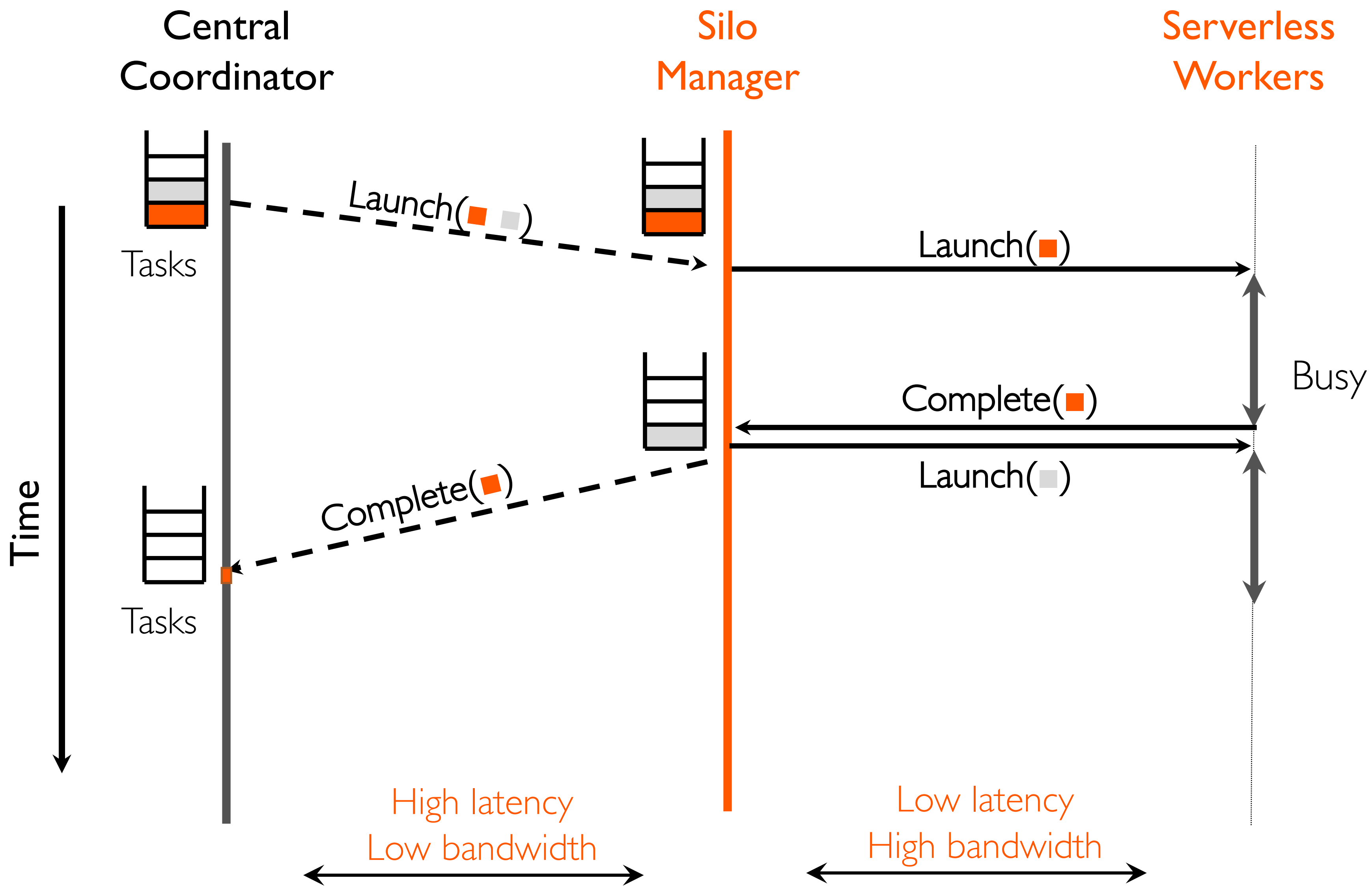
3X worse completion times for <u>machine learning</u> when running on 10Gbps vs 1 Gbps networks

Low Bandwidth

High Latency

Compute Idling

Sol in One Slide



Challenges

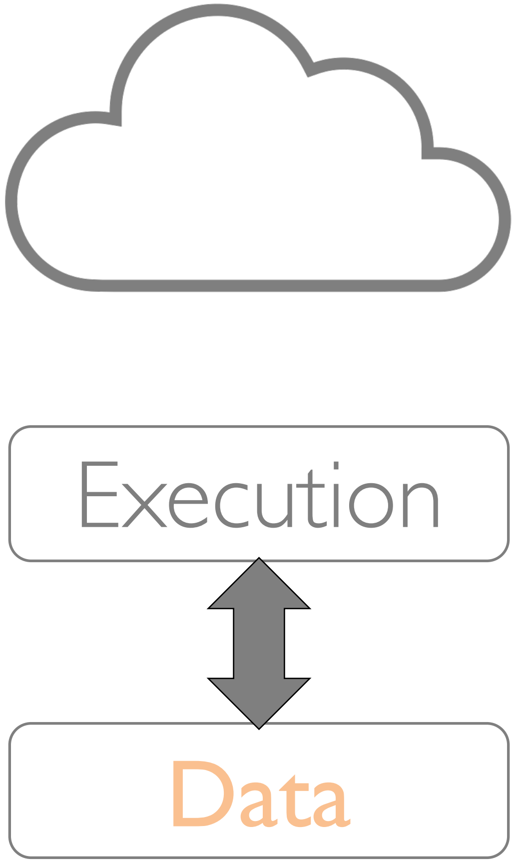
1. How many tasks to push?
2. When to push?
3. How to handle dependencies?
4. How to handle failures?
5. …

Large Performance Improvements

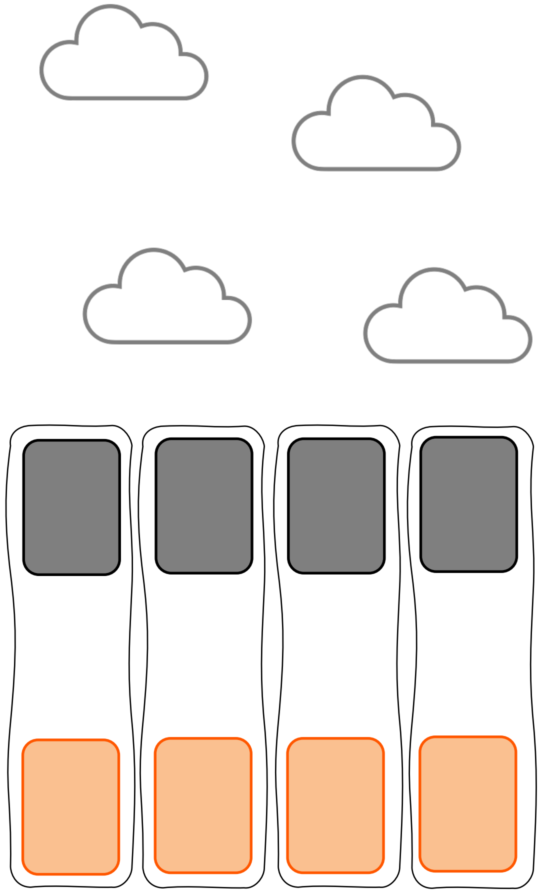
Deployed across 10 silos
Baseline: Apache Spark
Workloads: TPC-DS/H and HiBench

