Scaling DeltaFS In-Situ Indexing to 131,072 CPU Cores

Qing Zheng

Chuck Cranor, George Amvrosiadis, Greg Ganger, Garth Gibson,
Brad Settlemyer, Gary Grider, Fan Guo

Carnegie Mellon University
Los Alamos National Laboratory

SC 2018

Understanding Our Universe

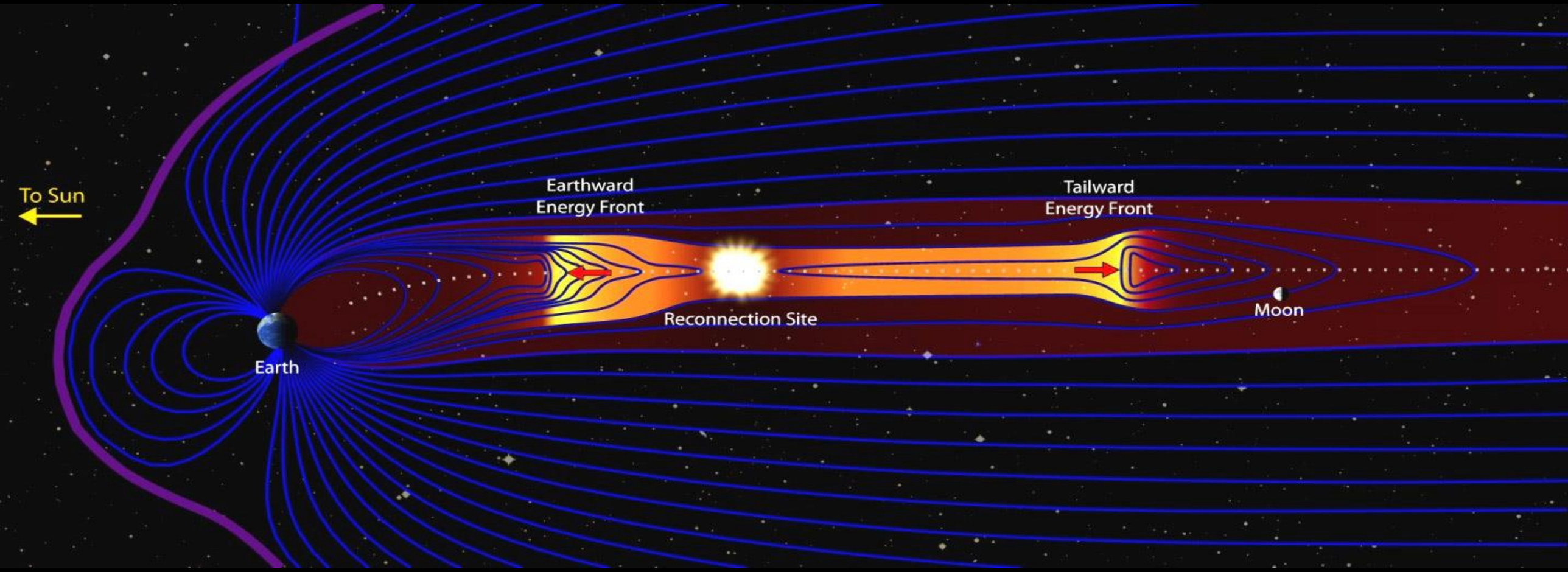


Image from http://esp.igpp.ucla.edu. Simulation movie frames from LANL https://www.lanl.gov.

