CLEMSON UNIVERSITY

Computing Load Aware and Long-View Load Balancing for Cluster Storage Systems

Guoxin Liu, Haiying Shen and Haoyu Wang
Holcombe Department of Electrical and Computer Engineering
Clemson University

Presented by Haoyu Wang

Outline

1, Introduction

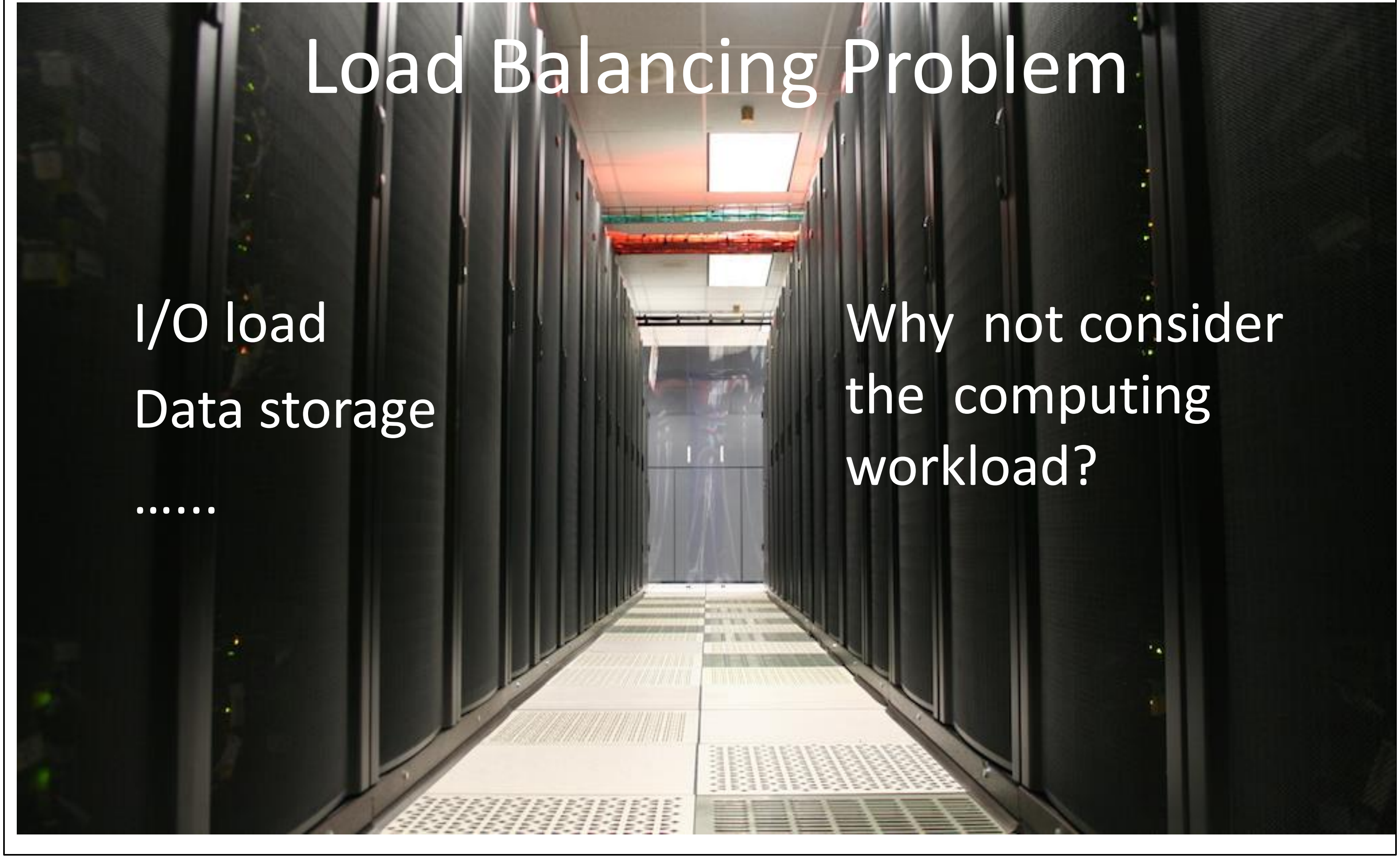
2, System design

3, Performance evaluation

4, Conclusion

Introduction

Background (Clemson Palmetto Clusters)



Load Balancing Problem

I/O load

Data storage

……

Why not consider the computing workload?