- **4X-16X** improvement in cross-silo federated learning and analytics
- **1.8X** improvement in compute utilization

Cloud L&A



Cross-Silo FL&A



Cross-Device FL&A

Cross-Device FL&A

1. Heterogeneous data
2. Heterogeneous devices
3. Enormous scale
4. Pervasive uncertainty

Oort

Efficient Federated Learning via Guided Participant Selection

w/ Fan Lai and others

OSDI'21 Distinguished Artifact

Random Client Selection Can be Suboptimal

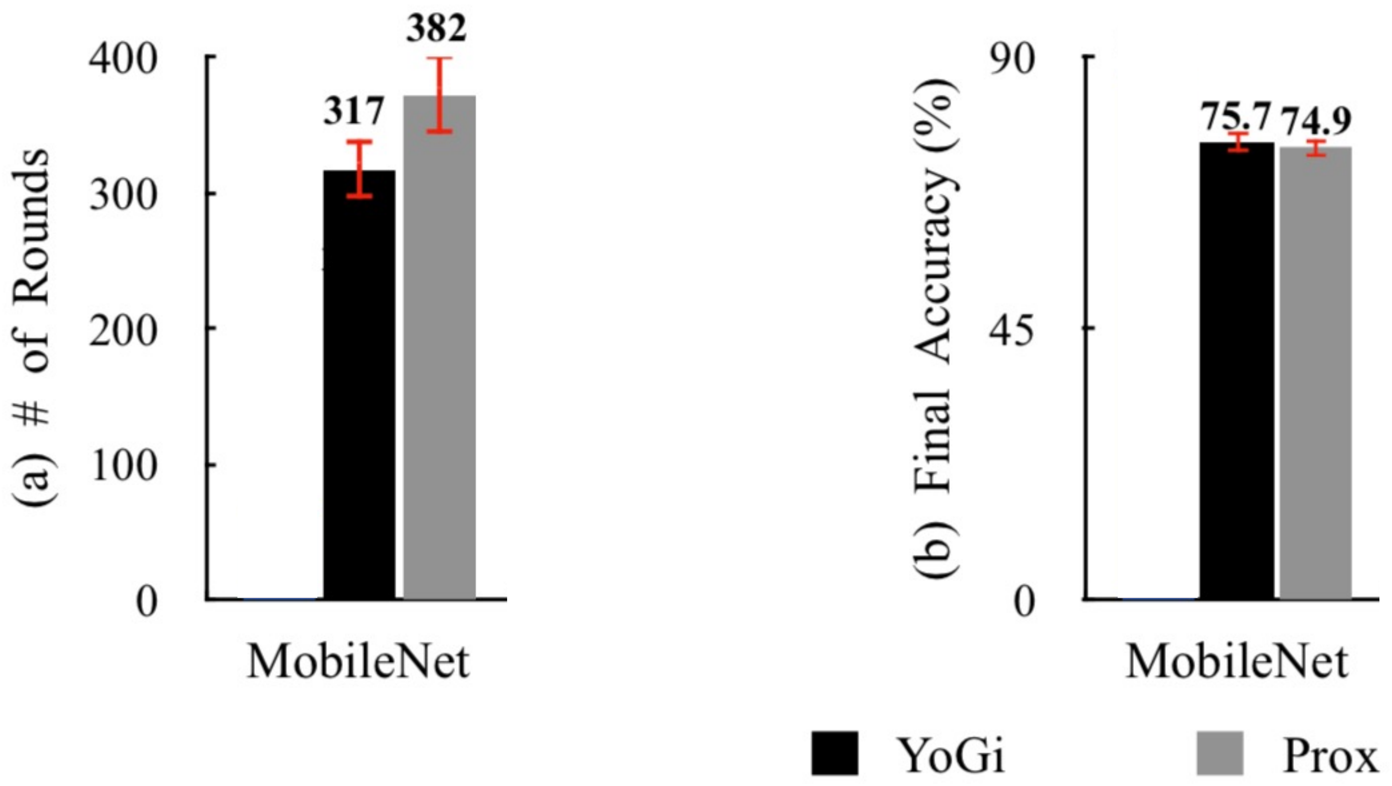
Inefficient training when overlooking heterogeneity

- Non-IID data leads to more rounds, lower accuracy
- Heterogenous devices lead to longer rounds

No guarantees on what the sampled population is being tested

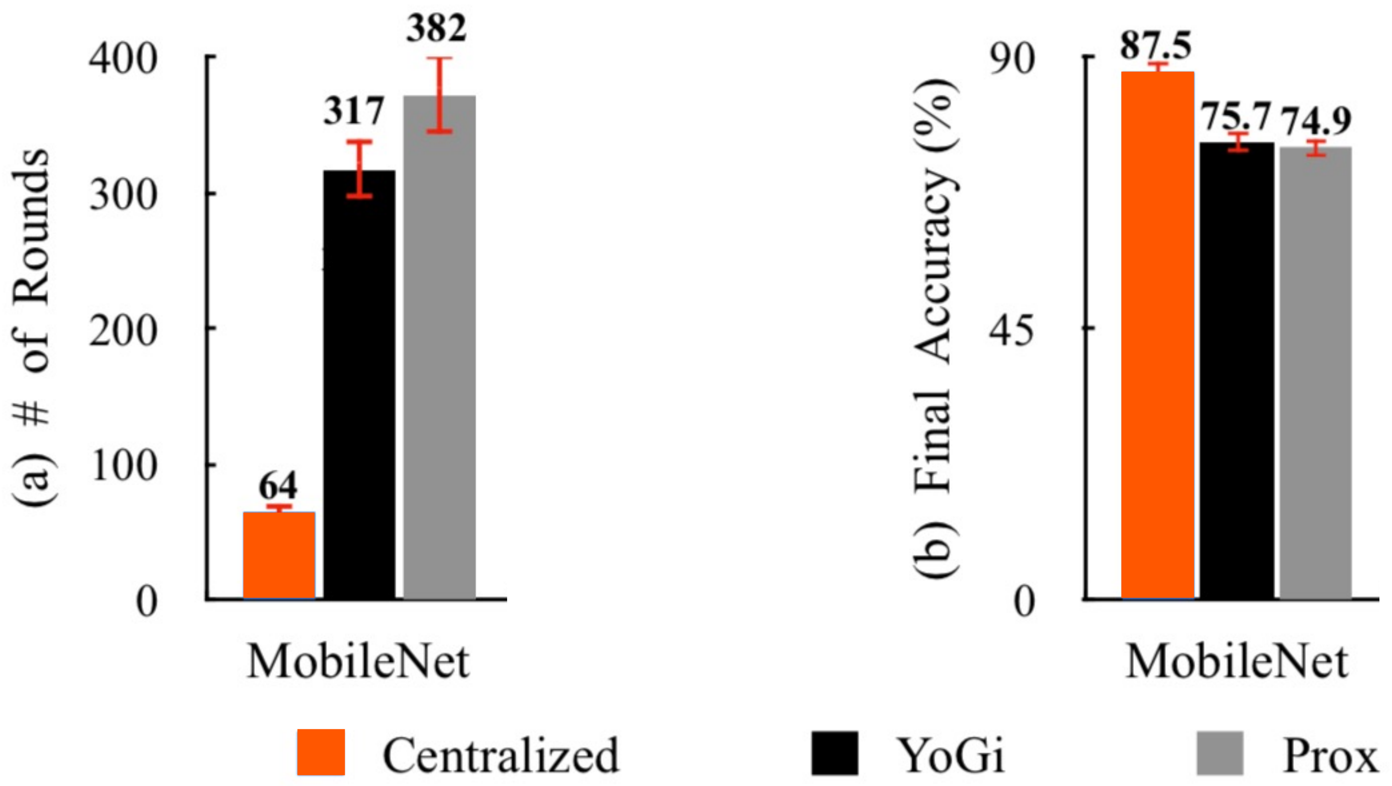
- Developer may want representative distribution

