Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds

Kevin Hsieh

Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R. Ganger, Phillip B. Gibbons, Onur Mutlu†

Carnegie Mellon University

†ETH zürich

Machine Learning and Big Data

- Machine learning is widely used to derive useful information from large-scale data

Pictures

Image Classification

Videos

Video Analytics

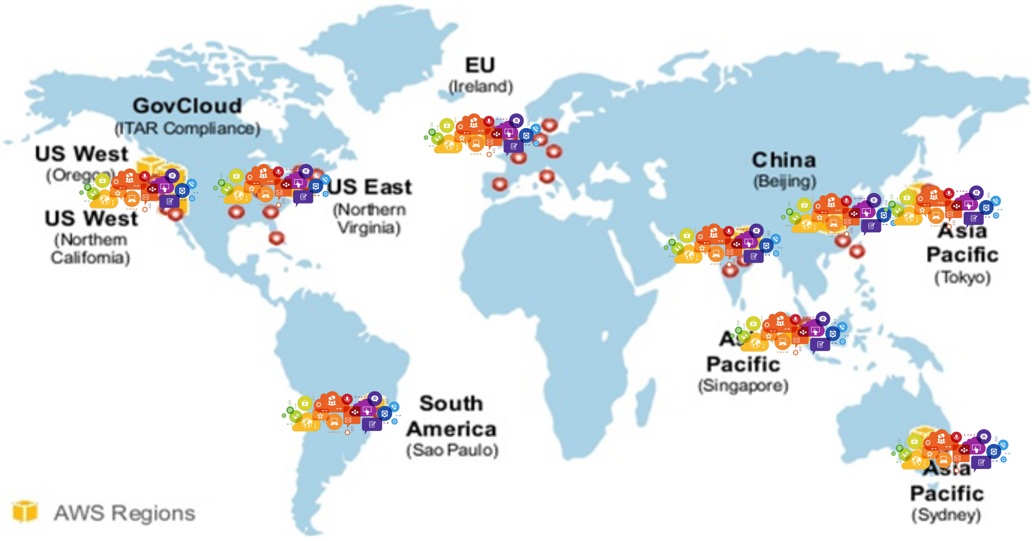
User Activities

……

Preference Prediction

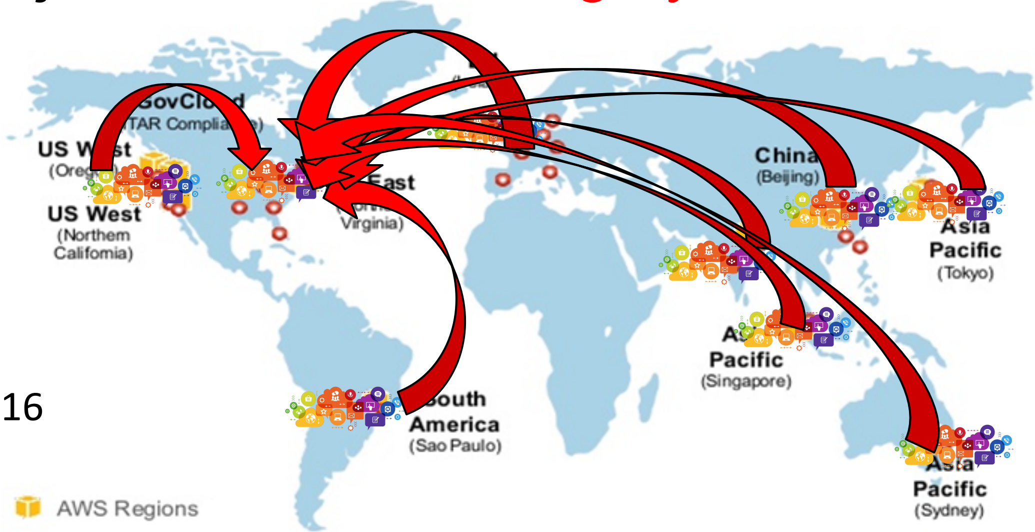
Big Data is Geo-Distributed

- A large amount of data is generated rapidly, all over the world



Centralizing Data is Infeasible [1, 2, 3]

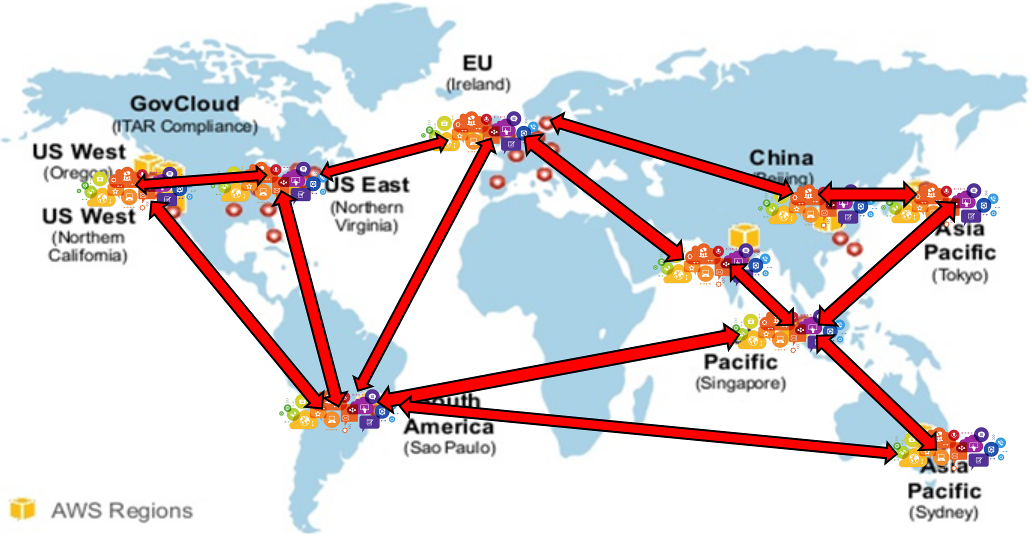
- Moving data over wide-area networks (WANs) can be extremely slow
- It is also subject to data sovereignty laws



1. Vulimiri et al., NSDI’15
2. Pu et al., SIGCOMM’15
3. Viswanathan et al., OSDI’16

Geo-distributed ML is Challenging

- No ML system is designed to run across data centers (up to 53X slowdown in our study)



Our Goal

- Develop a geo-distributed ML system
    - Minimize communication over wide-area networks
    - Retain the accuracy and correctness of ML algorithms
    - Without requiring changes to the algorithms

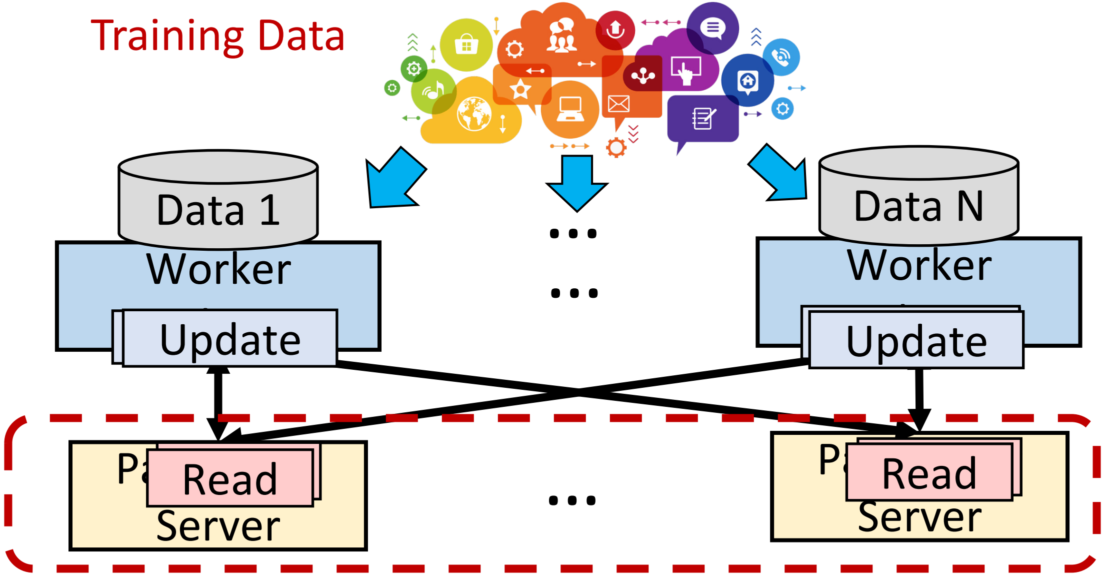
Key Result: 1.8-53.5X speedup over state-of-the-art ML systems on WANs

Outline

- Problem & Goal
- Background & Motivation
- Gaia System Overview
- Approximate Synchronous Parallel
- System Implementation
- Evaluation
- Conclusion

Background: Parameter Server Architecture

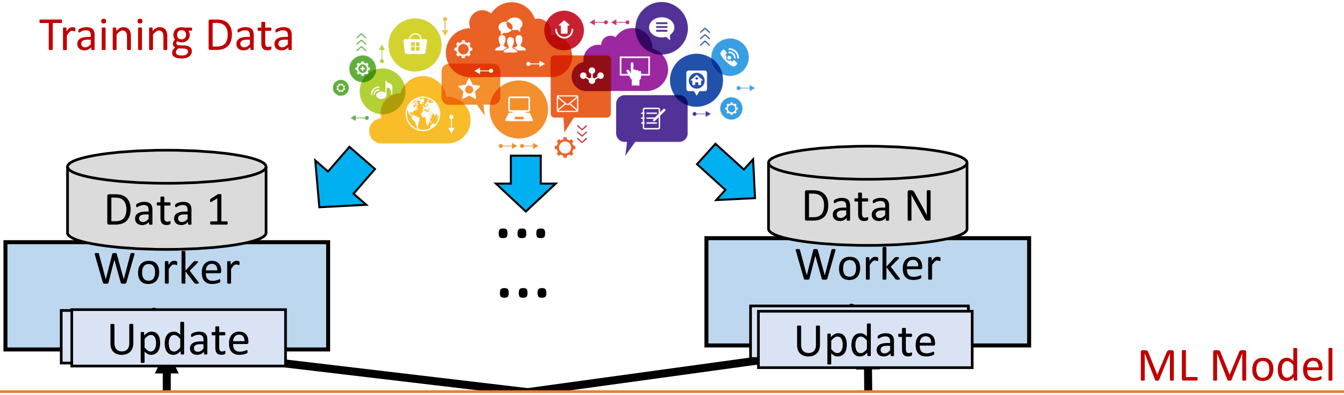
- The parameter server architecture has been widely adopted in many ML systems