Understanding Our Universe

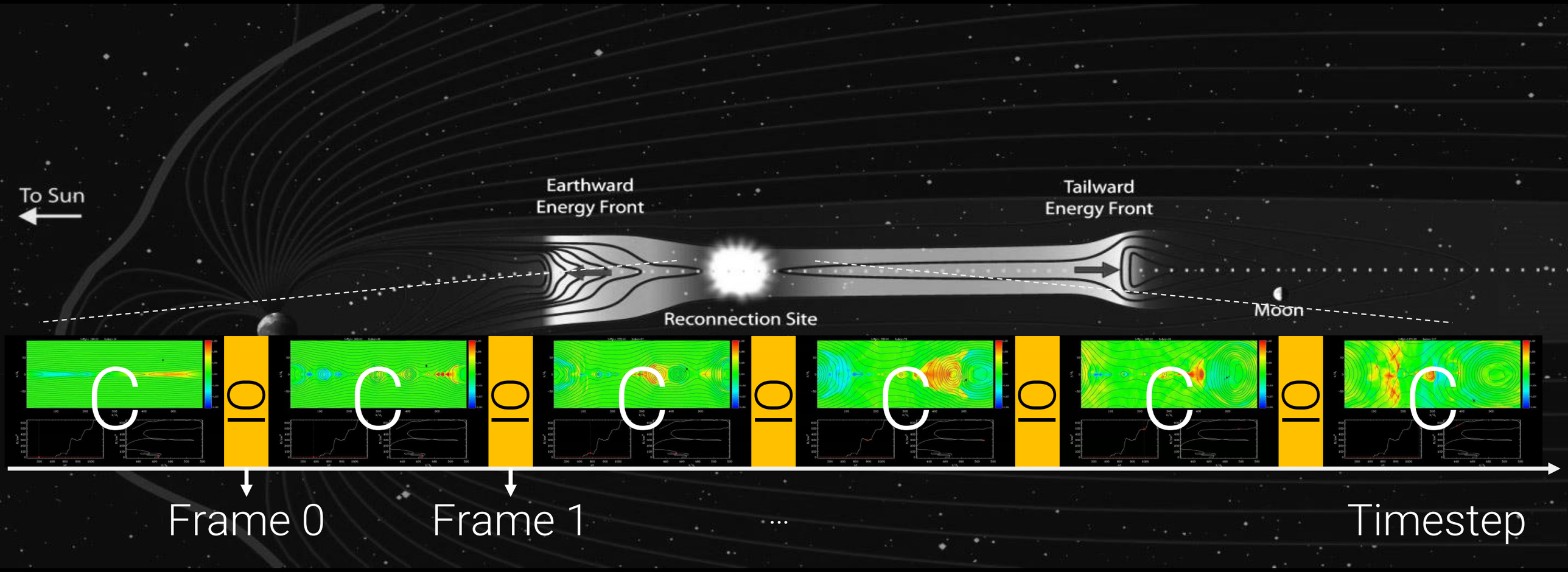
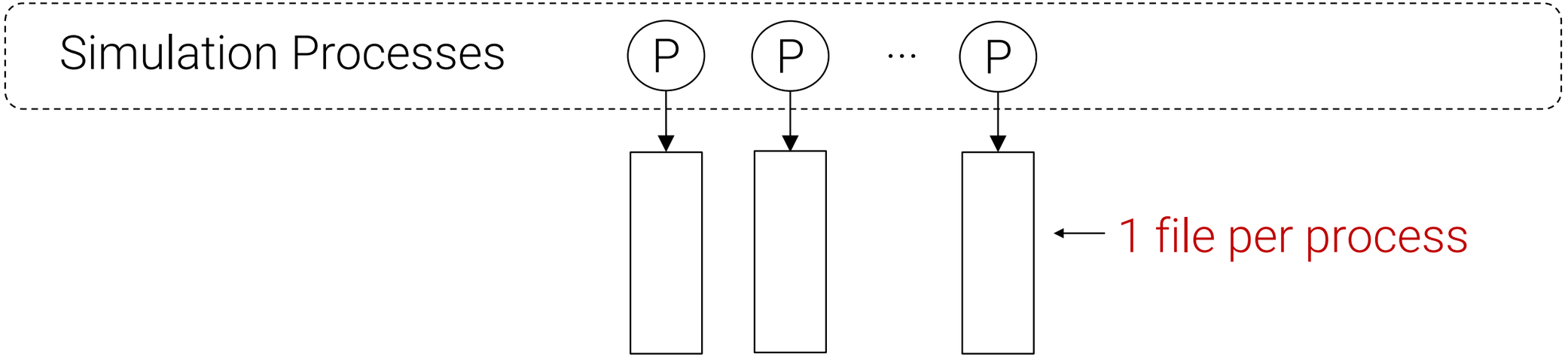


Image from http://esp.igpp.ucla.edu. Simulation movie frames from LANL https://www.lanl.gov.