Random Selection Can be Suboptimal



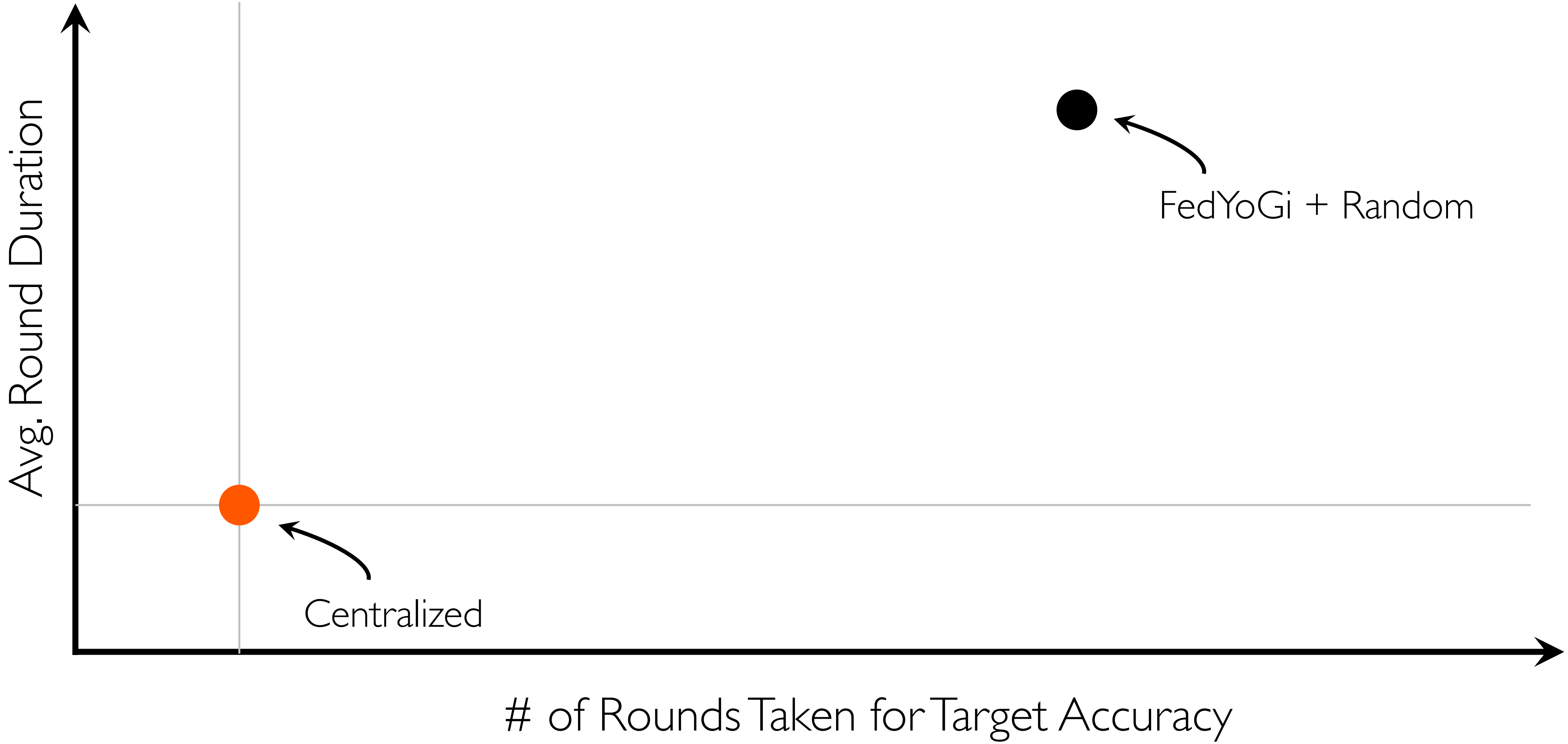
OpenImage dataset with 1.6M images
14k clients; 100 per round (randomly selected)