Introduction

Previous work

Previous work

- Challenge for load balancing
  - Data locality
  - Task delay
  - Long-term load balance
  - Cost-efficient & scalable
- Related work
  - Random data allocation
  - Balancing the number of data blocks
  - Balancing the I/O load

System Design

Main contribution

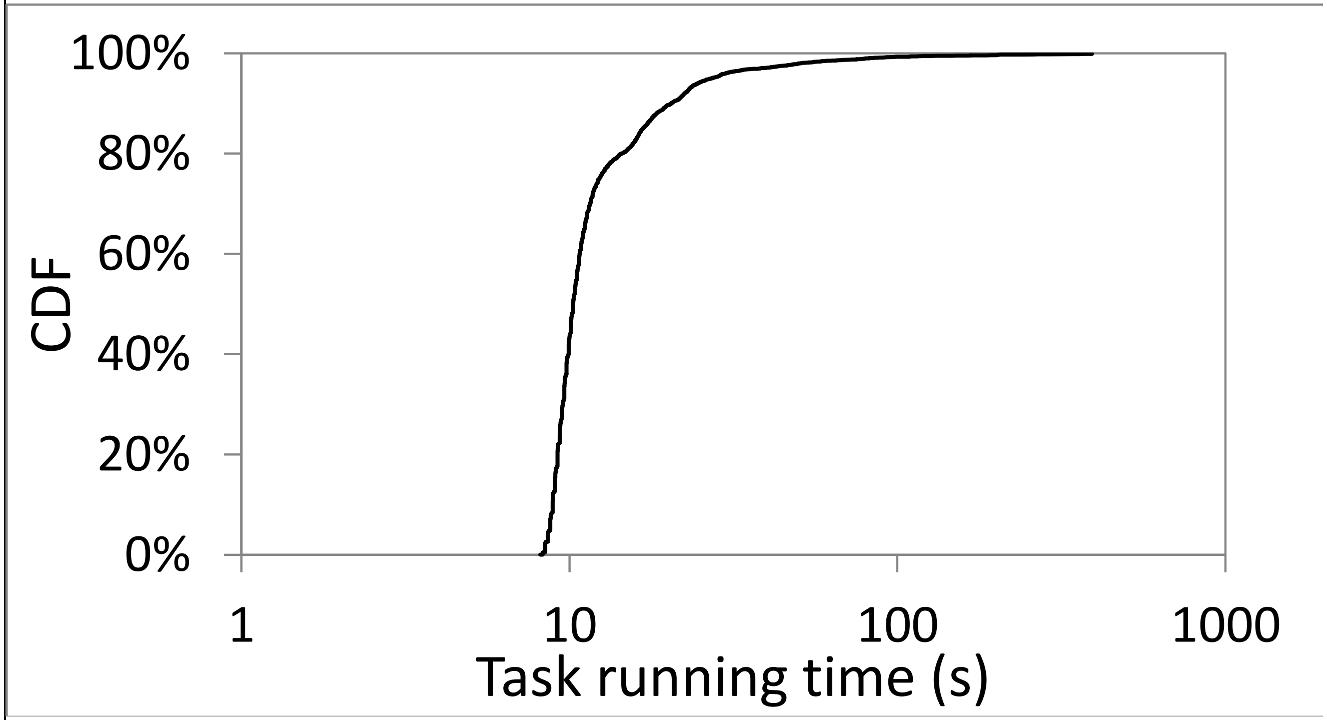
1, Trace analysis on computing workloads