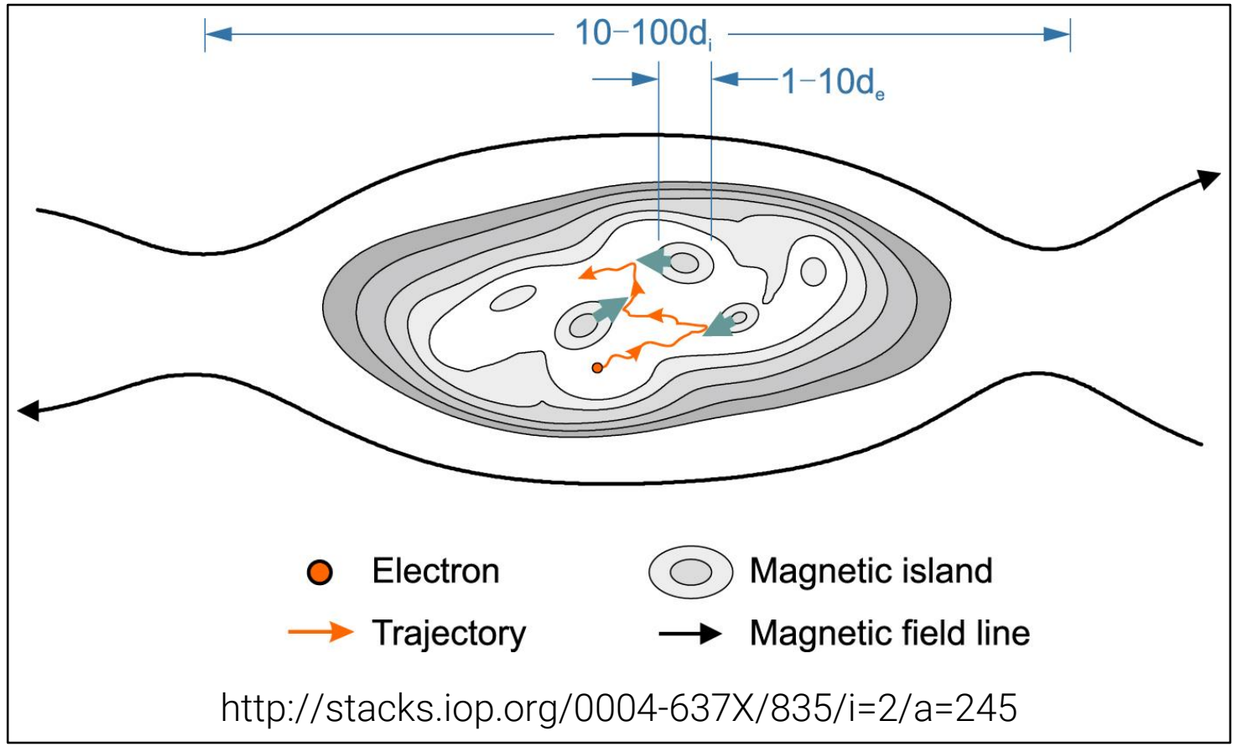
Storing Results

Frame data is written to the underlying filesystem



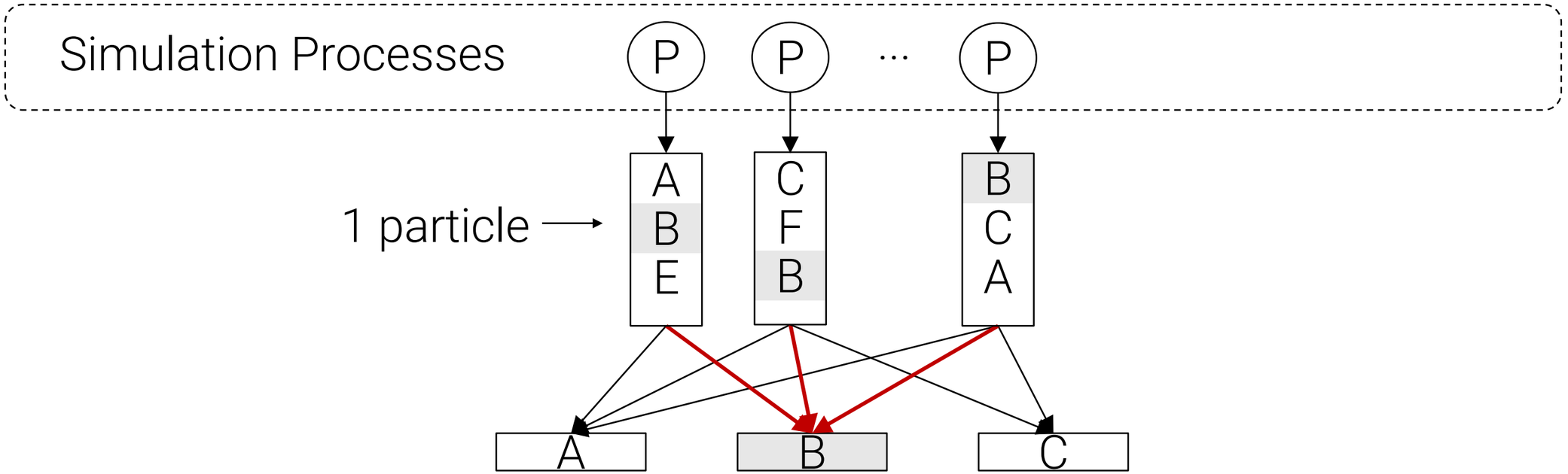
Needle In a Haystack

New analysis type: trace 1 object out of 1 trillion



Particles Move Across Processes

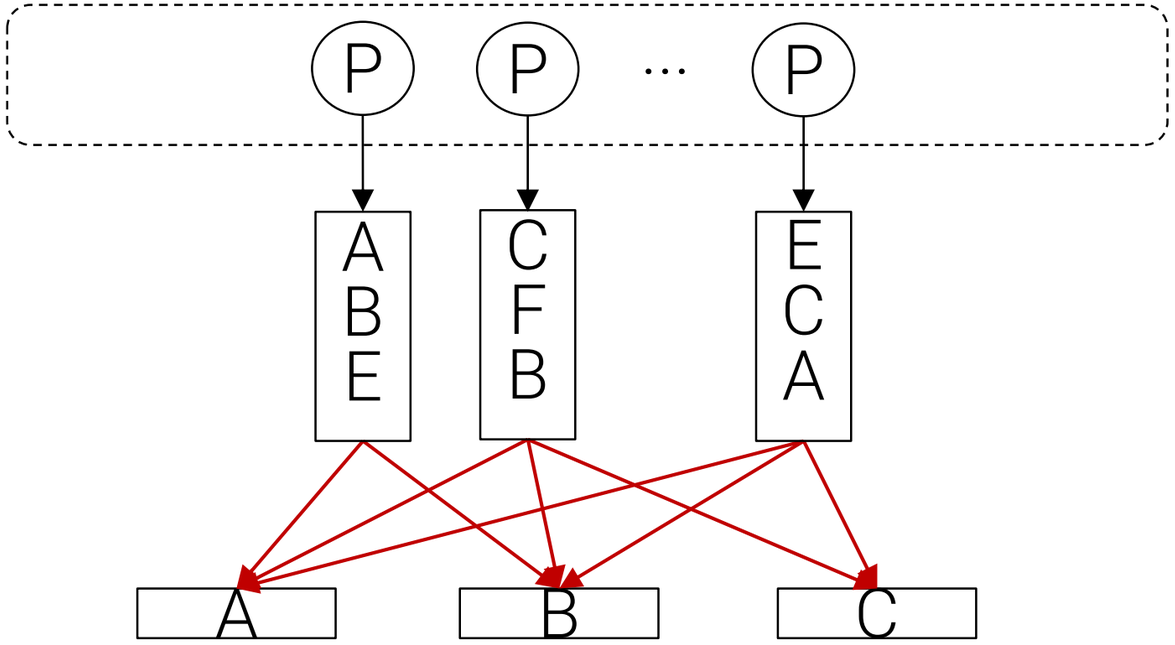
Problem: each query reads all files



Need to Do Things Differently

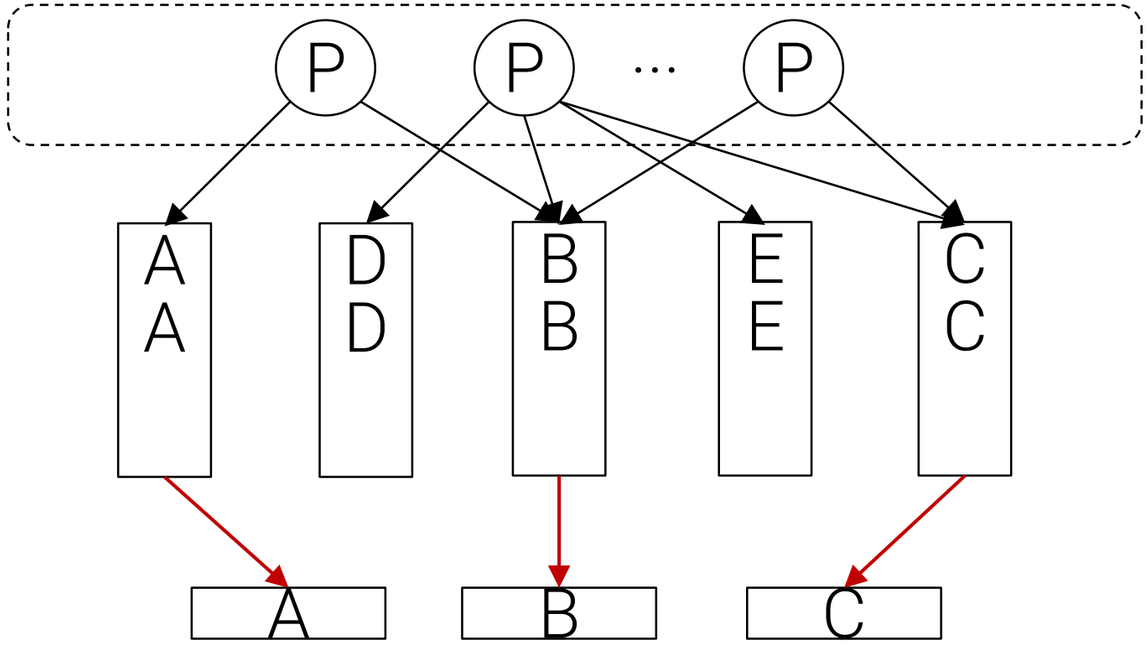
1 file per process doesn't work

Each query reads **ALL** files