Random Selection Can be Suboptimal



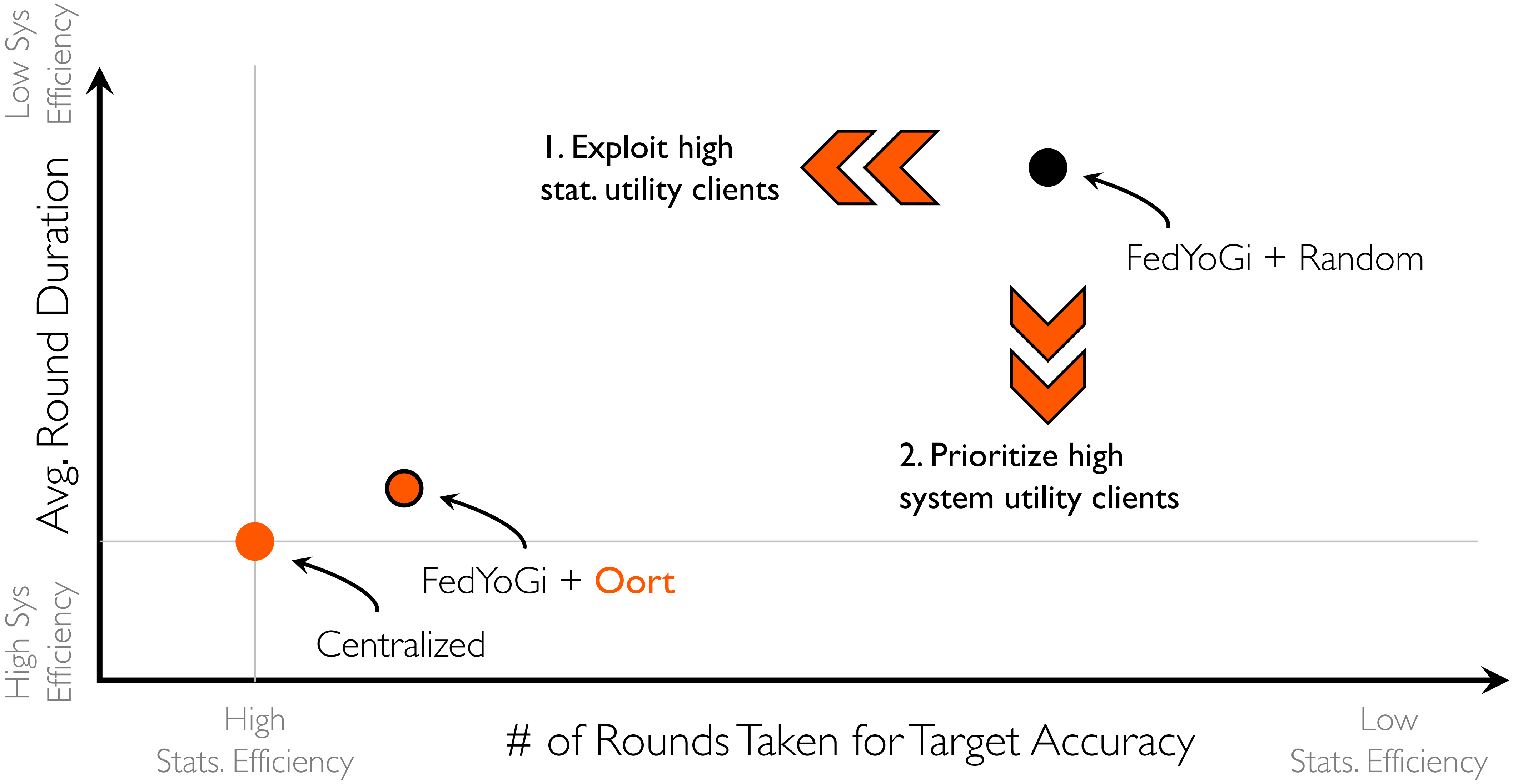
OpenImage dataset with 1.6M images
14k clients; 100 per round (randomly selected)

Time-to-Accuracy in Training



MobileNet on OpenImage dataset

Oort in One Slide



MobileNet on OpenImage dataset

Challenges

1. Which clients would improve statistical efficiency?
2. How to tradeoff statistical and system efficiency?
3. How to avoid stale information at scale?
4. How to be robust against noise?
5. …

Large Performance Improvements

FedYoGi+**Oort** over FedYoGi+Random	Stats.	Sys.	Overall	Accuracy
OpenImage/MobileNet	**2.3X**	**1.5X**	**3.3X**	**+9.8%**
Reddit/Albert	**1.5X**	**4.9X**	**7.3X**	**+4.4%**
Google Speech/ResNet-34	**1.2X**	**1.1X**	**1.3X**	**+2.2%**

$\underbrace{\text{Stats. / Sys. / Overall}}_{\textbf{Faster}}$

FedScale.ai

Benchmarking Model and System Performance of Federated Learning at Scale

w/ Fan Lai and others

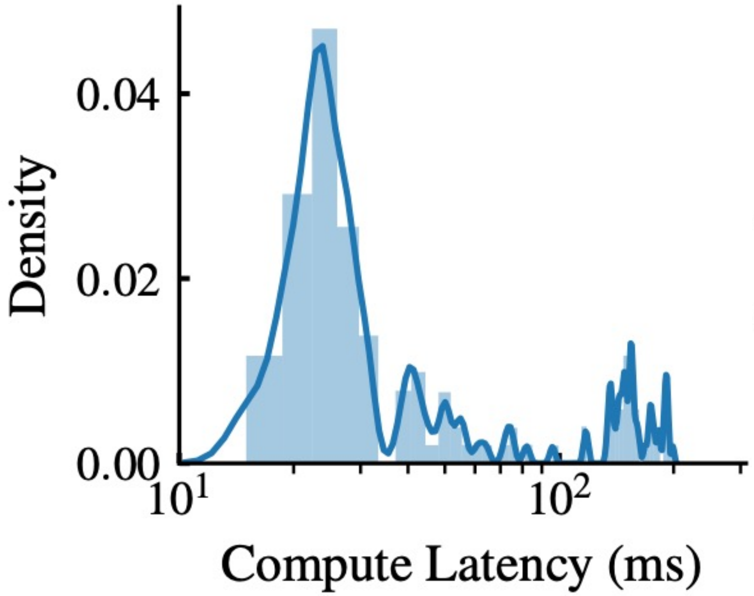
ResilientFL'21 Best Paper

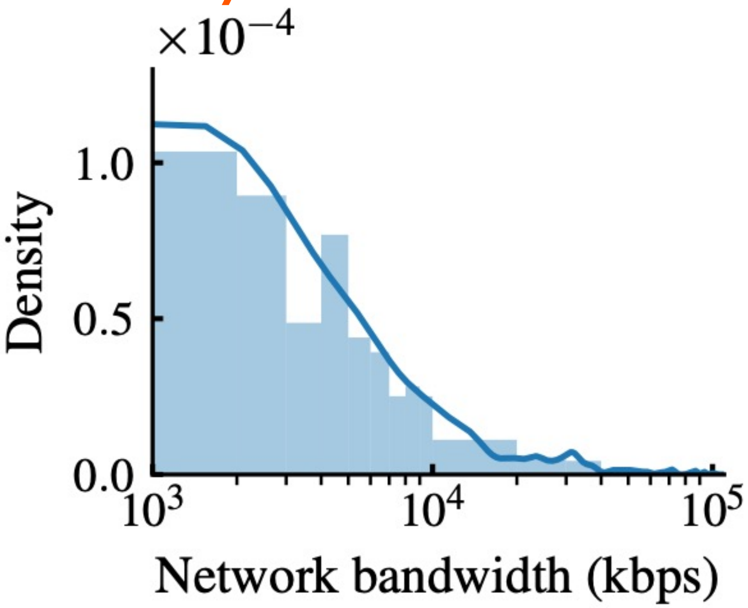
arXiv'21 (2105.11367)