2, Computing load aware long-view load balancing method

3, Trace-driven experiments

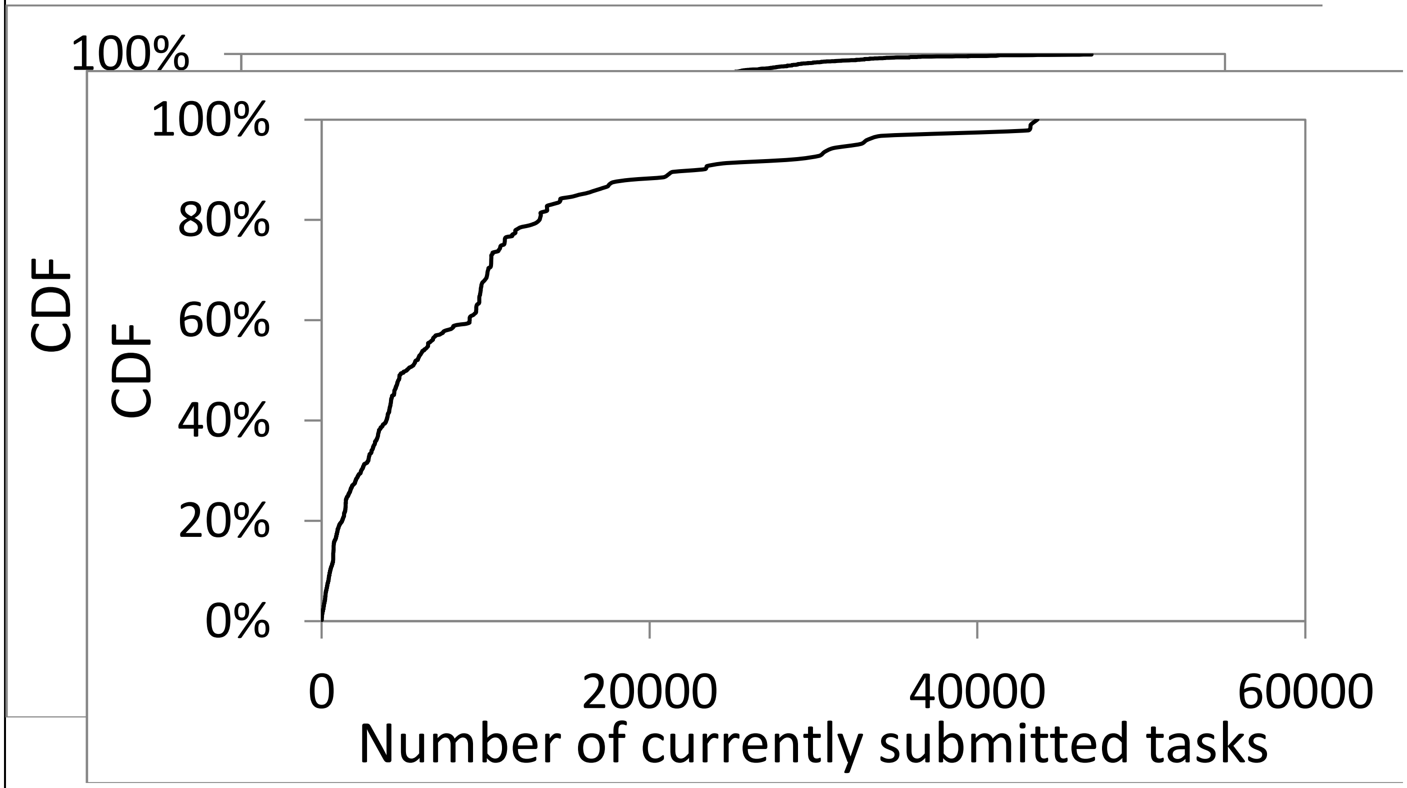
System Design

Trace Data Analysis