Let's do 1 file per object

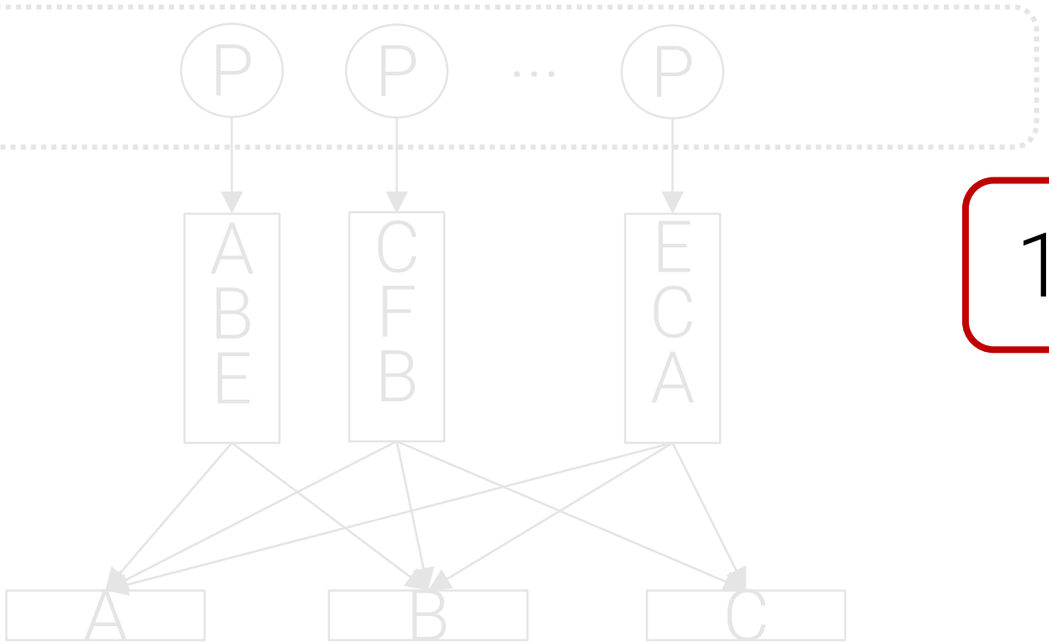
Each query reads **1** file



Need to Do Things Differently

1 file per process doesn't work

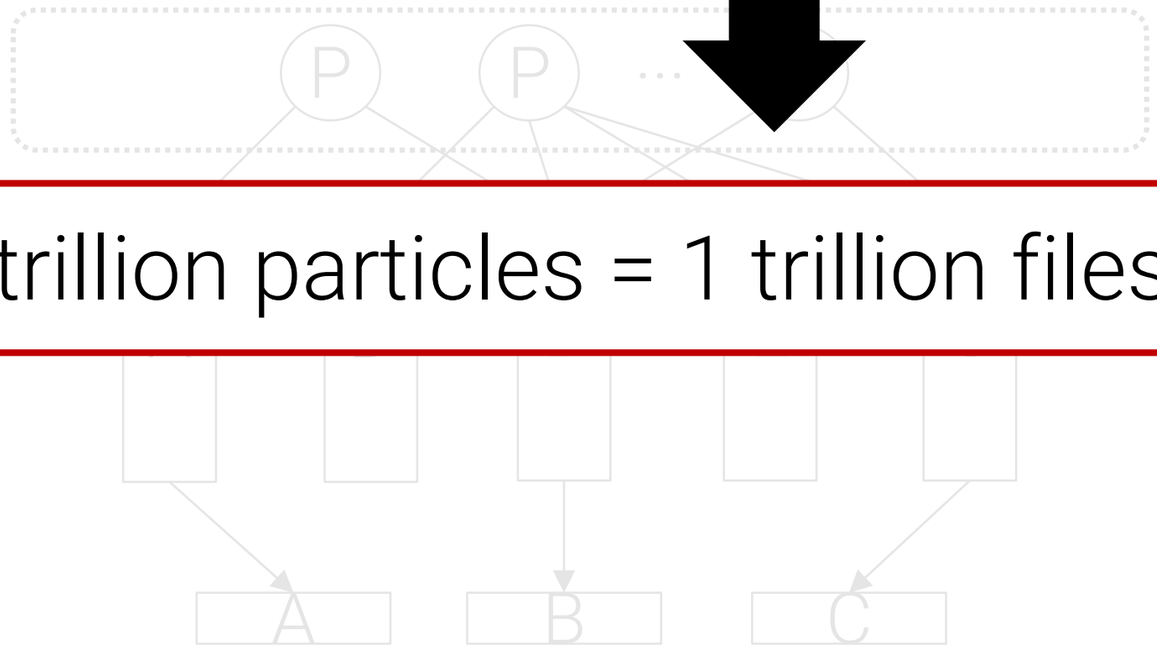
Each query checks ALL files



Let's do

Each query checks 1 file

1 file per object



1 trillion particles = 1 trillion files

Need to Do Things Differently

1 file per proce… work

Let's do 1 file per object

Each query checks 1 file

Broke a world record

1 trillion files perfect for news titles