ML Model

Background: Parameter Server Architecture

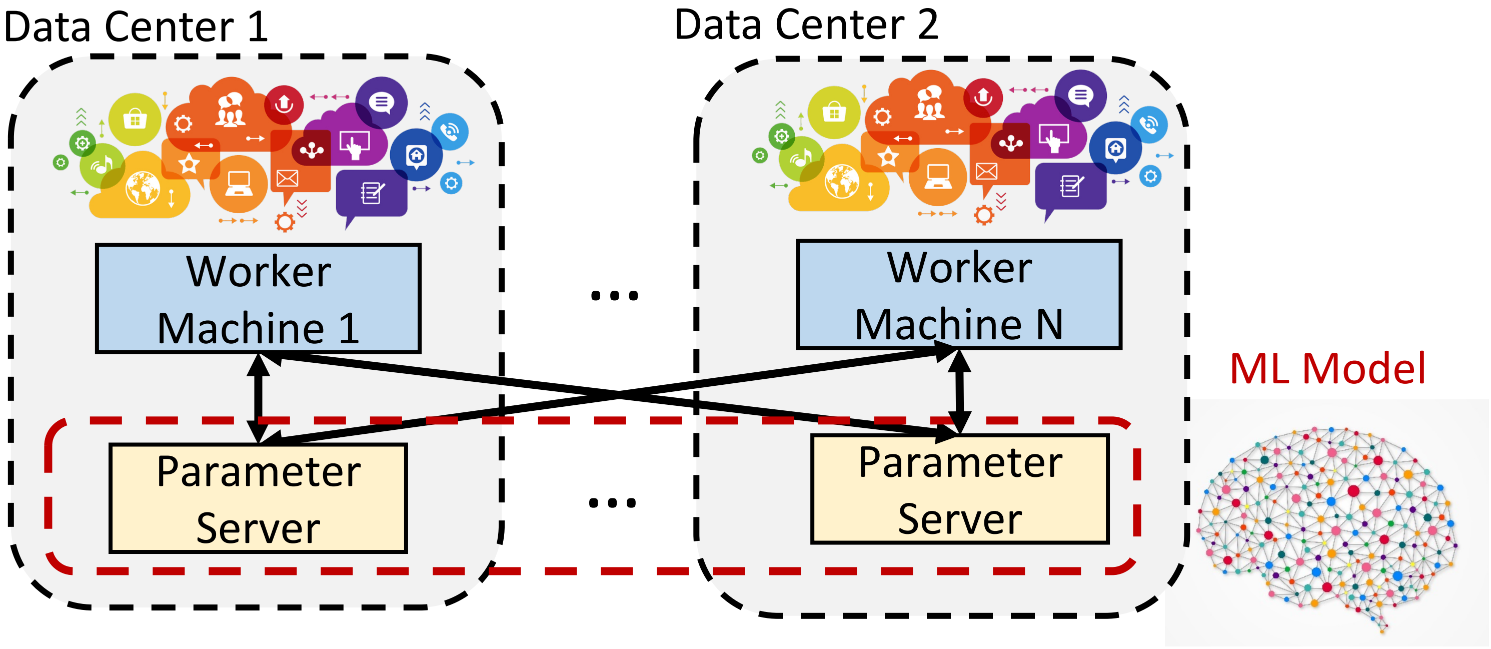
- The parameter server architecture has been widely adopted in many ML systems



Synchronization is critical to the accuracy and correctness of ML algorithms

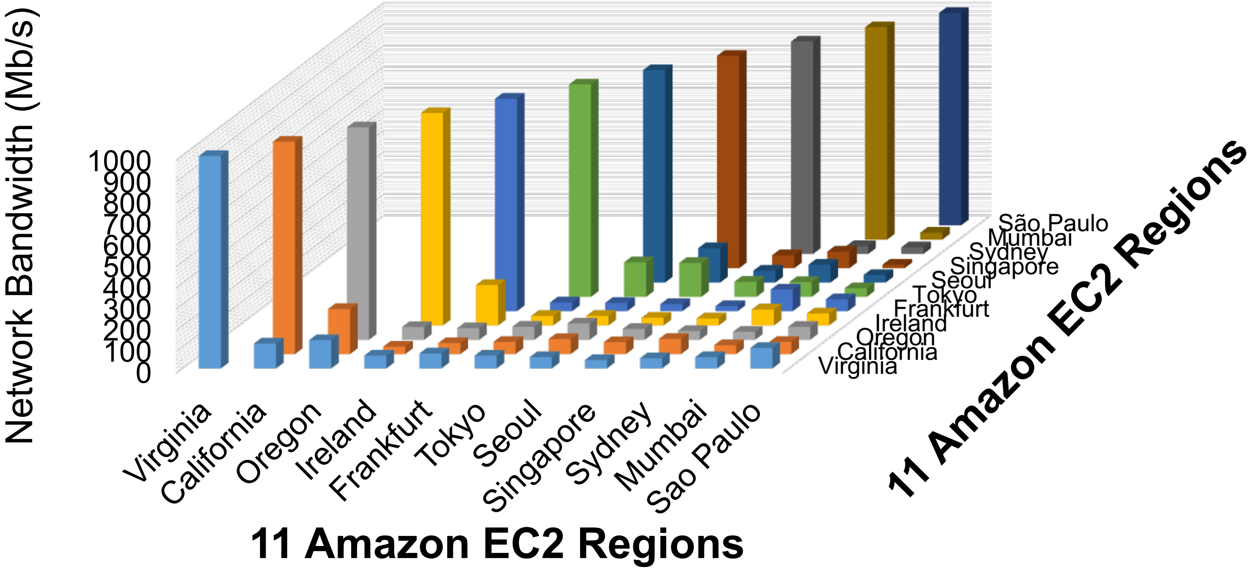
Deploy Parameter Servers on WANs

- Deploying parameter servers across data centers requires a lot of communication over WANs

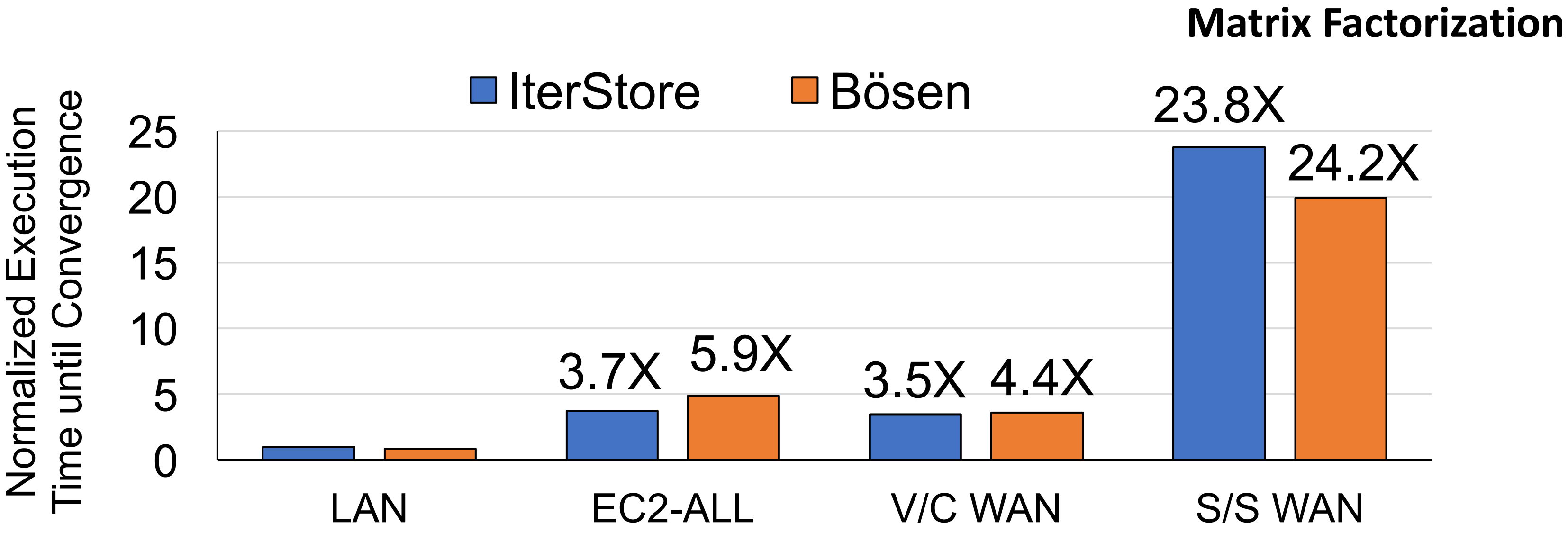


WAN: Low Bandwidth and High Cost

- WAN bandwidth is 15X smaller than LAN bandwidth on average, and up to 60X smaller
- In Amazon EC2, the monetary cost of WAN communication is up to 38X the cost of renting machines



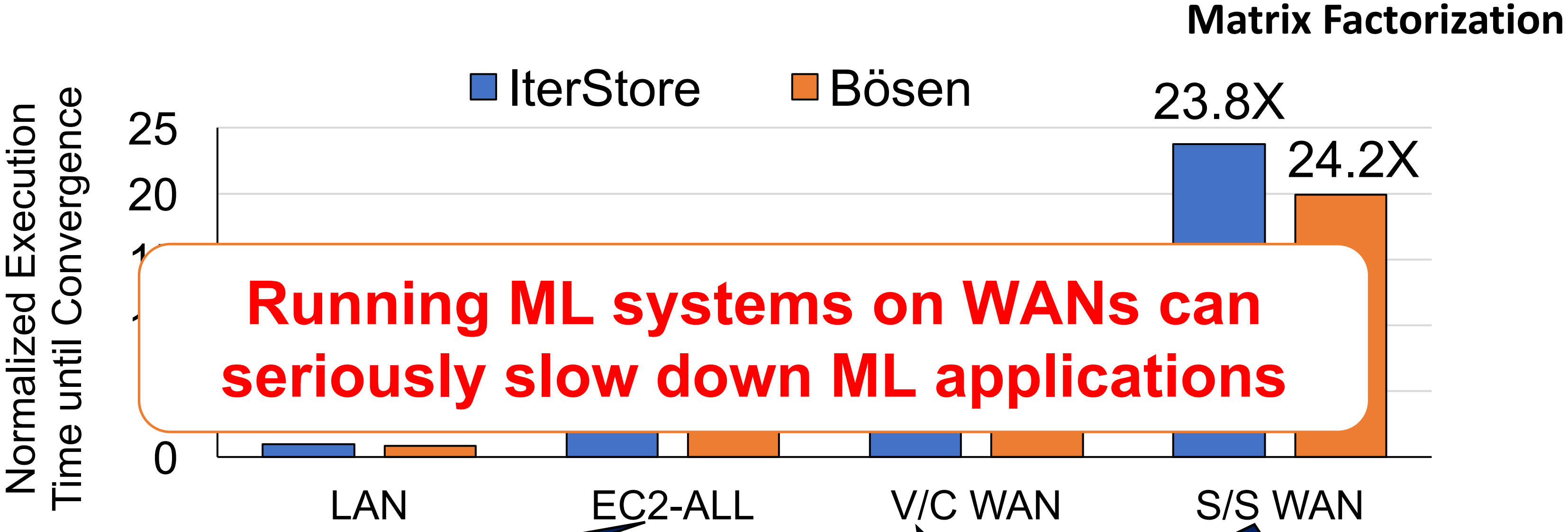
ML System Performance on WANs



1) Cui et al., "Exploiting Iterative-ness for Parallel ML Computations", SoCC'14

2) Wei et al., "Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics", SoCC'15