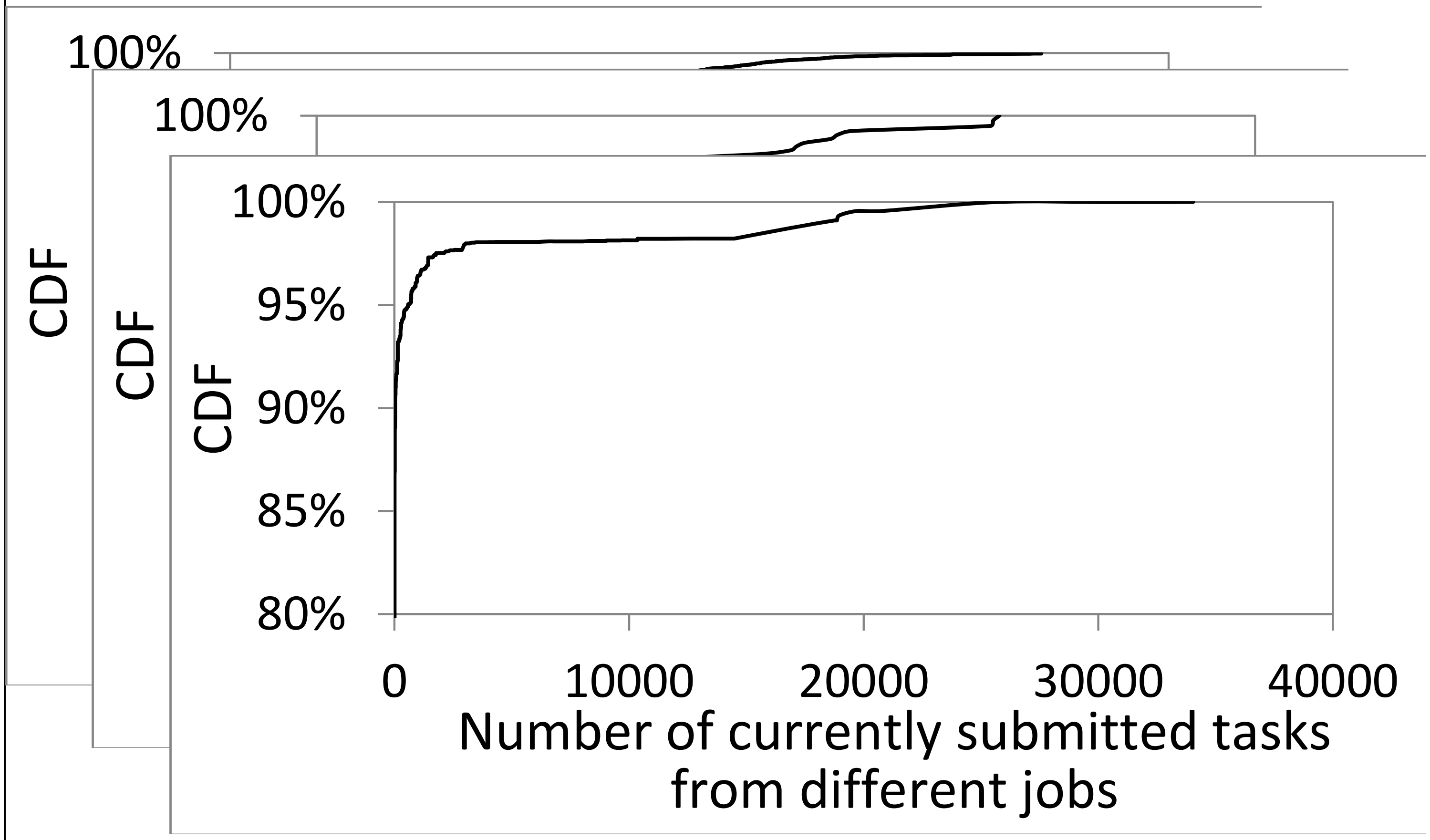
System Design

Trace Data Analysis



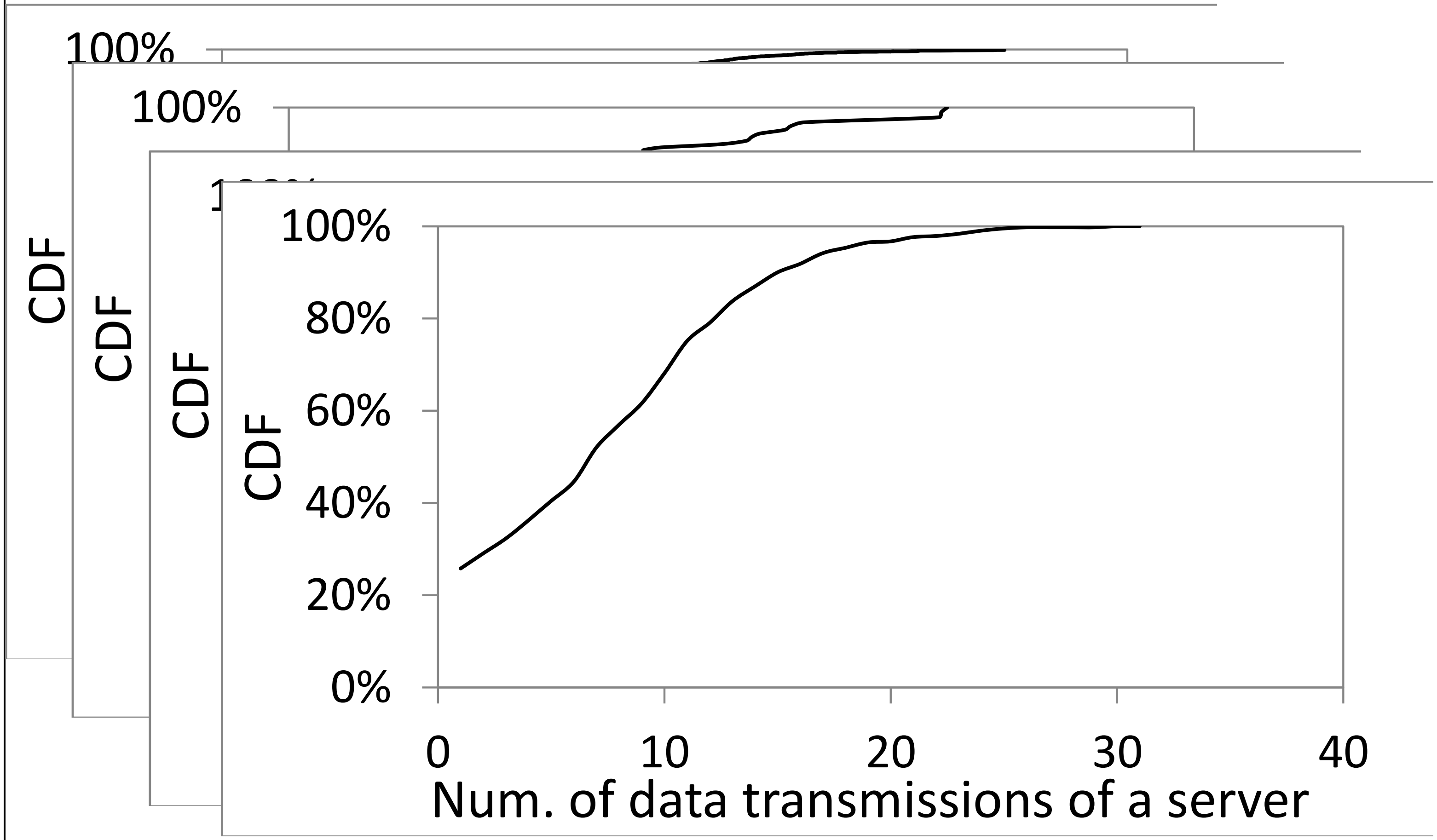
System Design

Trace Data Analysis