Need to Do Things Differently

1 file per process doesn't work

Each query checks ALL files

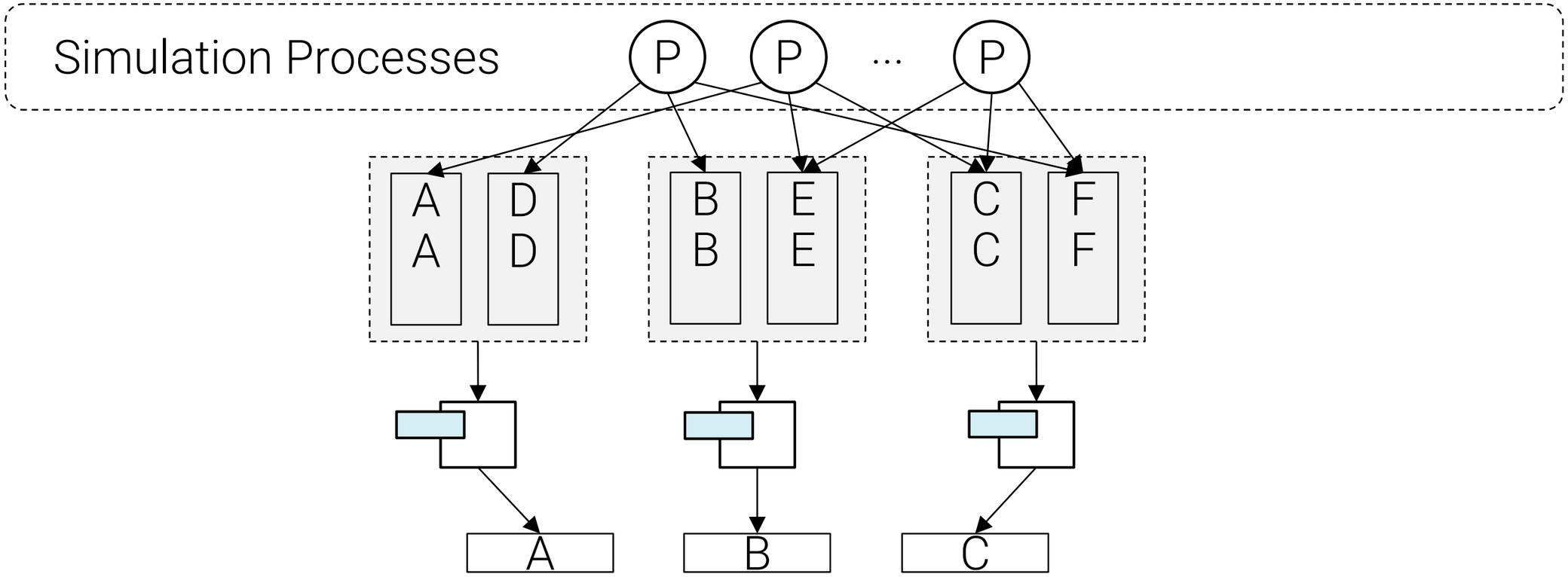
Let's do 1 file per object

Each query checks 1 file

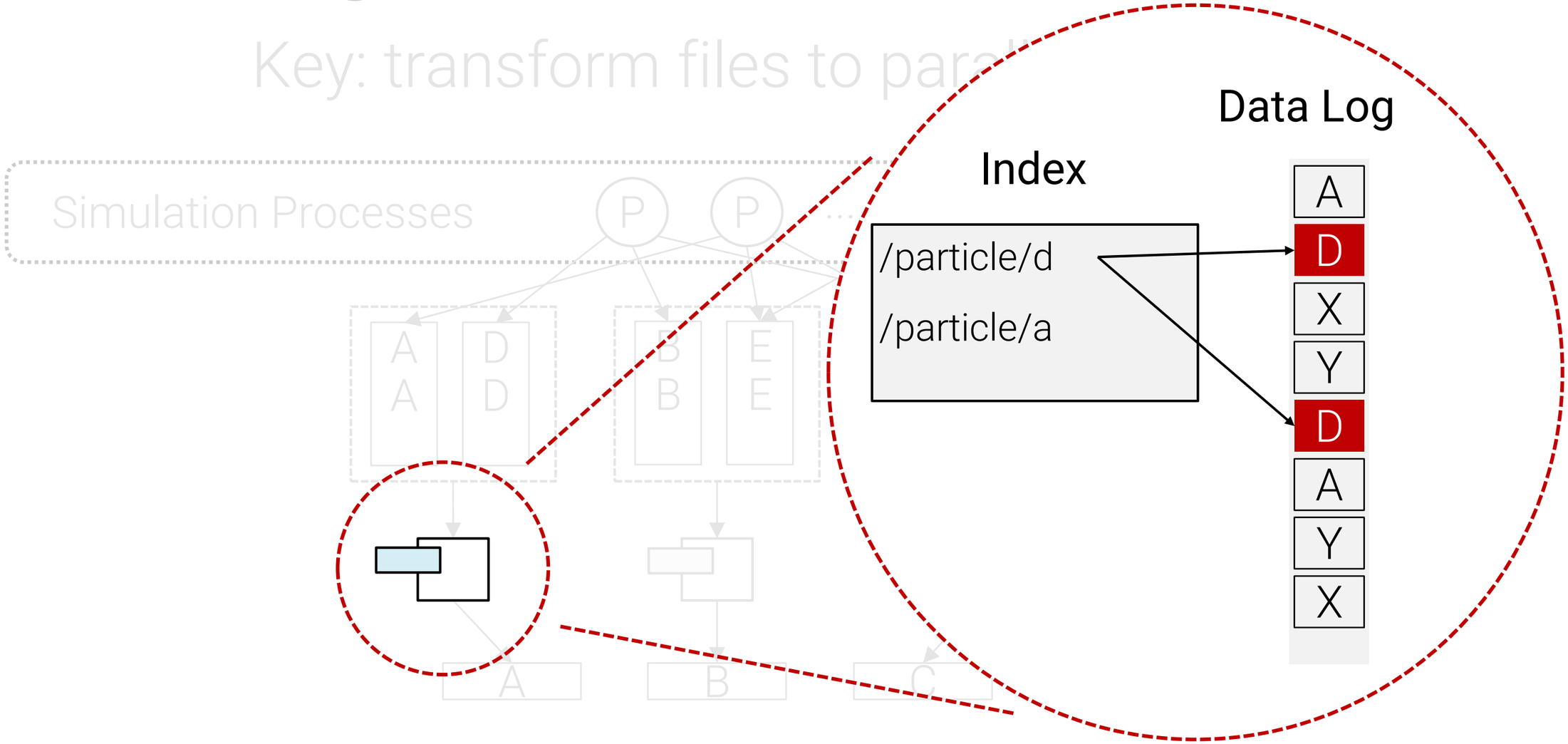
And scientists really love 1 trillion files

Efficiently Work with 1 Trillion Files

Key: transform files to parallel logs



Efficiently Work with 1 Trillion Files



Indexed Massive Directory

Dynamically reorganize files for fast retrieval