ML System Performance on WANs



Running ML systems on WANs can seriously slow down ML applications

11 EC2 Regions

Virginia / California

Singapore / São Paulo

1) e-ness for Parallel Co

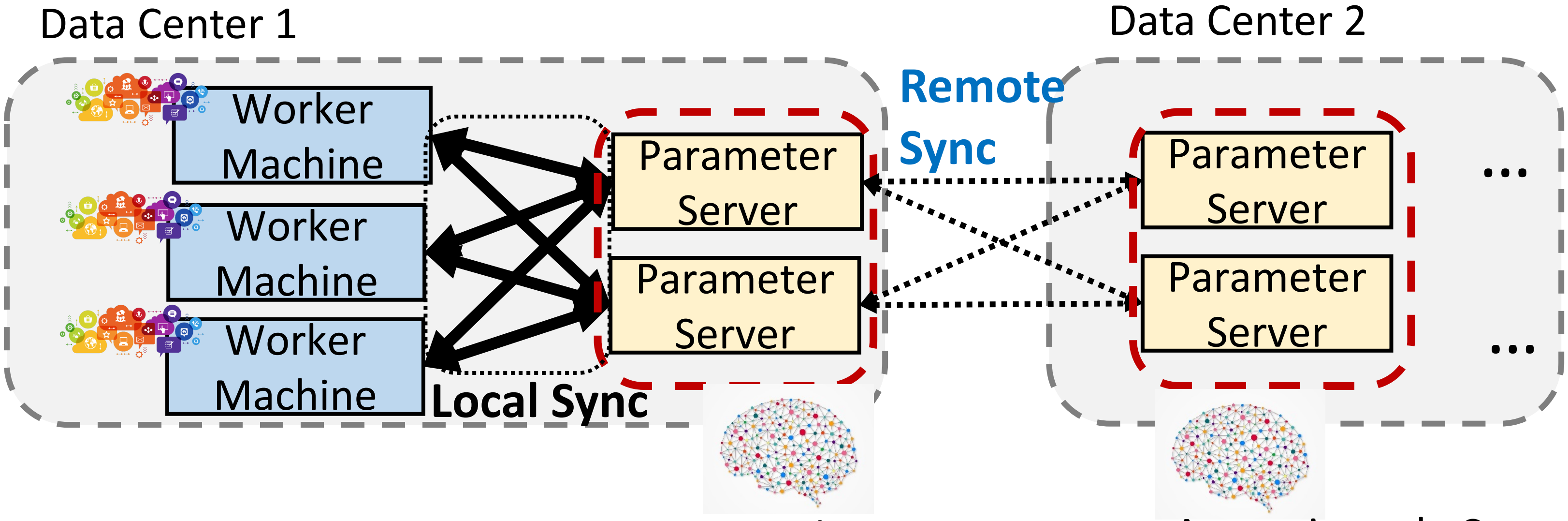
2) Wei et al., "Managed Commun ast Data-Parallel Iterative Analytics", So

Outline

- Problem & Goal
- Background & Motivation
- Gaia System Overview
- Approximate Synchronous Parallel
- System Implementation
- Evaluation
- Conclusion

Gaia System Overview

- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers

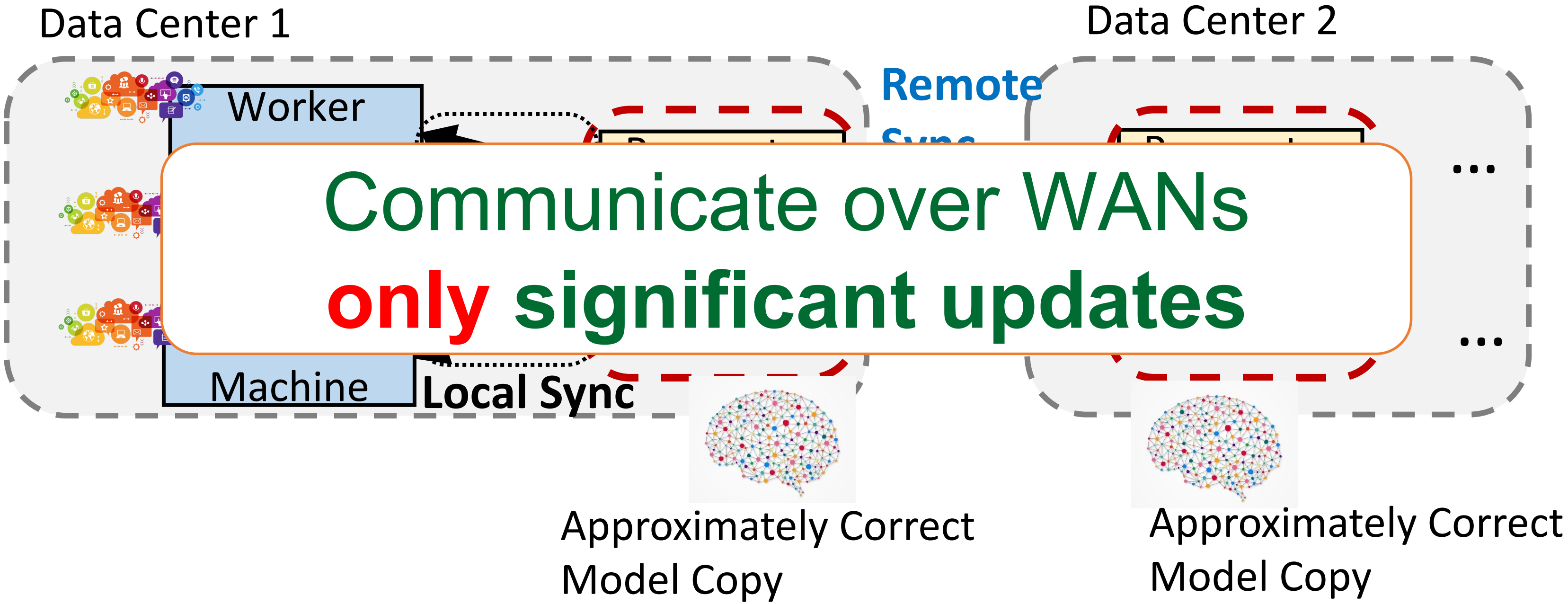


Approximately Correct Model Copy

Approximately Correct Model Copy

Gaia System Overview

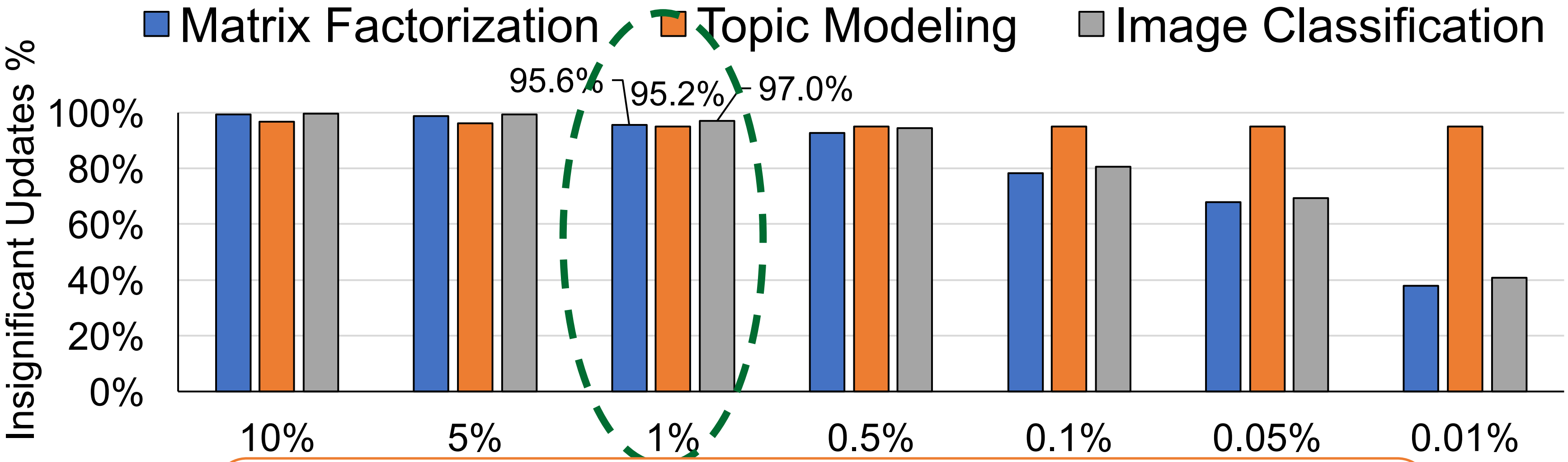
- **Key idea**: Decouple the synchronization model *within* the data center from the synchronization model *between* data centers


Data Center 1
Data Center 2
Worker
Machine
Local Sync
Remote
Sync
…
…
Approximately Correct Model Copy
Approximately Correct Model Copy

Communicate over WANs

only significant updates

Key Finding: Study of Update Significance



The vast majority of updates are ***insignificant***

Outline

- Problem & Goal
- Background & Motivation
- Gaia System Overview
- Approximate Synchronous Parallel
- System Implementation
- Evaluation
- Conclusion