Missing Pieces in Existing Benchmarks

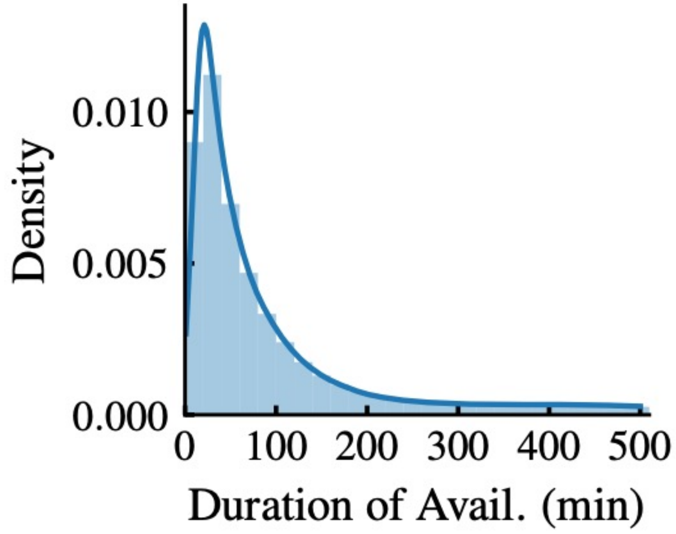
- **Systems details**
  - Network latency-bandwidth characteristics
  - End device characteristics (compute resources, battery, connectivity etc.)
  - Cloud resource characteristics
- **Scale**
  - Heterogeneity of client data
  - Availability of clients