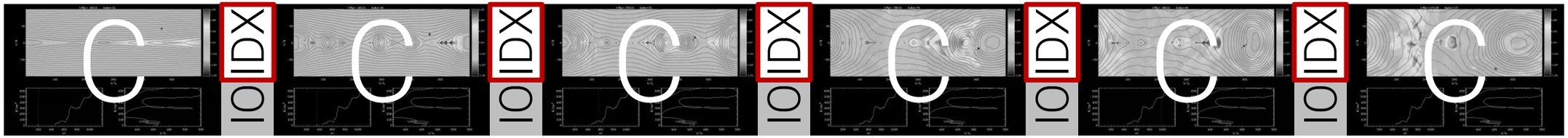
1 trillion files

File reads guaranteed to be efficient

Indexed Massive Directory in English

You Can Hire This Hero for Free

All work done using idle CPU cycles



Results from LANL **Trinitite** Cluster

(96 nodes, 3,072 CPU cores)

5,000x faster at queries

5% longer write time

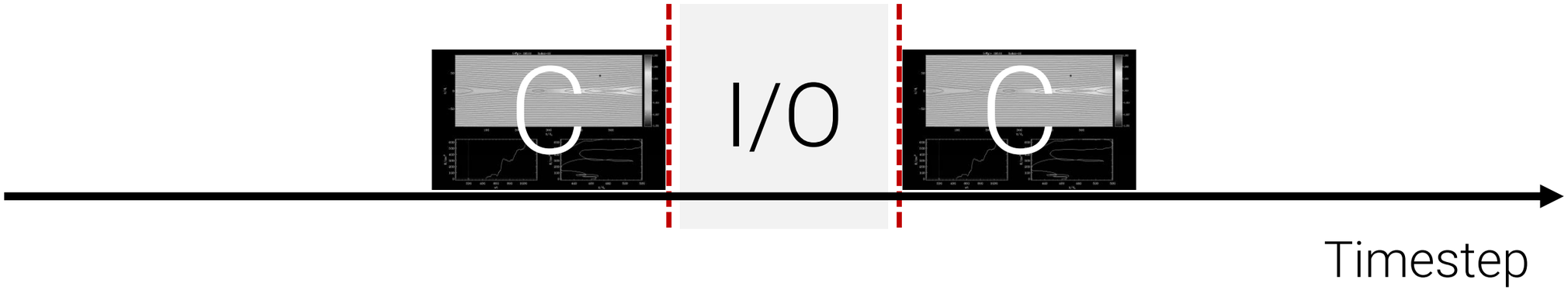
The Rest of The Talk

- **Challenges** for embedded in-situ indexing
- Techniques for **scaling**
- Real-world results