Approximate Synchronous Parallel

The significance filter

- Filter updates based on their significance

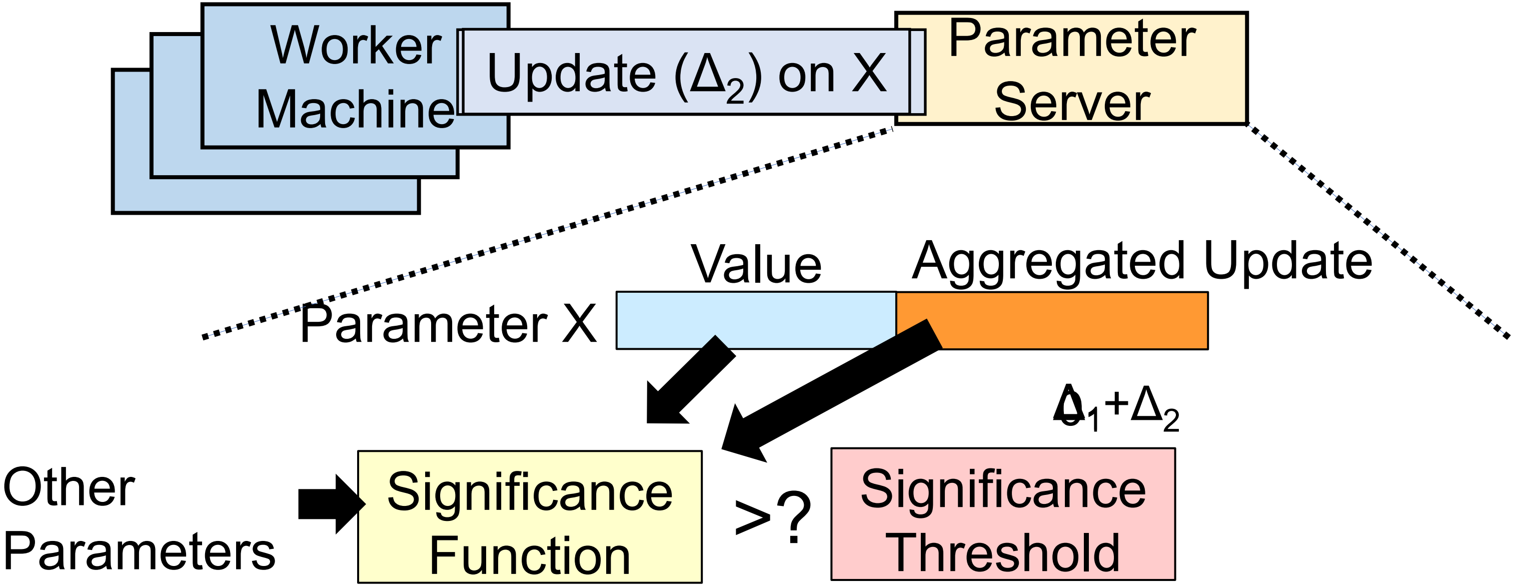
ASP selective barrier

- Ensure significant updates are read in time

Mirror clock

- Safe guard for pathological cases

The Significance Filter



$$\left\|\frac{Agg.Update}{Value}\right\|$$

$$1\%$$

$$\frac{10\%}{\sqrt{T}}$$

Approximate Synchronous Parallel

The significance filter

- Filter updates based on their significance

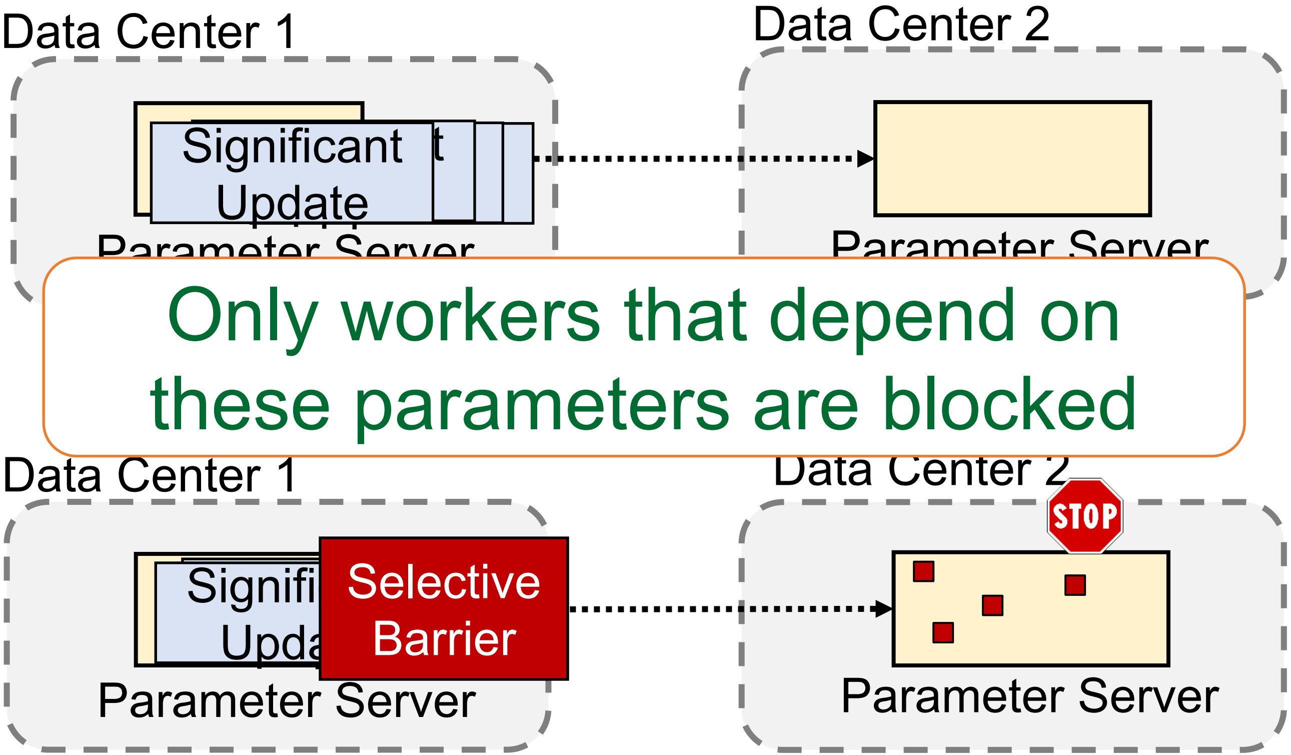
ASP selective barrier

- Ensure significant updates are read in time

Mirror clock

- Safeguard for pathological cases

ASP Selective Barrier

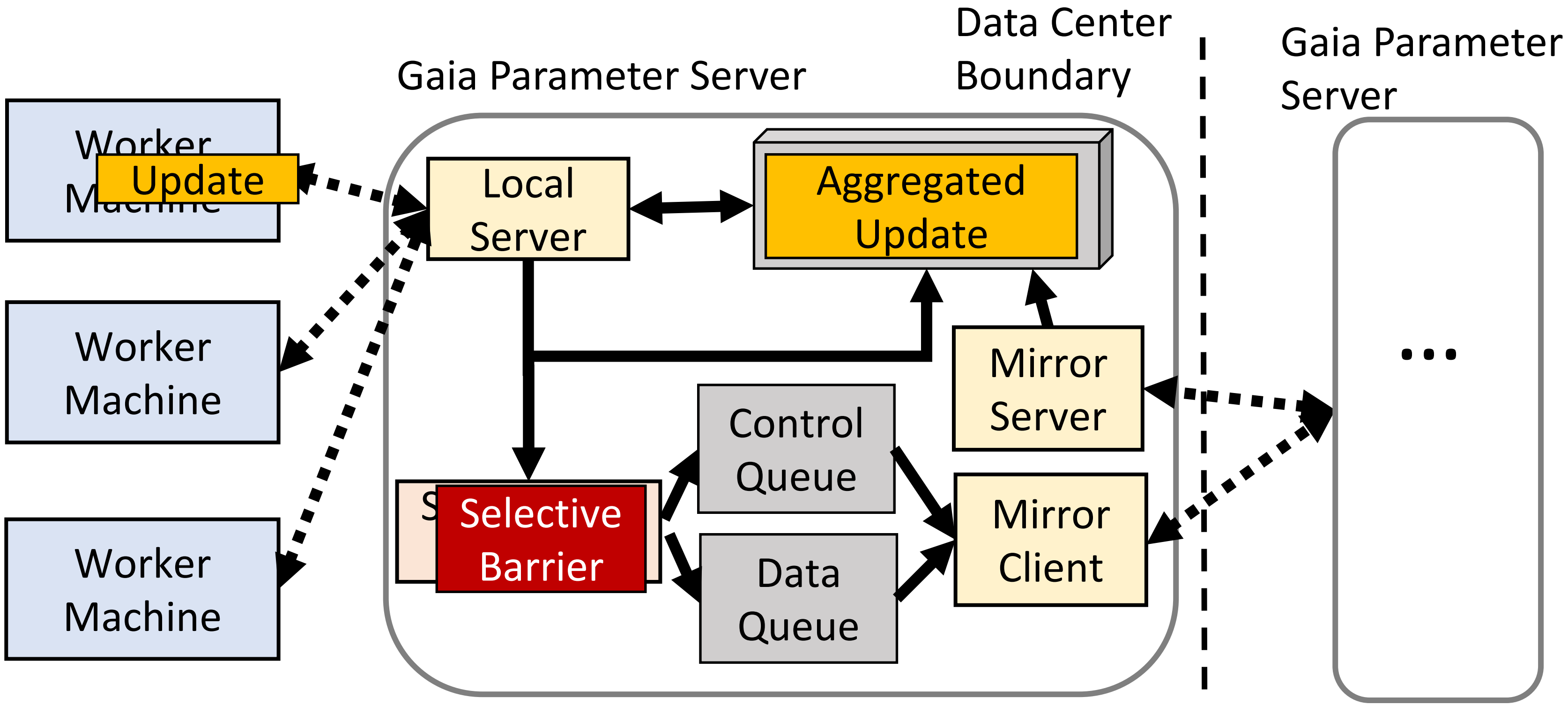


Only workers that depend on these parameters are blocked

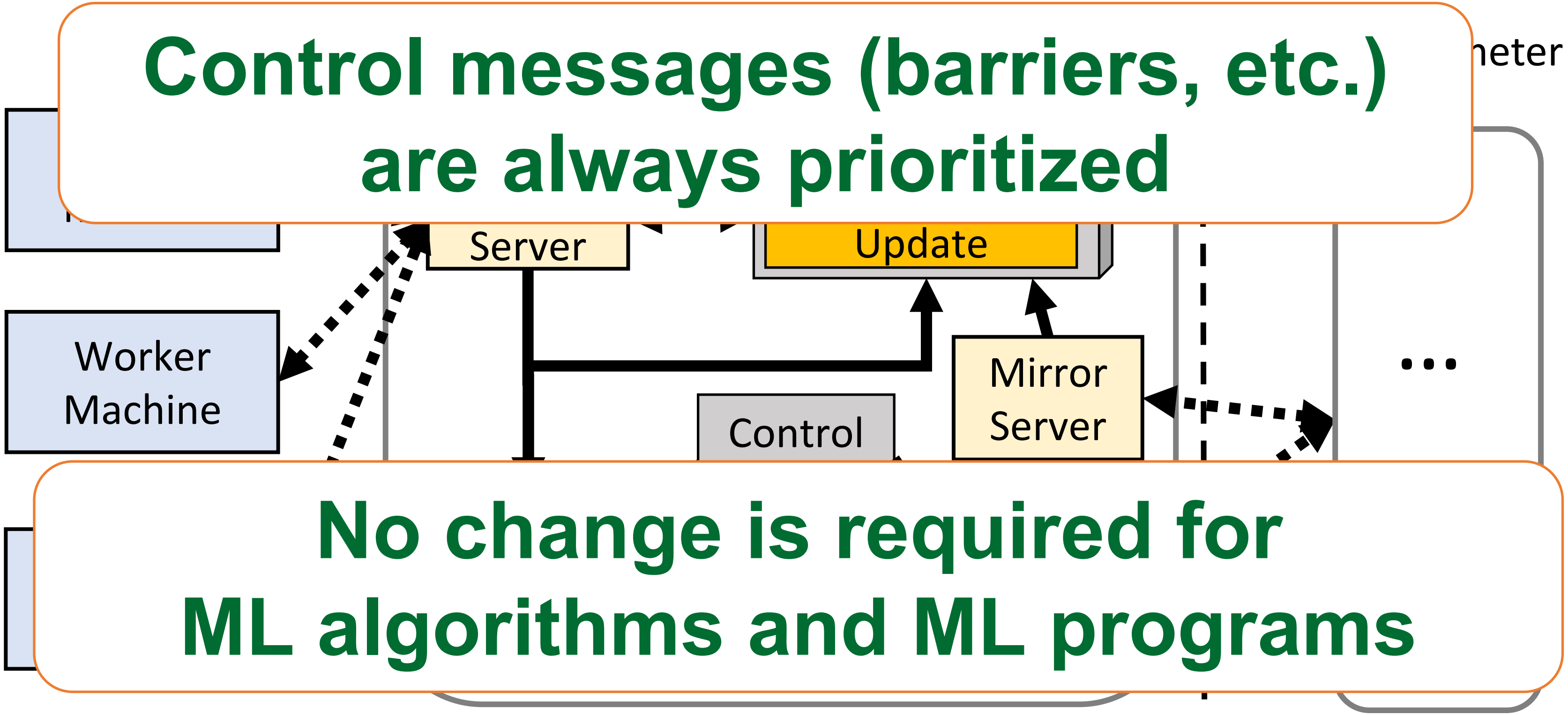
Outline

- Problem & Goal
- Background & Motivation
- Gaia System Overview
- Approximate Synchronous Parallel
- System Implementation
- Evaluation
- Conclusion