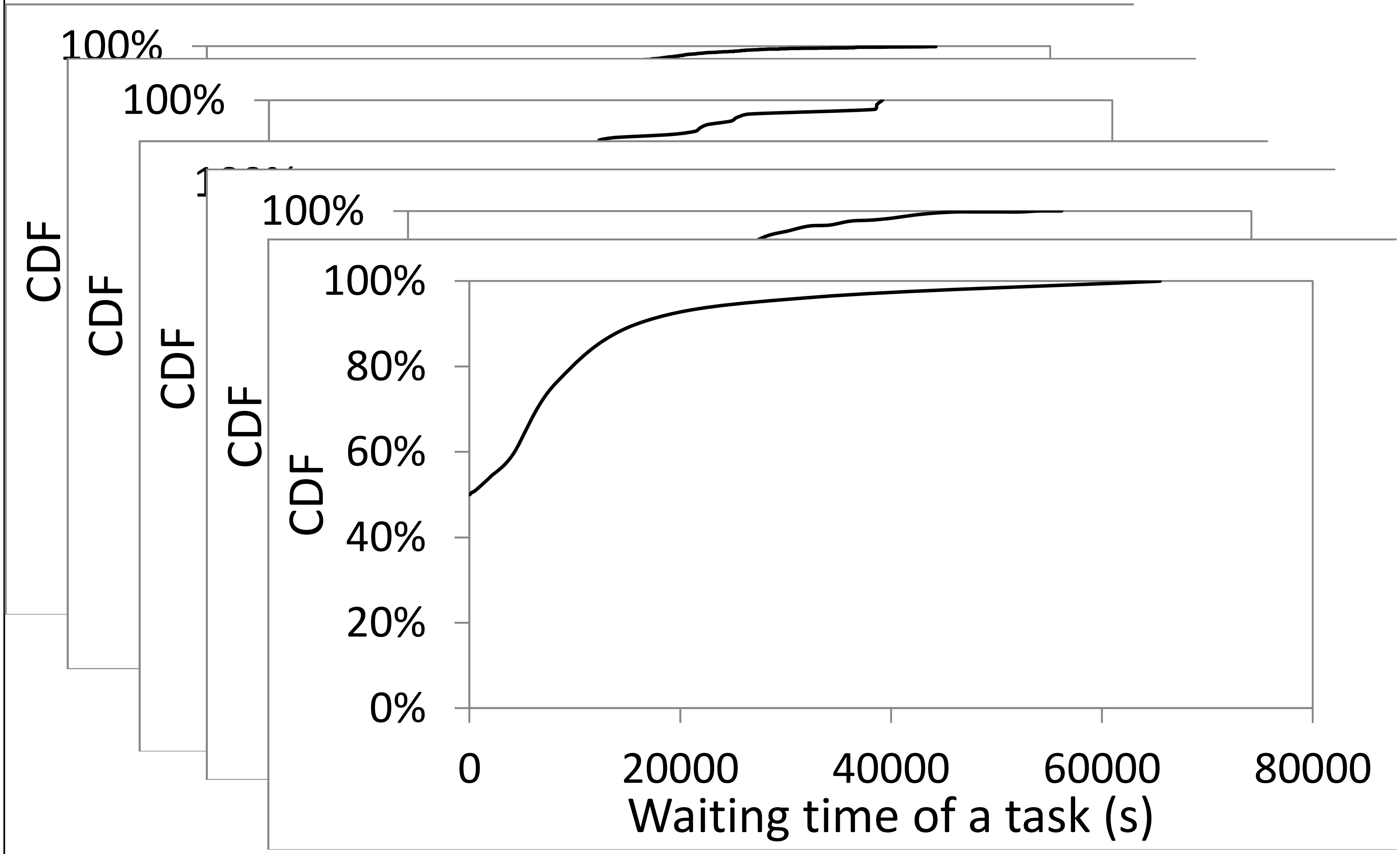
System Design

Trace Data Analysis



System Design

Trace Data Analysis



CALV System Overview

Coefficient-based data reallocation

Principle 1:

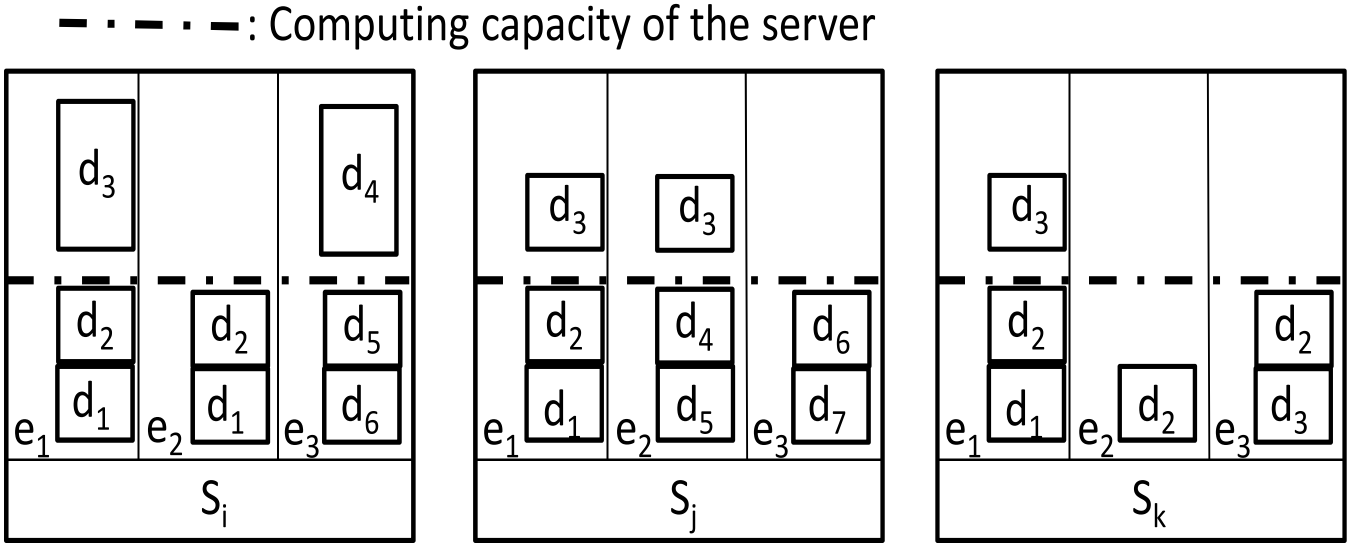
The data blocks contributing more computing workloads at more overloaded epochs in the spatial space and temporal space have a higher priority to be selected to reallocate

Principle2:

Among all data blocks contributing workloads at an overloaded epoch, the data blocks contribute less workload at more underloaded epochs have a higher priority to be selected to reallocate.

CALV System Overview

Coefficient-based data reallocation



(a) Reduce num. of reported data blocks in spatial space

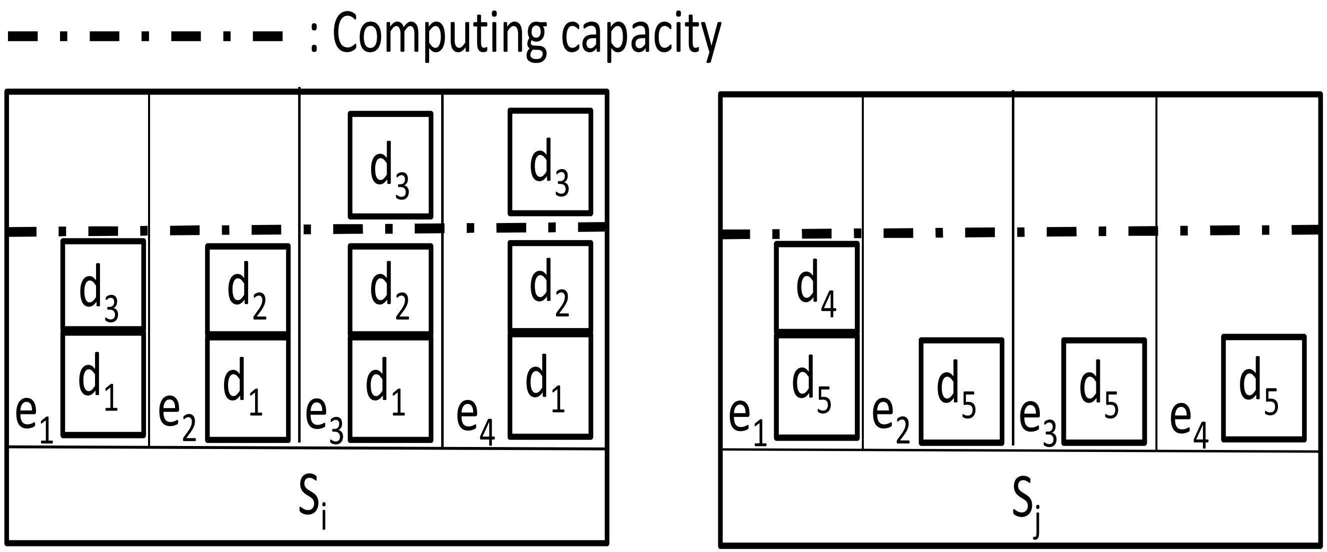
(b) Reduce num. of reported data blocks in temporal space

(c) Avoid server underload

Selection of data block to reallocate

CALV System Overview

Lazy Data Block Transmission



Lazy data block transmission

Performance Evaluation

Trace-driven experiments

Simulated environment:

3000 servers with typical fat-tree topology.