Key Challenges

1. No dedicated cycles

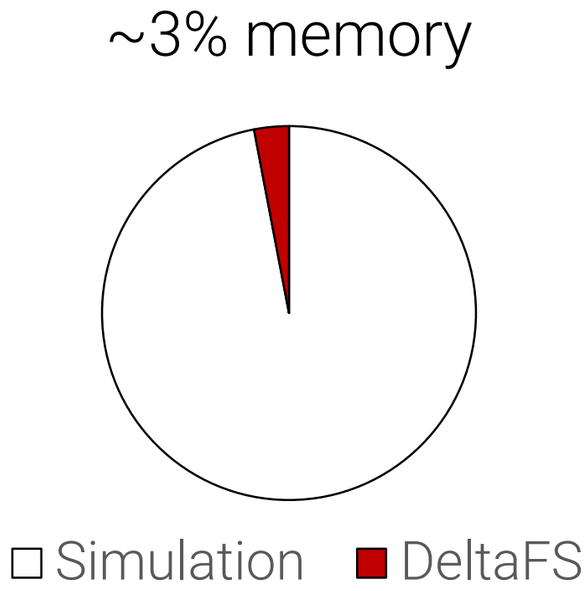
No work during simulation computation



Key Challenges

1. No dedicated cycles

→ 2. Intensive memory pressure

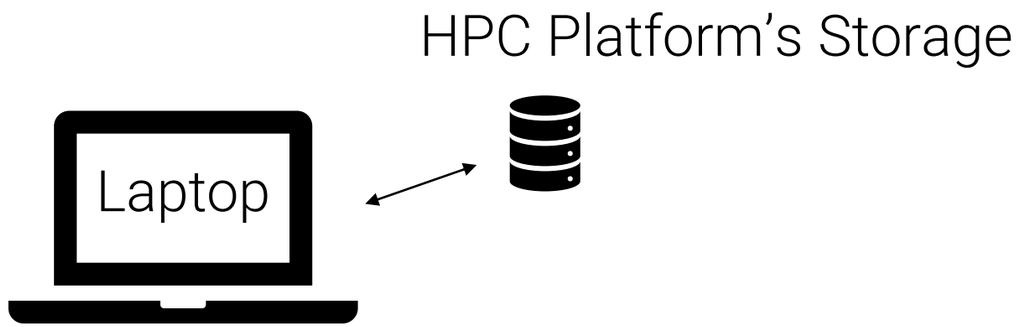


Key Challenges

1. No dedicated cycles

2. Intensive memory pressure

→ 3. Resource-skinny queries



No need to use a supercomputer

Requires New Techniques

Self-balancing Data Structures (e.g. LSM)