Put it All Together: The Gaia System



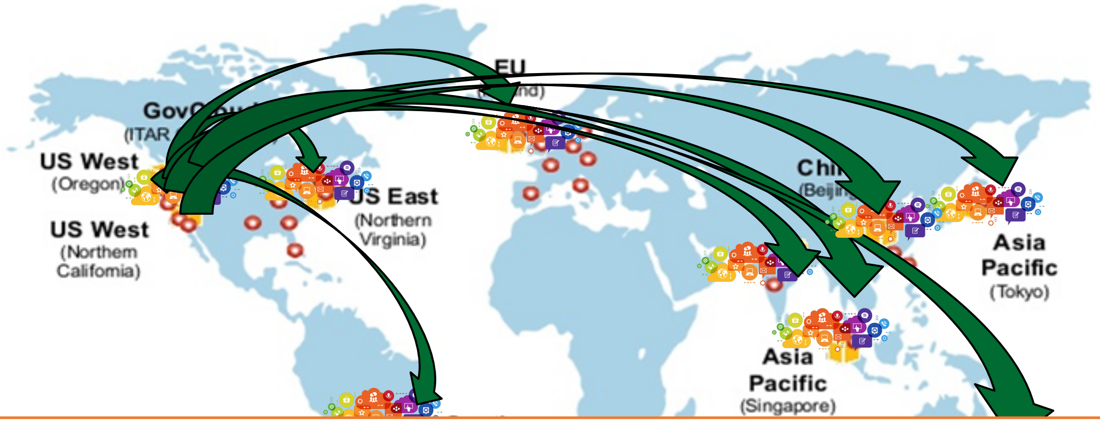
Put it All Together: The Gaia System



Control messages (barriers, etc.) are always prioritized

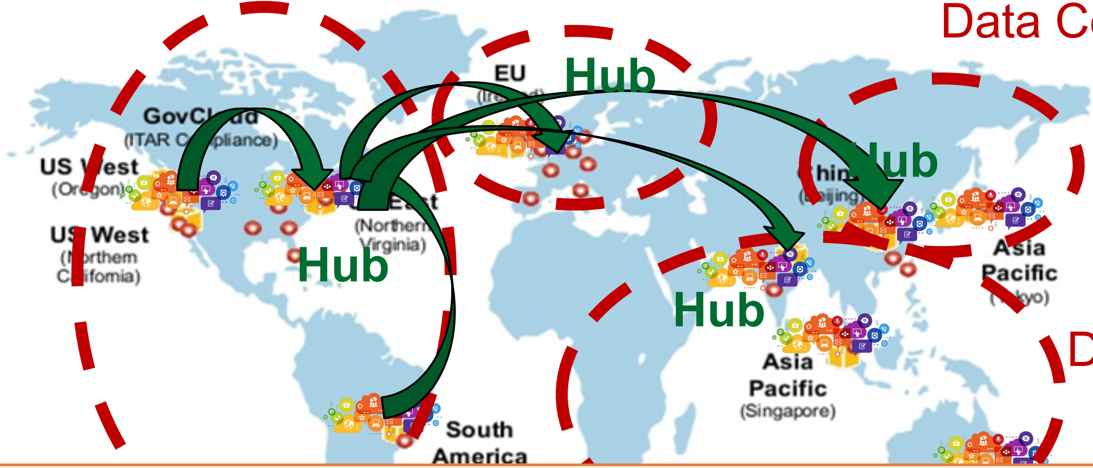
No change is required for ML algorithms and ML programs

Problem: Broadcast Significant Updates



Communication overhead is proportional to the number of data centers

Mitigation: Overlay Networks and Hubs



Data Center Group

Data Center Group

Data Center Group

Data Center Group

Save communication on WANs by aggregating the updates at hubs

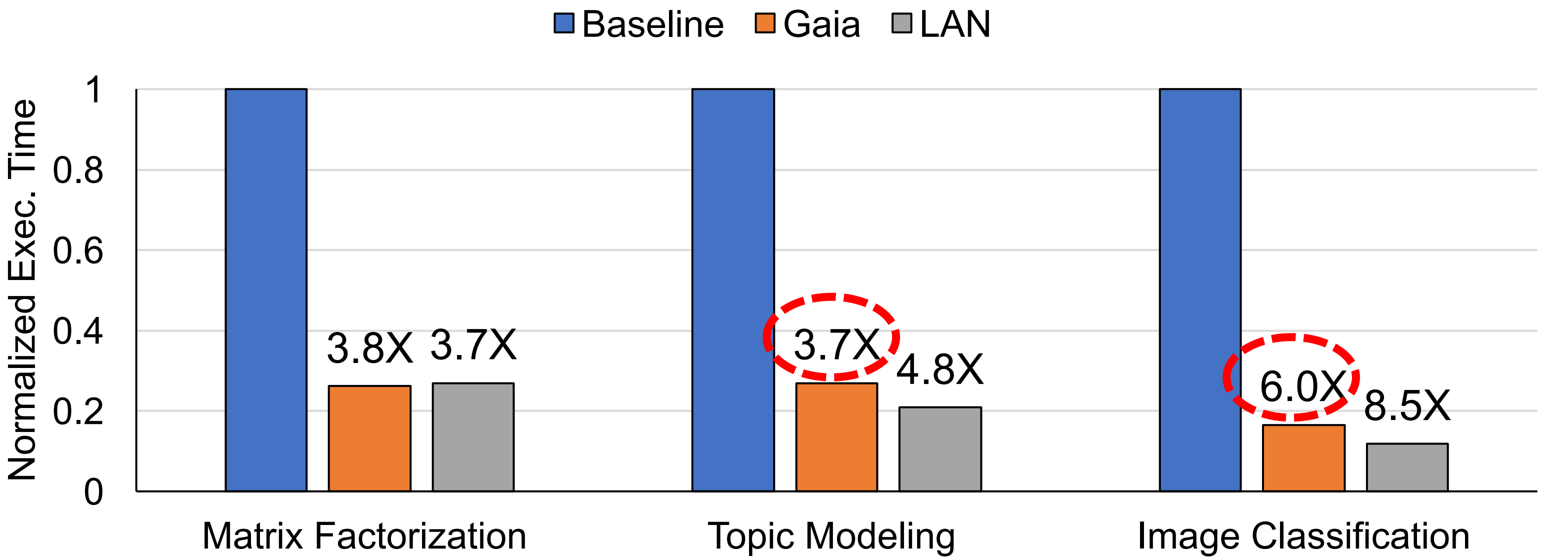
Outline

- Problem & Goal
- Background & Motivation
- Gaia System Overview
- Approximate Synchronous Parallel
- System Implementation
- Evaluation
- Conclusion

Methodology

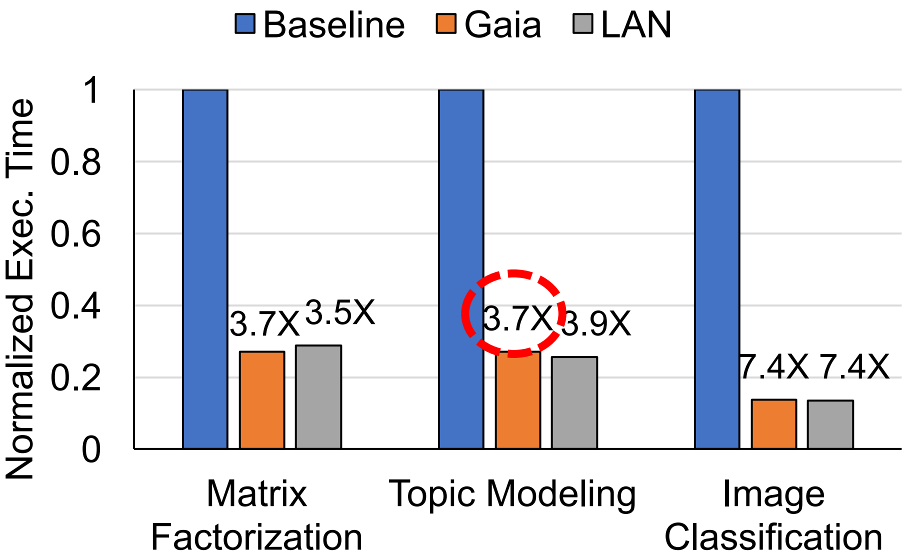
- **Applications**
  - Matrix Factorization with the *Netflix* dataset
  - Topic Modeling with the *Nytimes* dataset
  - Image Classification with the *ILSVRC12* dataset
- **Hardware platform**
  - 22 machines with emulated EC2 WAN bandwidth
  - We validated the performance with a real EC2 deployment
- **Baseline**
  - IterStore (Cui et al., SoCC'14) and GeePS (Cui et al., EuroSys'16) on WAN
- **Performance metrics**
  - Execution time until algorithm convergence
  - Monetary cost of algorithm convergence