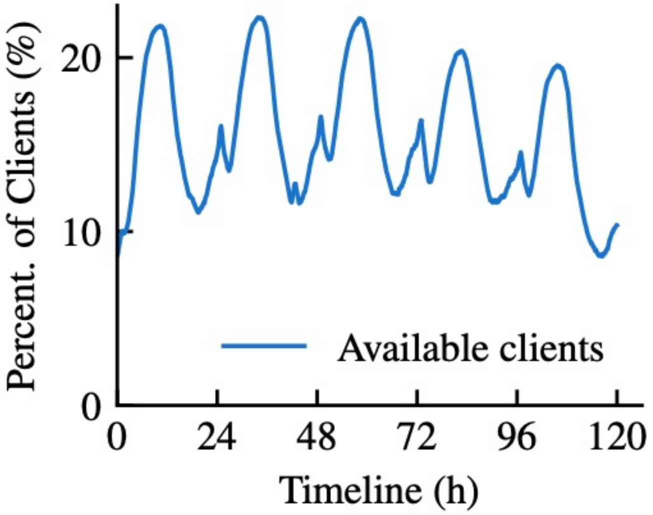
Millions of Client Systems Traces





Heterogeneous computation & communication speed



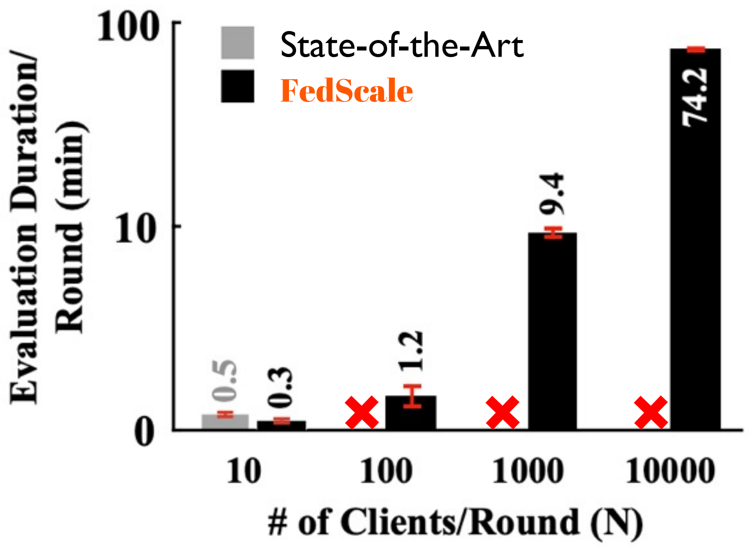


Dynamics of client availability in the wild

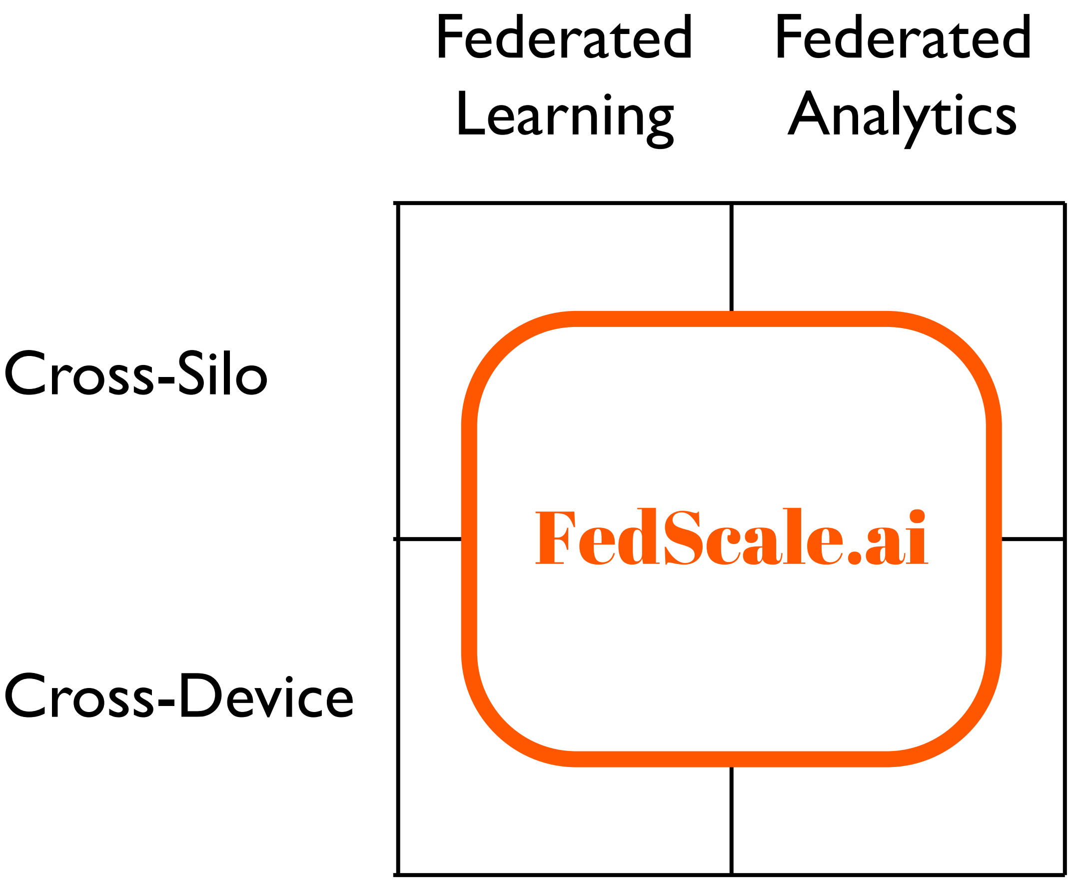
Large Datasets and Scalable Runtime

Category	Name	Data Type	#Clients	#Instances
CV	iNature	Image	2,295	193K
	FEMNIST	Image	3,400	640K
	OpenImage	Image	13,771	1.3M
	Google Landmark	Image	43,484	3.6M
	Charades	Video	266	10K
	VLOG	Video	4,900	9.6K
	Waymo Motion	Video	496,358	32.5M
NLP	Europarl	Text	27,835	1.2M
	Blog Corpus	Text	19,320	137M
	Stackoverflow	Text	342,477	135M
	Reddit	Text	1,660,820	351M
	Amazon Review	Text	1,822,925	166M
	CoQA	Text	7,189	114K
	LibriTTS	Text	2,456	37K
	Google Speech	Audio	2,618	105K
	Common Voice	Audio	12,976	1.1M
Misc ML	Taobao	Text	182,806	20.9M
	Fox Go	Text	150,333	4.9M

FedScale can support **orders-of-magnitude** more clients on the same underlying cluster



ShuffleNet on OpenImage dataset
10 GPUs



1. Data traces
2. System traces
3. Models
4. Scale factors
5. Scalable runtime
6. Diverse backends
7. Metrics
8. …

Application

Planning

Execution

Data

CellScope@MobiCom'18

Fed-ensemble@arXiv'21

Auxo

QOOP@OSDI'18

Oort@OSDI'21

NOCS@SPAA'19

Terra@arXiv'19

Sol@NSDI'20

Flamingo

FedScale@arXiv'21

Pando@NSDI'20

Research: Rethink software stacks

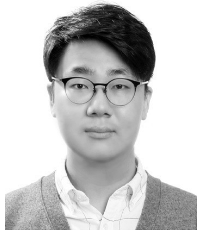
- Network-Aware
- Heterogeneity-Aware
- Adaptive

Service: Create evaluation platforms

- Faithful representation
- Easy to use
- Fast and scalable

https://github.com/ SymbioticLab

Current PhD Students


Jae-Won Chung


Insu Jang


Fan Lai

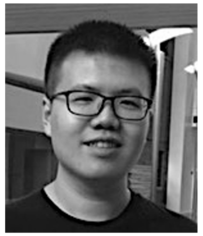

Jiachen Liu


Hasan Al Maruf


Sanjay Singapuram


Jie You


Peifeng Yu


Yiwen Zhang

Undergraduate & Master's

Zhezheng Chen
Yinwei Dai
Shuoren Fu
Yash Gaitonde
Songyuan Guan
Chuheng Hu
Jack Kosaian
Qinye Li
Yang Liu
Yuze Lou
Alexander Neben
Yuqing Qiu
Wenting Tan
Yue Tan
Kaiwei Tu
Yuchen Wang
Yujia Xie
Yilei Xu
Jiaxing Yang
Yiwei Zhang
Jiangchen Zhu
Jingyuan Zhu
Xiangfeng Zhu

Collaborators

Aditya Akella
Ganesh Ananthanarayanan
Wei Bai
Thomas M Braun
Vladimir Braverman
Shuchi Chawla
Kai Chen
Li Chen
Asaf Cidon
Sung Won Choi
Yanhui Geng
Ali Ghodsi
Ayush Goel
Robert Grandl
Juncheng Gu*
Chuanxiong Guo
Vibhuti Gupta
Anthony Huang
Anand P. Iyer
Myeongjae Jeon
Xin Jin
Samir Khuller
Raed Al Kontar
Tan N. Le
Youngmoon Lee
Li Erran Li
Hongqiang Liu
Zhenhua Liu
Harsha V. Madhyastha
Kshiteej Mahajan
Barzan Mozafari
Linh Nguyen
Aurojit Panda
Manish Purohit
Junjie Qian
Kannan Ramchandran
K.V. Rashmi
Naichen Shi
Kang G. Shin
Scott Shenker
Brent Stephens
Ion Stoica
Muneesh Tewari
Xiao Sun
Muhammed Uluyol
Shivaram Venkataraman
Carl Waldspurger
Hongyi Wang
Jingfeng Wu
Sheng Yang
Bairen Yi
Dong Young Yoon
Zhuolong Yu
Hong Zhang
Junxue Zhang
Yuhong Zhong
Yibo Zhu