8 computing slots for each server

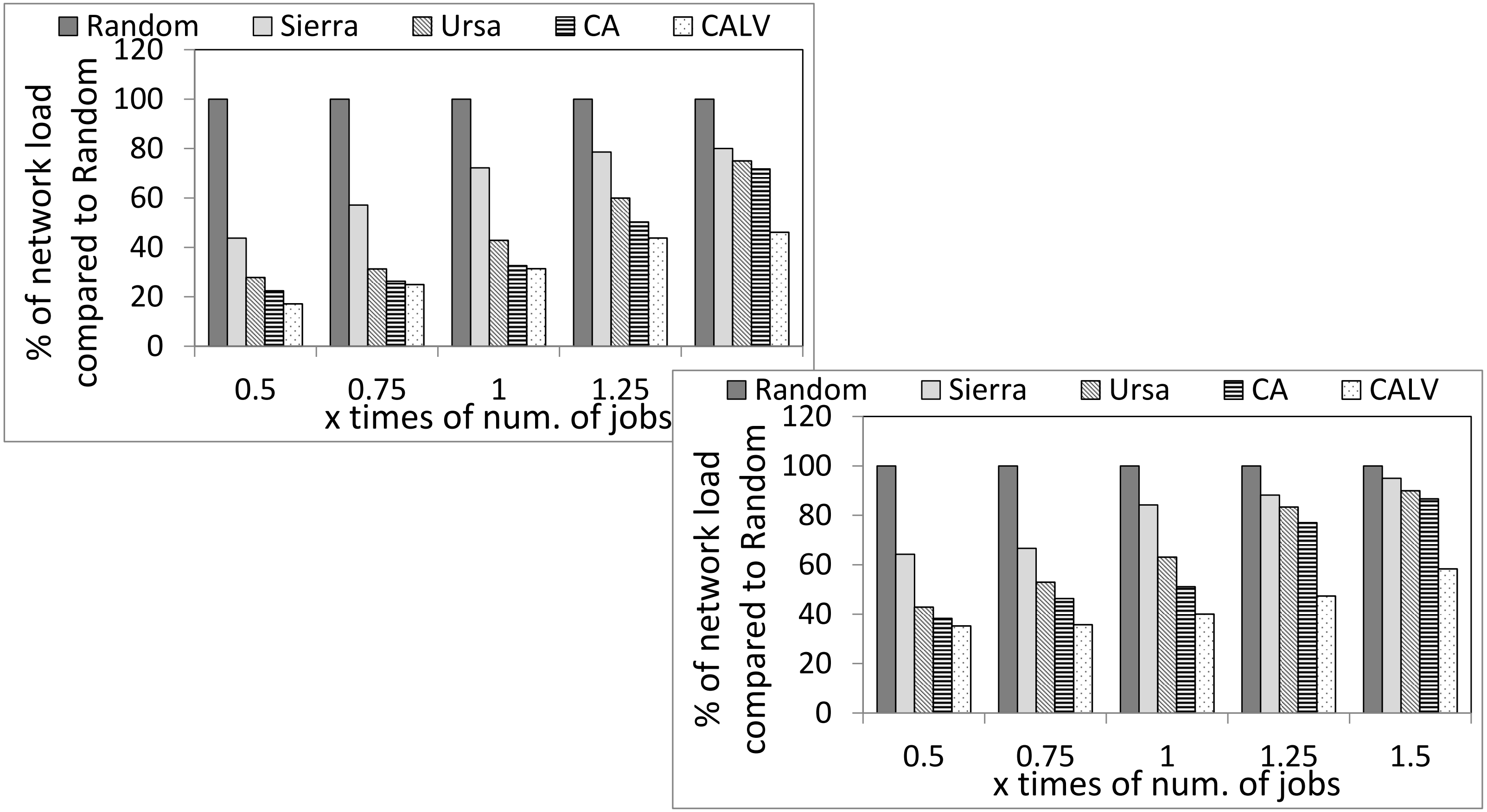
Epoch set to 1 second

Comparison method:

Random, Sierra, Ursa, CA

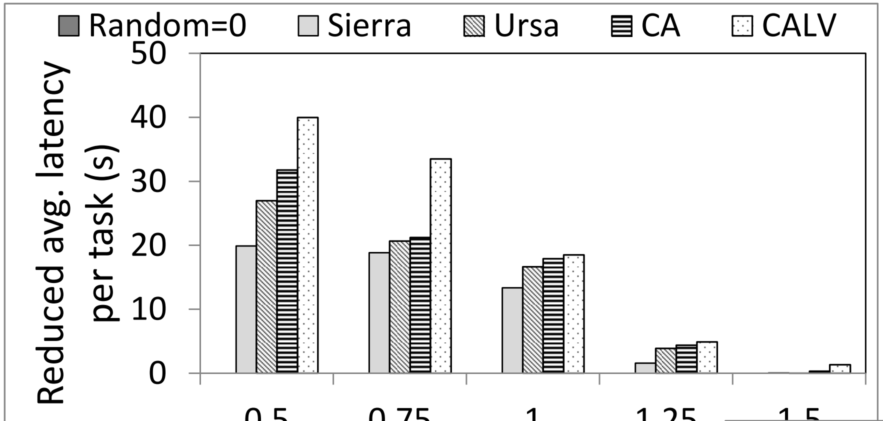
Performance Evaluation

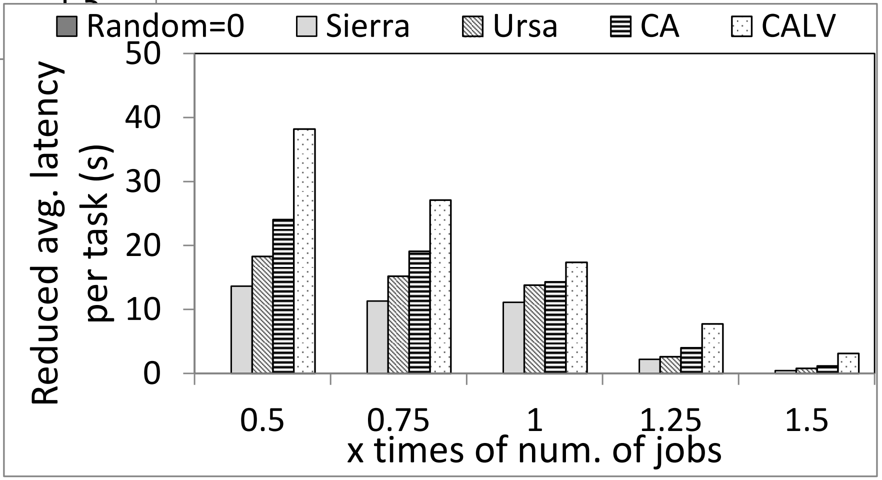
Trace-driven experiments
Performance of Data locality



Performance Evaluation

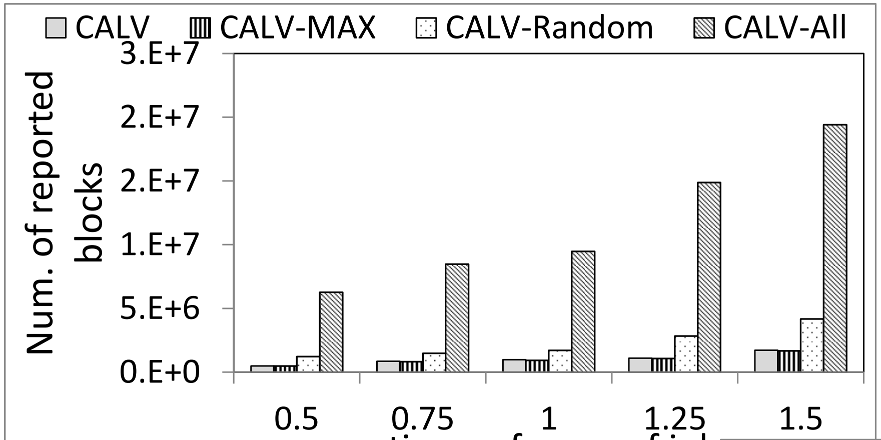
Trace-driven experiments
Performance of Task Latency



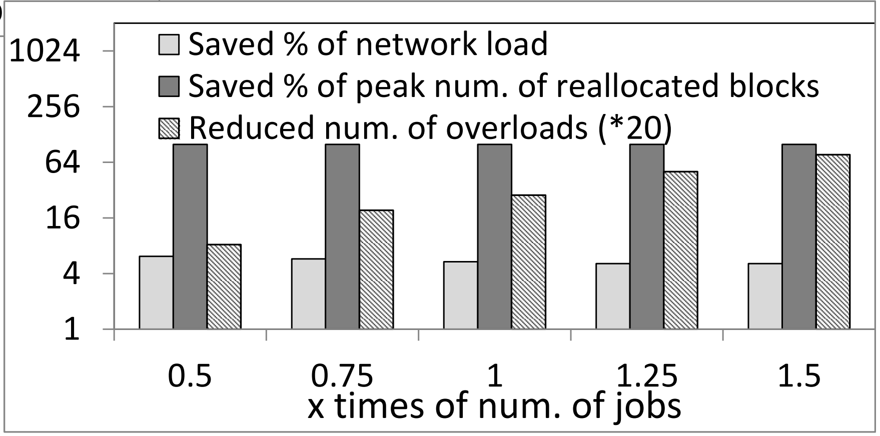


Performance Evaluation

Trace-driven experiments
Performance of Cost-Efficiency



Performance of Lazy Data transmission



Conclusion

Conclusion

The importance of considering the computing workloads

CALV is cost-efficient and could get long-term load balance

The End

Thanks!

Questions?