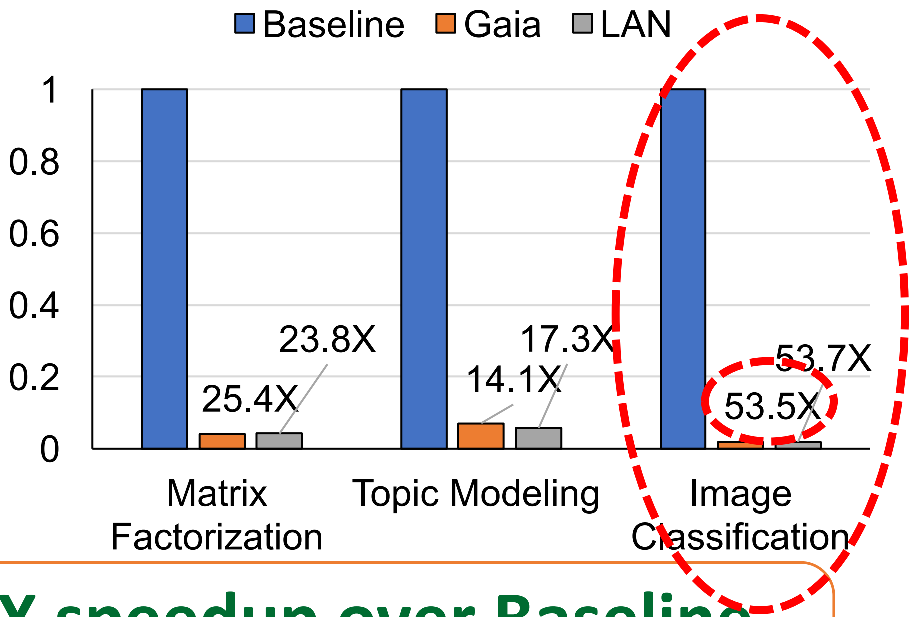
Performance – 11 EC2 Data Centers



Gaia achieves 3.7-6.0X speedup over Baseline
Gaia is at most 1.40X of LAN speeds

Performance and WAN Bandwidth

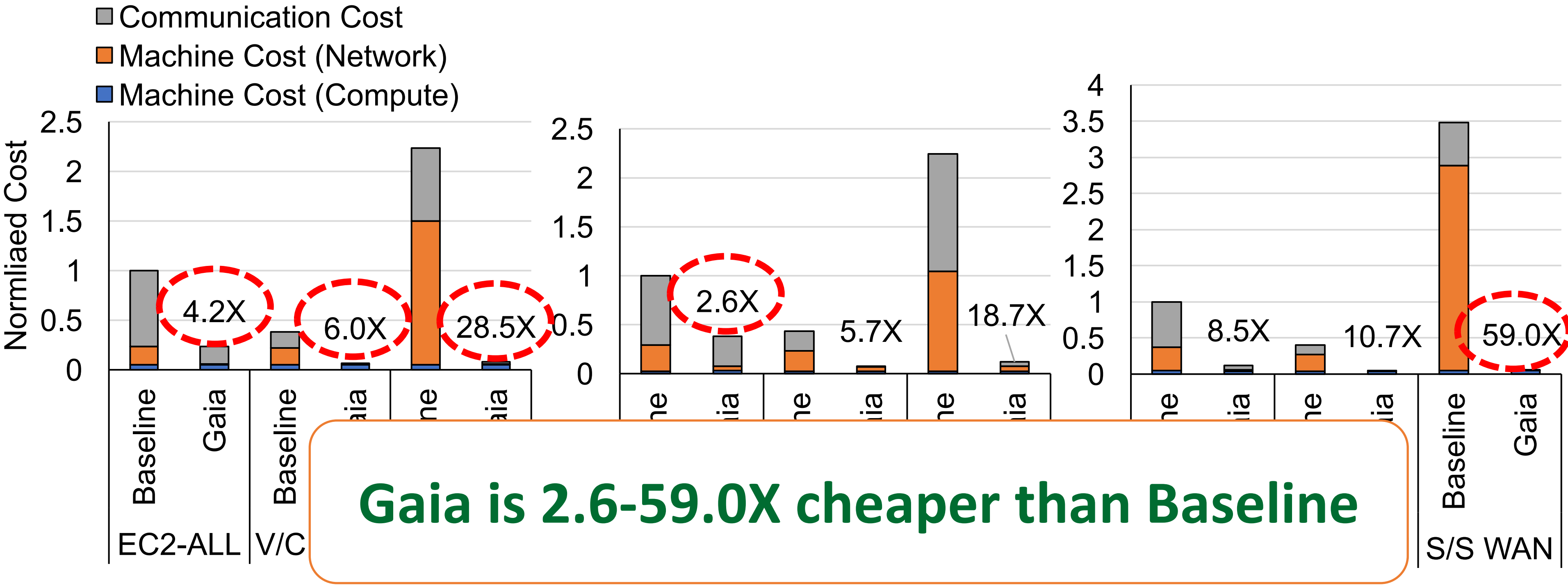




Gaia achieves 3.7-53.5X speedup over Baseline

Gaia is at most 1.23X of LAN speeds

Results – EC2 Monetary Cost



Gaia is 2.6-59.0X cheaper than Baseline

Matrix Factorization **Topic Modeling** **Image Classification**

More in the Paper

- **Convergence proof** of Approximate Synchronous Parallel (ASP)

- ASP vs. fully asynchronous

- Gaia vs. centralizing data approach

Key Takeaways

- **The Problem**: **How to perform ML on geo-distributed data?**
  - Centralizing data is infeasible. Geo-distributed ML is very slow
- **Our Gaia Approach**
  - **Decouple the synchronization model** within the data center from that across data centers
    - **Eliminate insignificant updates across data centers**
  - A **new synchronization model:** Approximate Synchronous Parallel
    - **Retain the correctness and accuracy of ML algorithms**
- **Key Results**:
  - **1.8-53.5X speedup** over state-of-the-art ML systems on WANs
  - **at most 1.40X** of LAN speeds
  - **without requiring changes** to algorithms

Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds

Kevin Hsieh

Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R. Ganger, Phillip B. Gibbons, Onur Mutlu†

Carnegie Mellon University

†ETH zürich

Executive Summary

- **The Problem**: **How to perform ML on geo-distributed data?**
  - Centralizing data is infeasible. Geo-distributed ML is very slow
- **Our Goal**
  - **Minimize communication** over WANs
  - Retain the **correctness and accuracy** of ML algorithms
  - **Without requiring changes** to ML algorithms
- **Our Gaia Approach**
  - **Decouple the synchronization model** within the data center from that across data centers: Eliminate insignificant updates on WANs
  - A **new synchronization model:** Approximate Synchronous Parallel
- **Key Results**:
  - **1.8-53.5X speedup** over state-of-the-art ML systems on WANs
  - **within 1.40X** of LAN speeds

Approximate Synchronous Parallel

The significance filter

- Filter updates based their significance

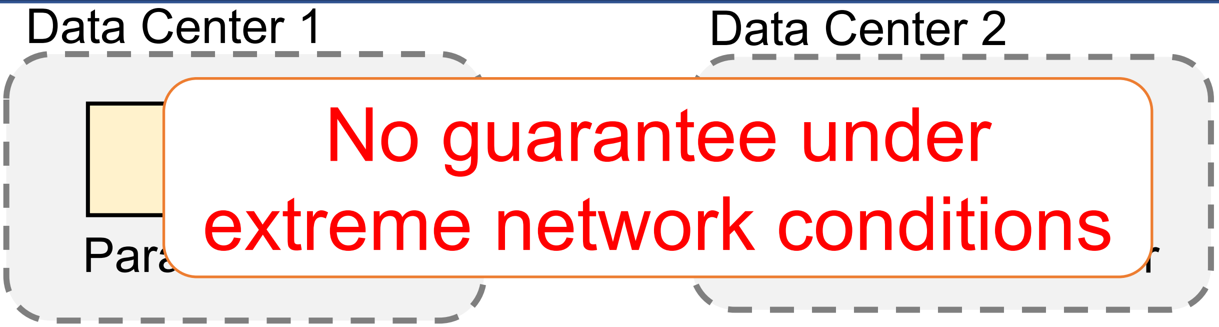
ASP selective barrier

- Ensure significant updates are read in time

Mirror clock

- Safeguard for pathological cases

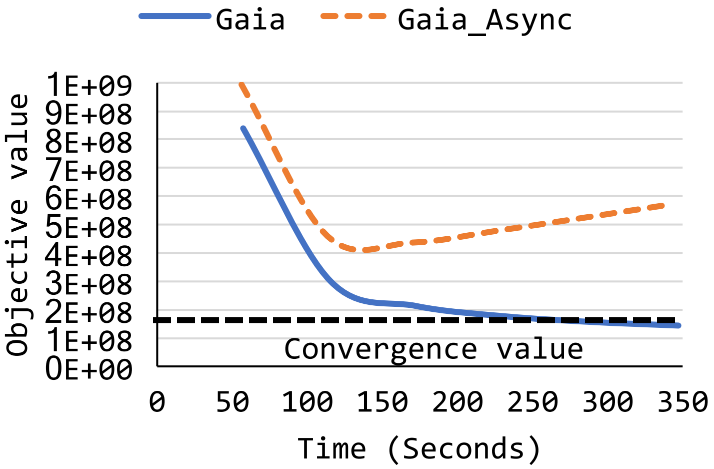
Mirror Clock



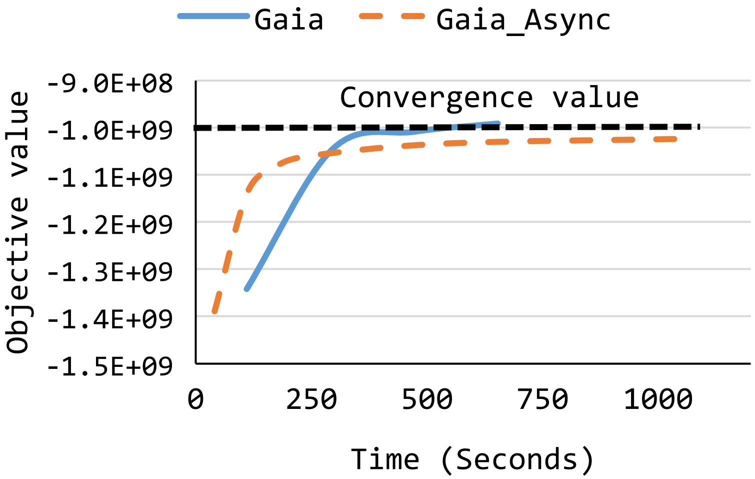
No guarantee under extreme network conditions

Guarantees all significant updates are seen after *DS* clocks

Effect of Synchronization Mechanisms



Matrix Factorization



Topic Modeling

Methodology Details

- **Hardware**
    - A 22-node cluster. Each has a 16-core Intel Xeon CPU (E5-2698), a NVIDIA Titan X GPU, 64GB RAM, and a 40GbE NIC
- **Application details**
    - Matrix Factorization: SGD algorithm, 500 ranks
    - Topic Modeling: Gibbs sampling, 500 topics
- **Convergence criteria**
    - The value of the objective function changes less than 2% over the course of 10 iterations
- **Significance Threshold**
    - 1% and shrinks over time $\left(\frac{1\%}{\sqrt{T}}\right)$

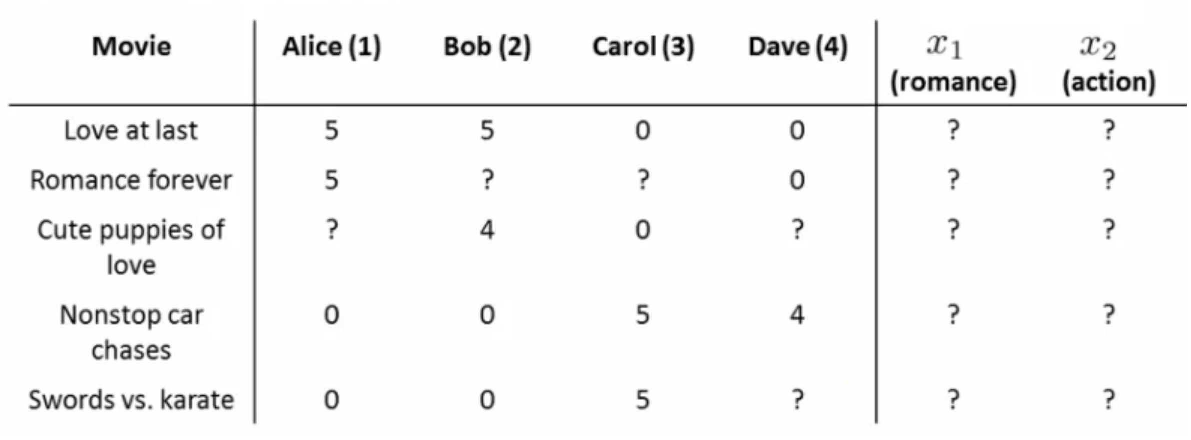
ML System Performance Comparison

- IterStore [Cui et al. SoCC’15] shows 10X performance improvement over PowerGraph [Gonzalez et al., OSDI’12] for Matrix Factorization

- PowerGraph matches the performance of GraphX [Gonzalez et al., OSDI’14], a Spark-based system