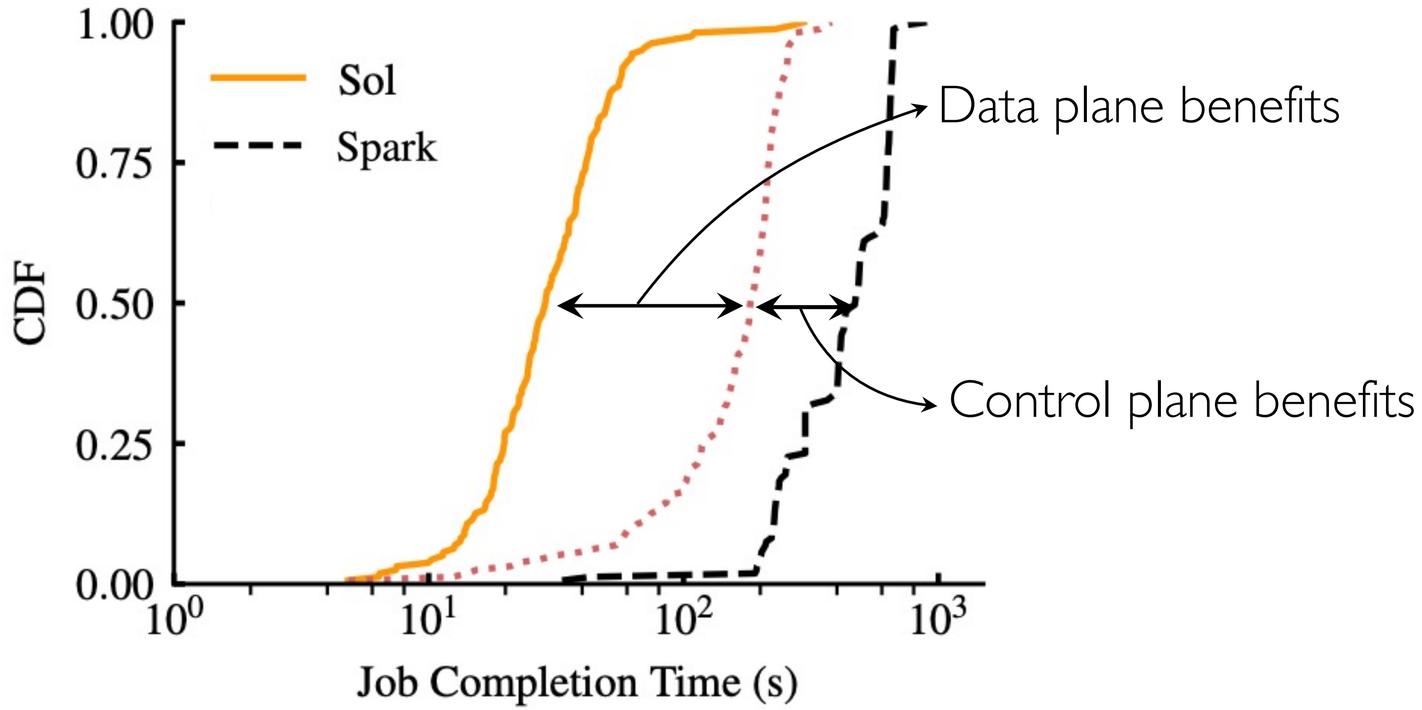
https://github.com/SymbioticLab

Core Ideas

Reduce compute idleness in silos by redesigning both control and data planes of federated systems

1. **Sol** Control Plane	**Proactively push work** *to workers in remote sites before they ask for additional work*
2. **Sol** Data Plane	**Decouple computation and communication** *roles of tasks using serverless compute and disaggregated storage*

Performance Breakdown



16.4X improvement in cross-silo federated analytics

How to Use Oort?

1. Select subset with <X deviation from the global distribution

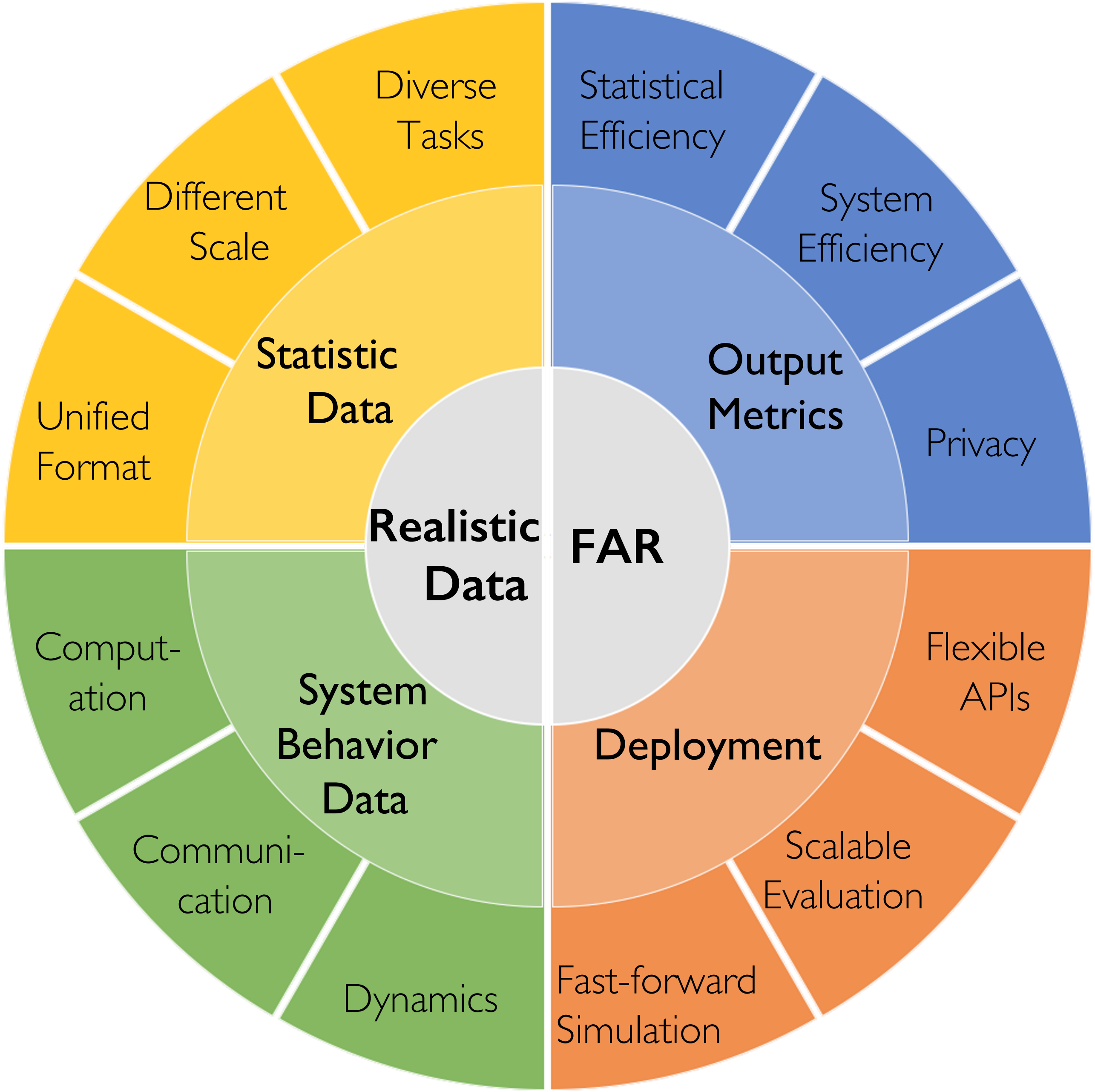
```
participants = oort.select_by_deviation(dev_target,
  range_of_capacity, total_num_clients)
```

2. Select $[N_1, N_2, \ldots, N_K]$ samples of categories $[C_1, C_2, \ldots, C_K]$

```
participants = oort.select_by_category(request_list,
  testing_config)
```

Comparison

	LEAF	FedEval	FedML	Flower	**FedScale**
Heter. Client Dataset	⭕	×	⭕	⭕	√
Heter. System Speed	×	×	×	×	√
Client Availability	×	×	×	×	√
Scalable Platform	×	⭕	⭕	√	√
Flexible APIs	×	×	√	√	√


Realistic Data
FAR
Statistic Data
Diverse Tasks
Different Scale
Unified Format
Output Metrics
Statistical Efficiency
System Efficiency
Privacy
System Behavior Data
Comput-ation
Communi-cation
Dynamics
Deployment
Flexible APIs
Scalable Evaluation
Fast-forward Simulation