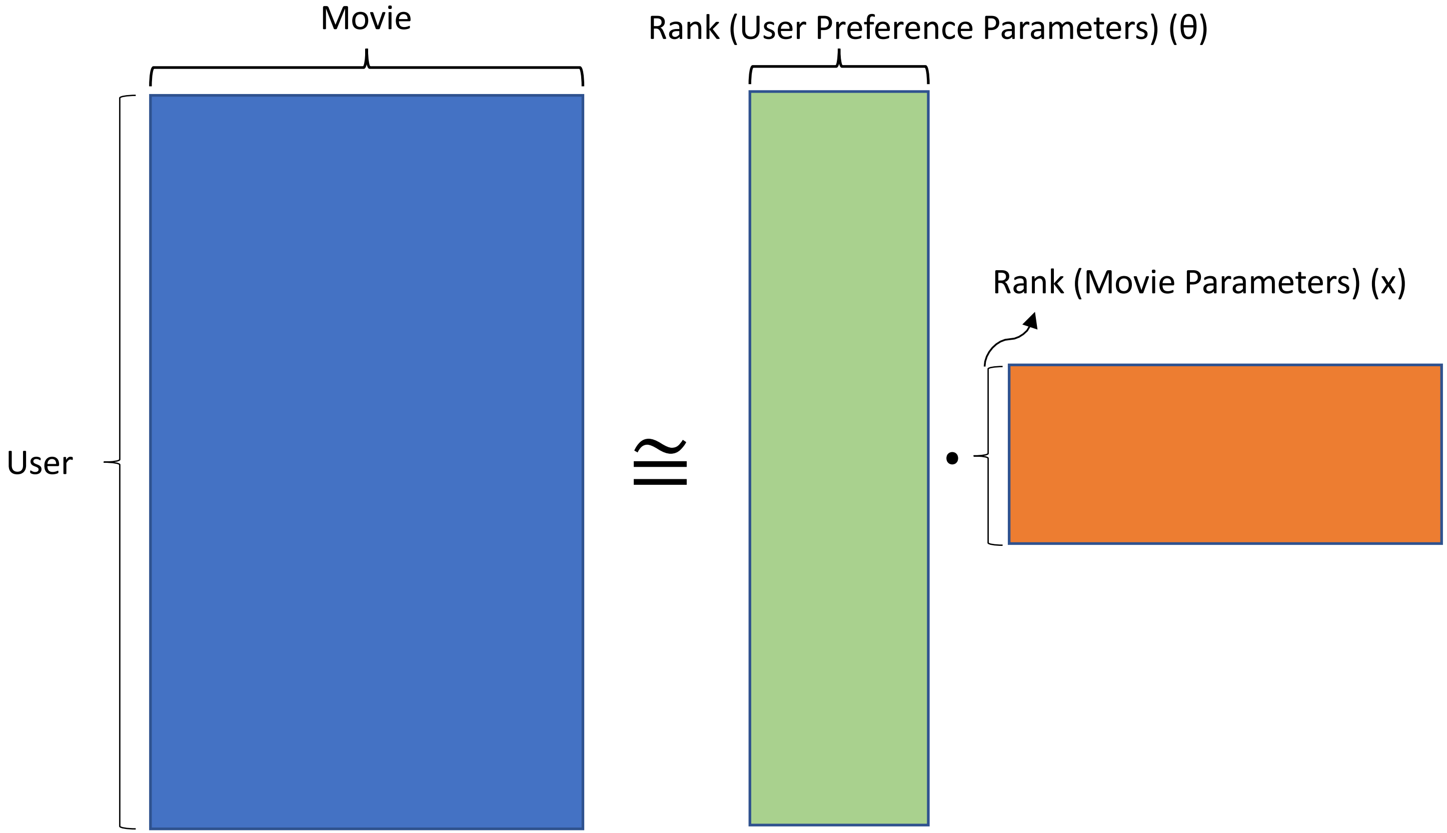
Matrix Factorization (1/3)

- Matrix factorization (also known as collaborative filtering) is a technique commonly used in recommender systems



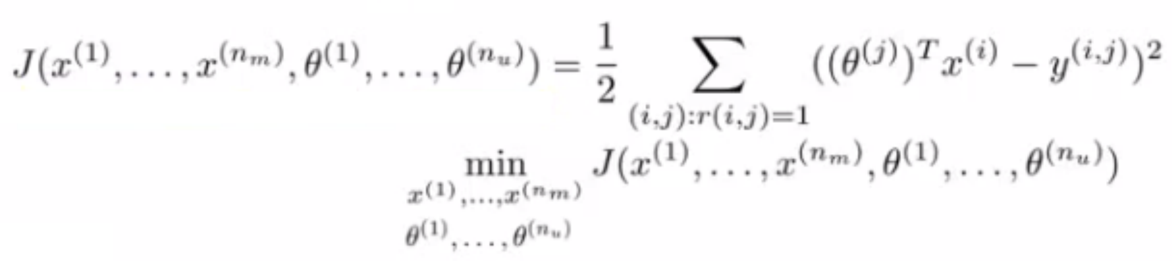
Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

Matrix Factorization (2/3)


Movie
Rank (User Preference Parameters) (θ)
Rank (Movie Parameters) (x)
User
≅
•

Matrix Factorization (3/3)

- Objective function (L2 regularization)



$$J(x^{(1)},\ldots,x^{(n_m)},\theta^{(1)},\ldots,\theta^{(n_u)}) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2$$

$$\min_{\substack{x^{(1)},\ldots,x^{(n_m)}\\ \theta^{(1)},\ldots,\theta^{(n_u)}}} J(x^{(1)},\ldots,x^{(n_m)},\theta^{(1)},\ldots,\theta^{(n_u)})$$

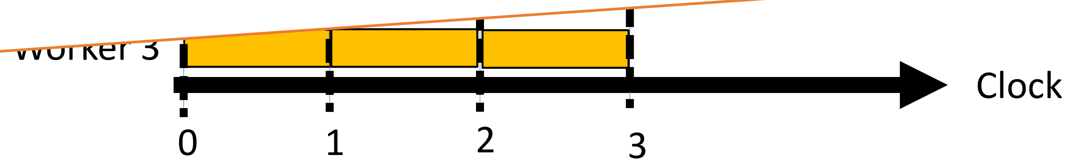
- Solve with stochastic gradient decent (SGD)

Background – BSP

- BSP (Bulk Synchronous Parallel)
  - All machines need to receive all updates before proceeding to the next iteration

Always sends and reads all updates

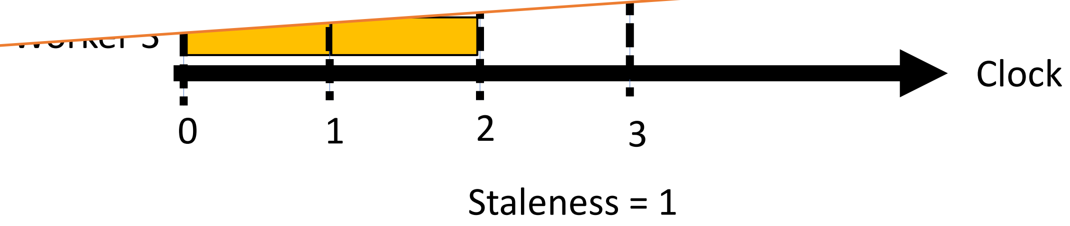
Accurate but slow



Background – SSP

- SSP (Stale Synchronous Parallel)
  - Allows the fastest worker ahead of the slowest worker by a bounded number of iterations

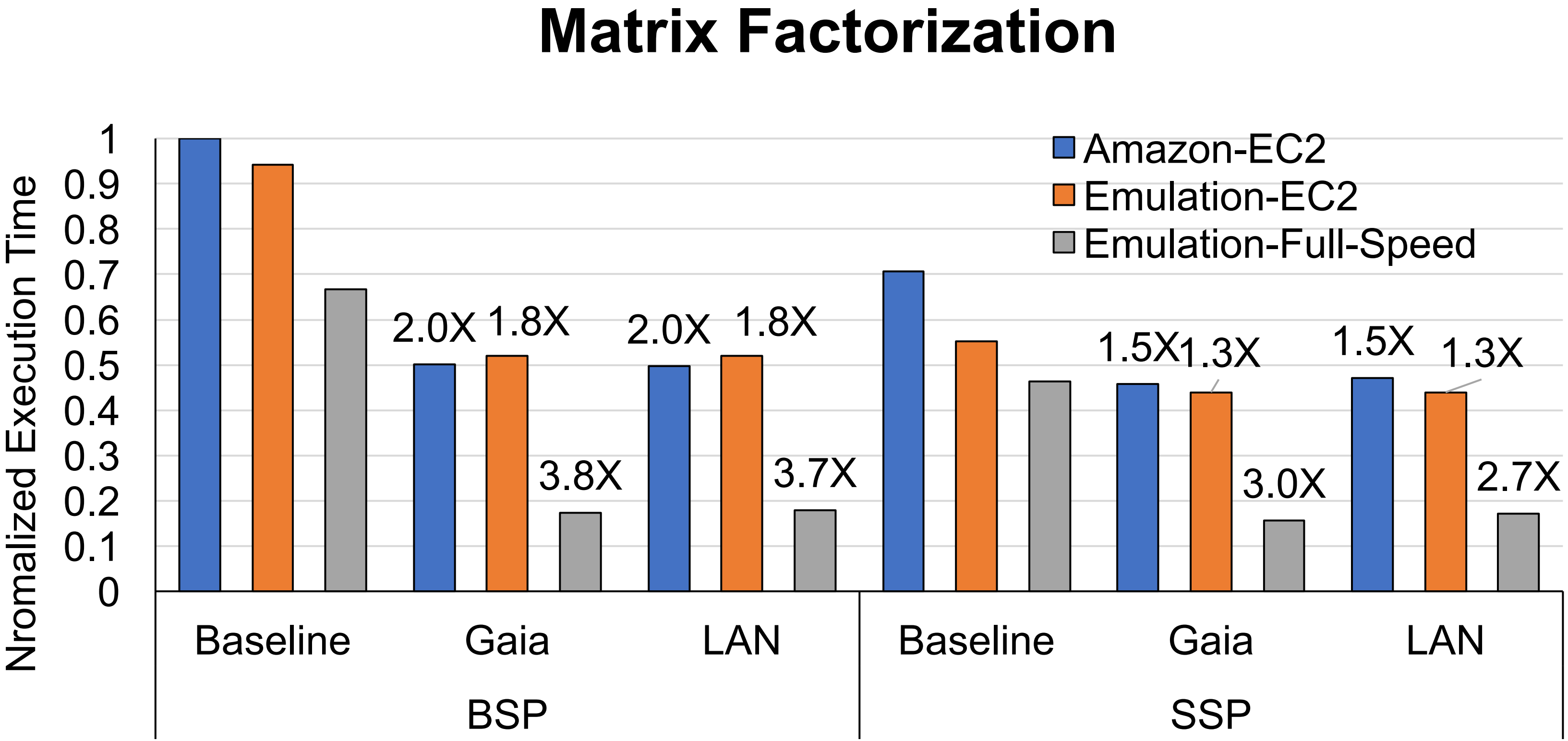
Alleviates the network bandwidth requirement,
but still sends all updates