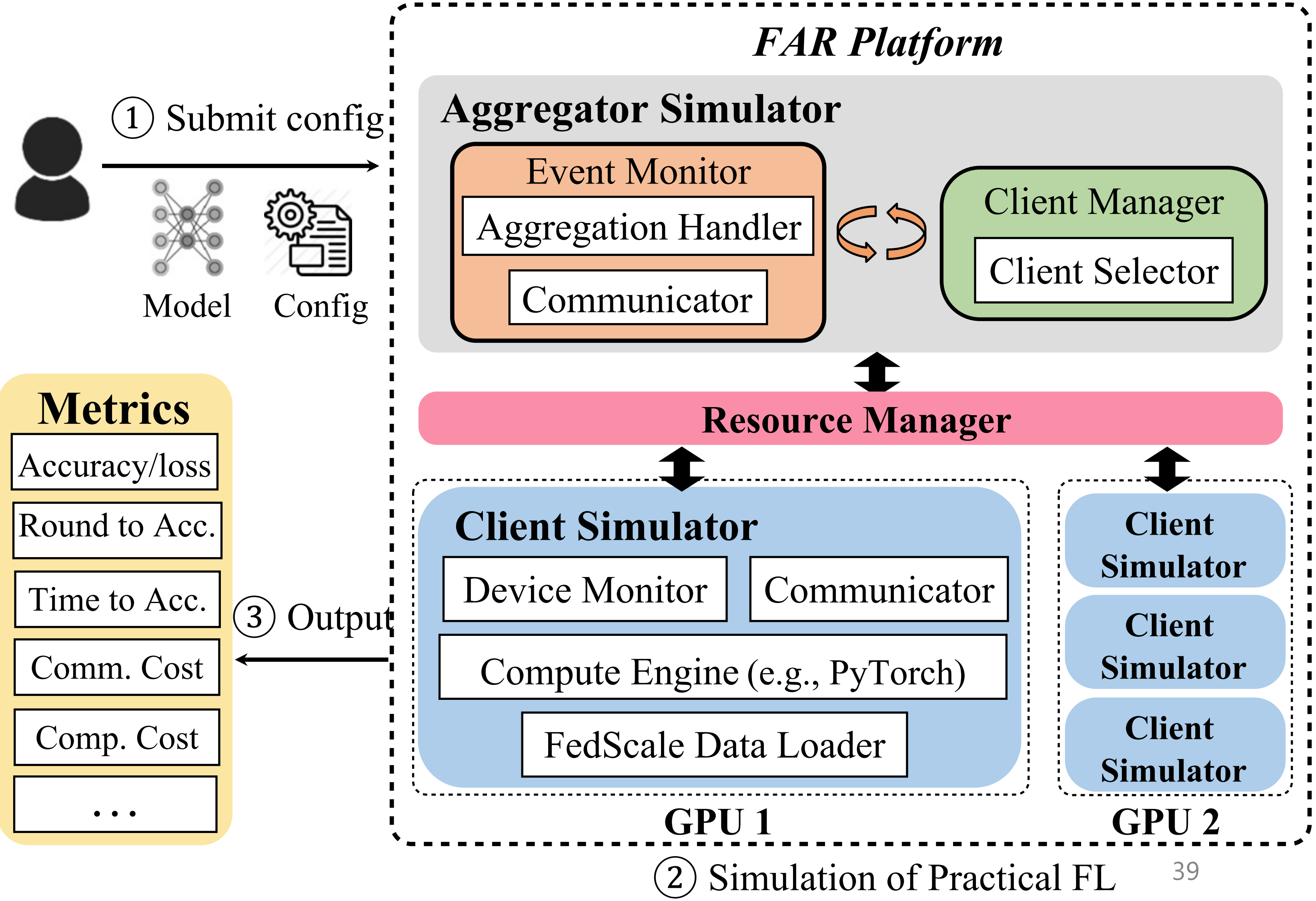
FAR: FedScale Automated Runtime

Scalable eval platform

- GPUs/CPUs
- High resource util.

Practical runtime

- Convergence
- System duration



FAR: Easily-Deployable Benchmarking

- **Flexible APIs to automatically integrate new plugins**
  - Little effort to customize/benchmark new designs

Module	API Name	Example Use Case
Aggregator Simulator	`round_completion_handler`(*args)	Adaptive/secure model aggregation
	`client_completion_handler`(client_id, msg)	Straggler mitigation
	`push_msg_to_client`(client_id, msg)	Model compression
Client Manager	`select_clients`(*args)	Client selection
	`select_model_for_client`(client_id)	Adaptive model selection
Client Simulator	`train`(client_data, model, config)	Local SGD/malicious attack
	`push_msg_to_aggregator`(msg)	Model compression

Oort [OSDI'21]

Some Example APIs

FAR: Easily-Deployable Benchmarking

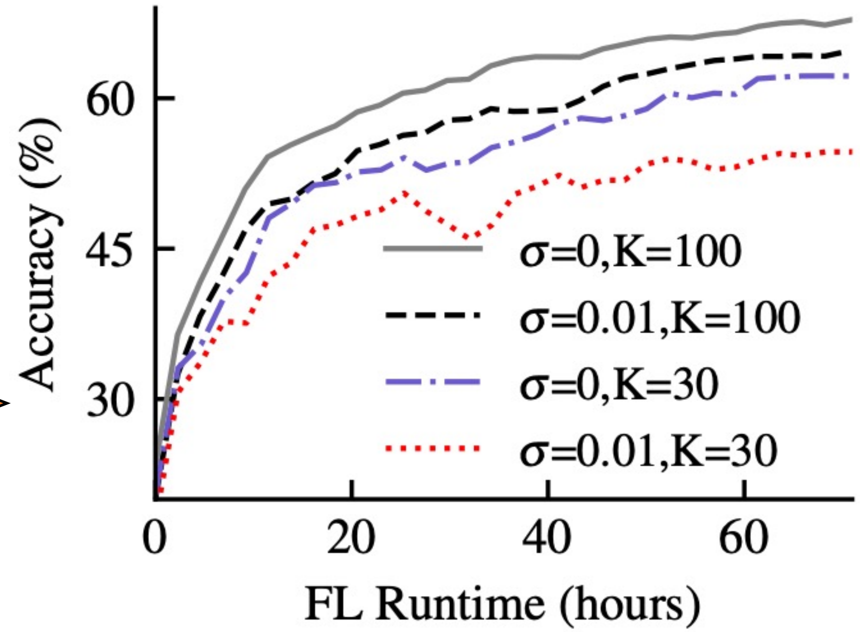
- **Flexible APIs to automatically integrate new plugins**
  - Little effort to customize/benchmark new designs

```python
from fedscale.core.client import Client

class Customized_Client(Client):
# Customize the training on each client
  def train(self,client_data,model,conf):
      # Get the training result from
      # the default training component
      training_result = super().train(
            client_data, model, conf)

      # Clip updates and add noise
      secure_result = secure_impl(
                  training_result)
      return secure_result
```

A few lines are enough for benchmarking

Differential Private-SGD

σ (privacy target)
K (# participants/round)



FedScale can benchmark more realistic statistical/system performance