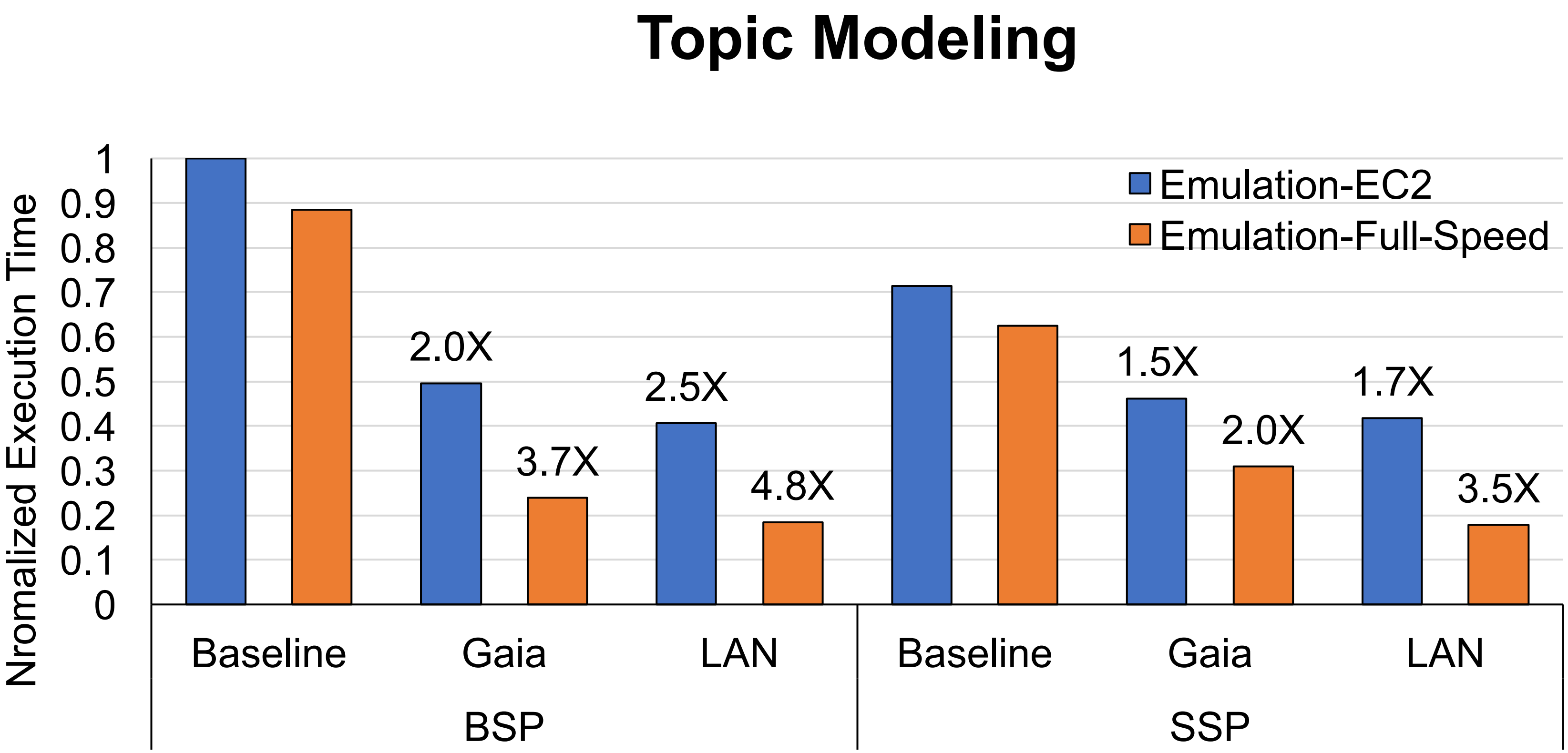
Compare Against Centralizing Approach

		Gaia Speedup over Centralize	Gaia to Centralize Cost Ratio
Matrix Factorization	EC2-ALL	1.11	3.54
	V/C WAN	1.22	1.00
	S/S WAN	2.13	1.17
Topic Modeling	EC2-ALL	0.80	6.14
	V/C WAN	1.02	1.26
	S/S WAN	1.25	1.92
Image Classification	EC2-ALL	0.76	3.33
	V/C WAN	1.12	1.07
	S/S WAN	1.86	1.08

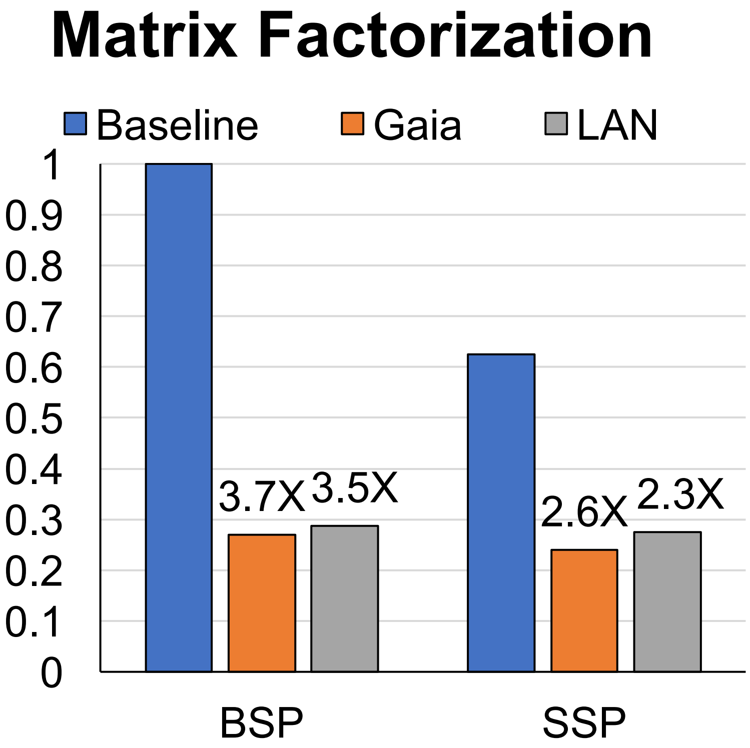
SSP Performance – 11 Data Centers

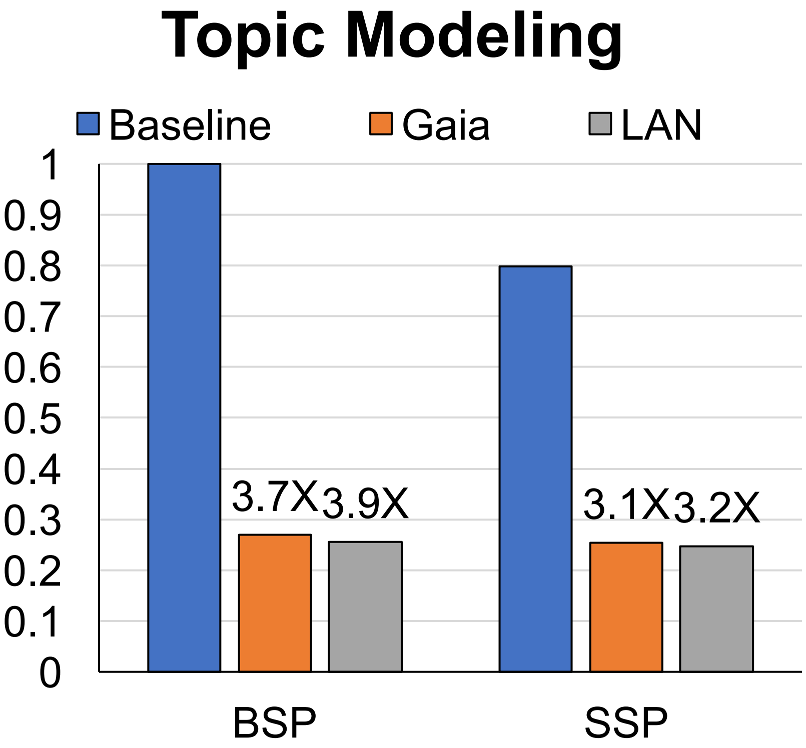


SSP Performance – 11 Data Centers

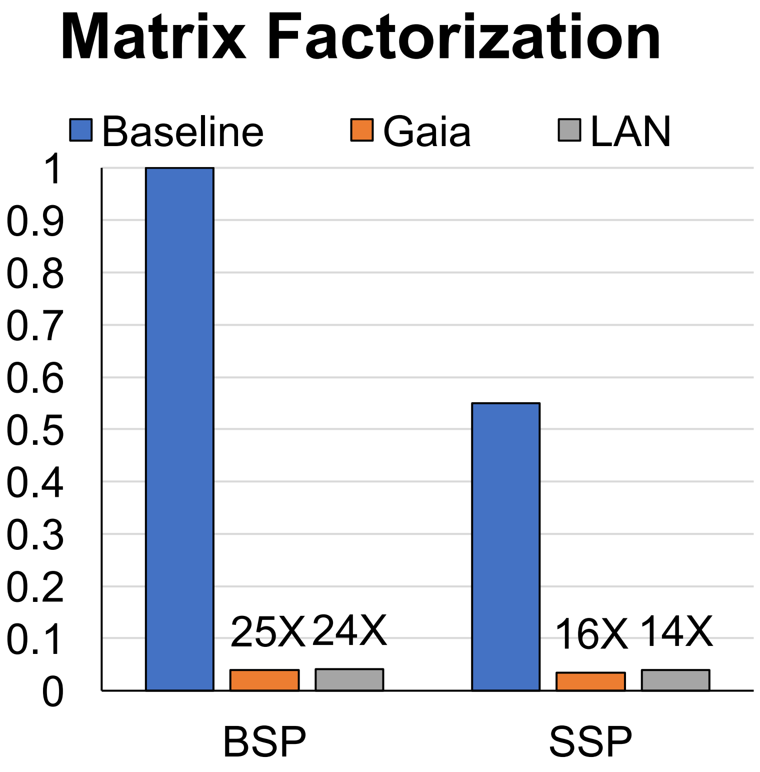


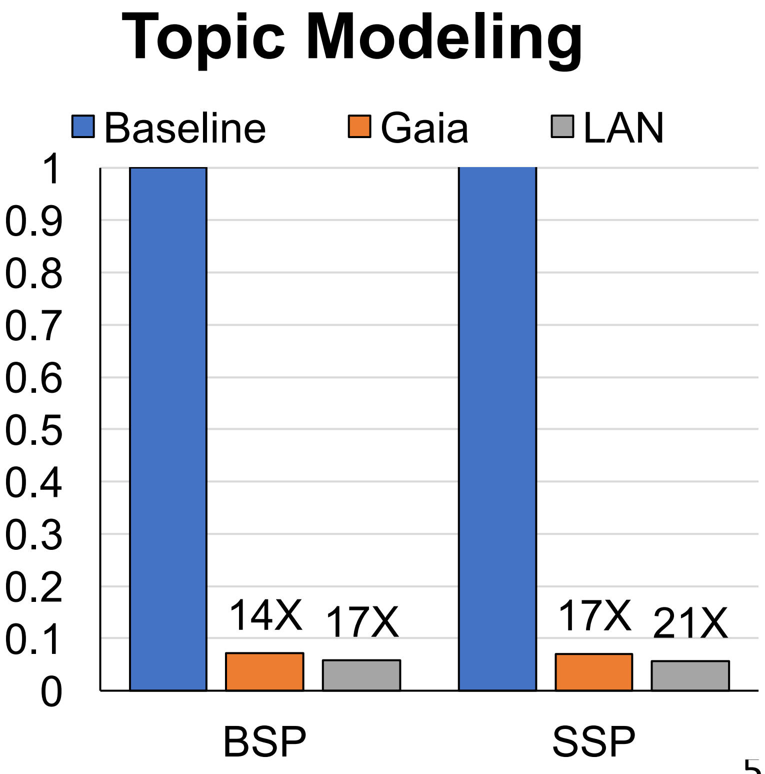
SSP Performance – V/C WAN


Matrix Factorization


Topic Modeling

SSP Performance – S/S WAN


Matrix Factorization
Baseline
Gaia
LAN
25X
24X
16X
14X
BSP
SSP


Topic Modeling
Baseline
Gaia
LAN
14X
17X
17X
21X
BSP
SSP