Fast queries
High write overhead

No Dynamic Indexing

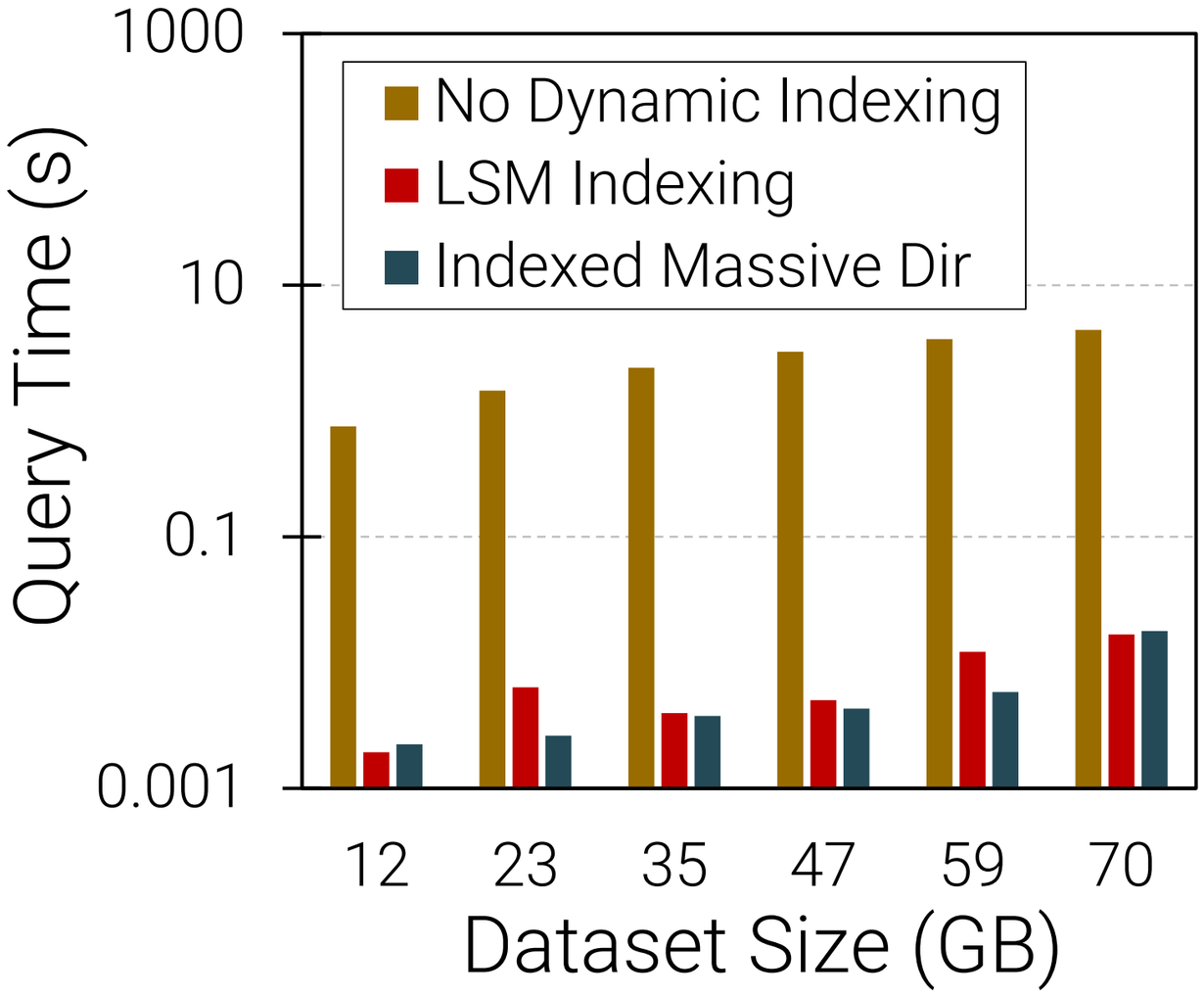
Slow queries
No extra write overhead

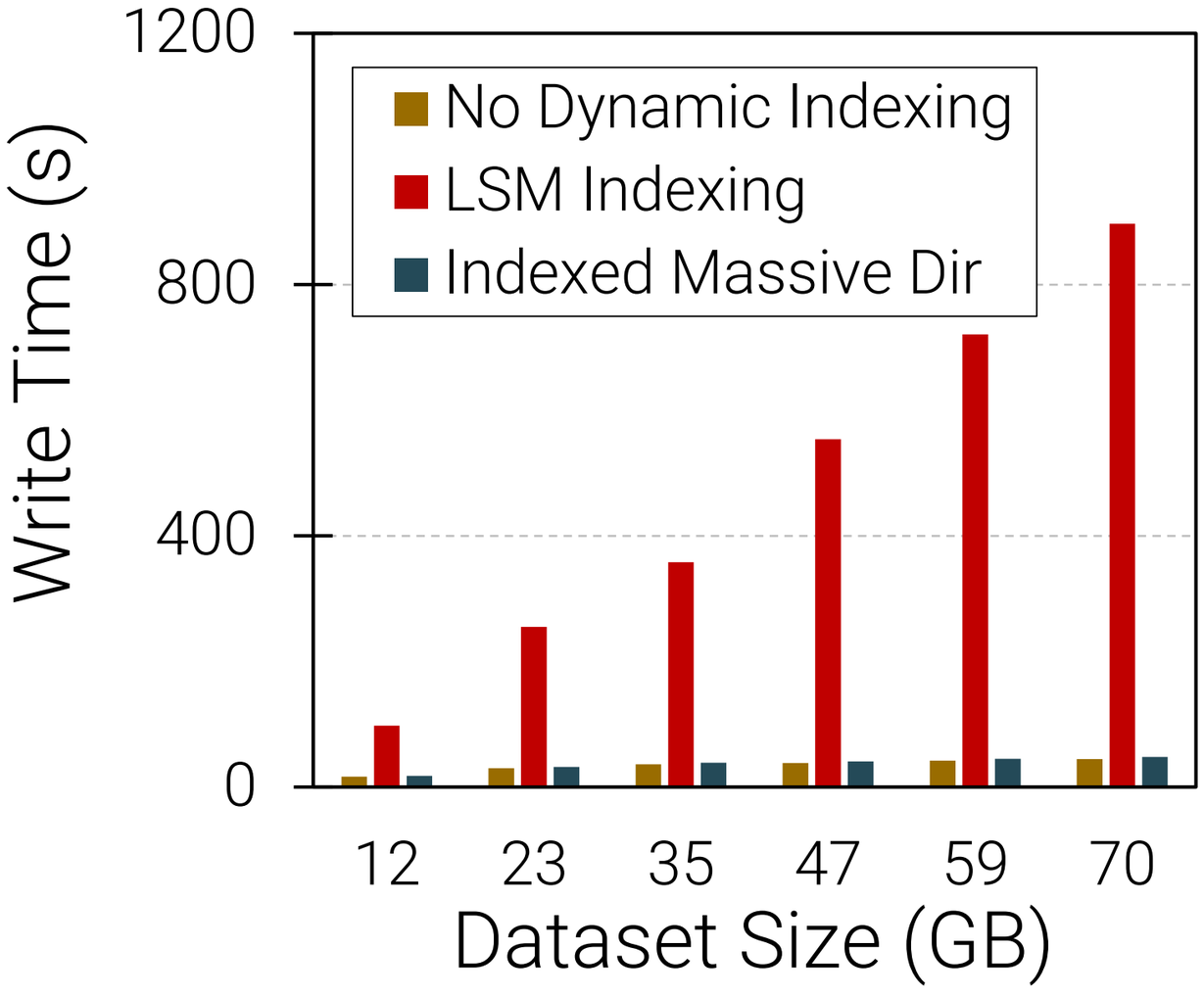
Indexed Massive Directory

Fast queries
Low write overhead

MORE INFO IN PAPER

Result: Faster Query, Low Write Overhead



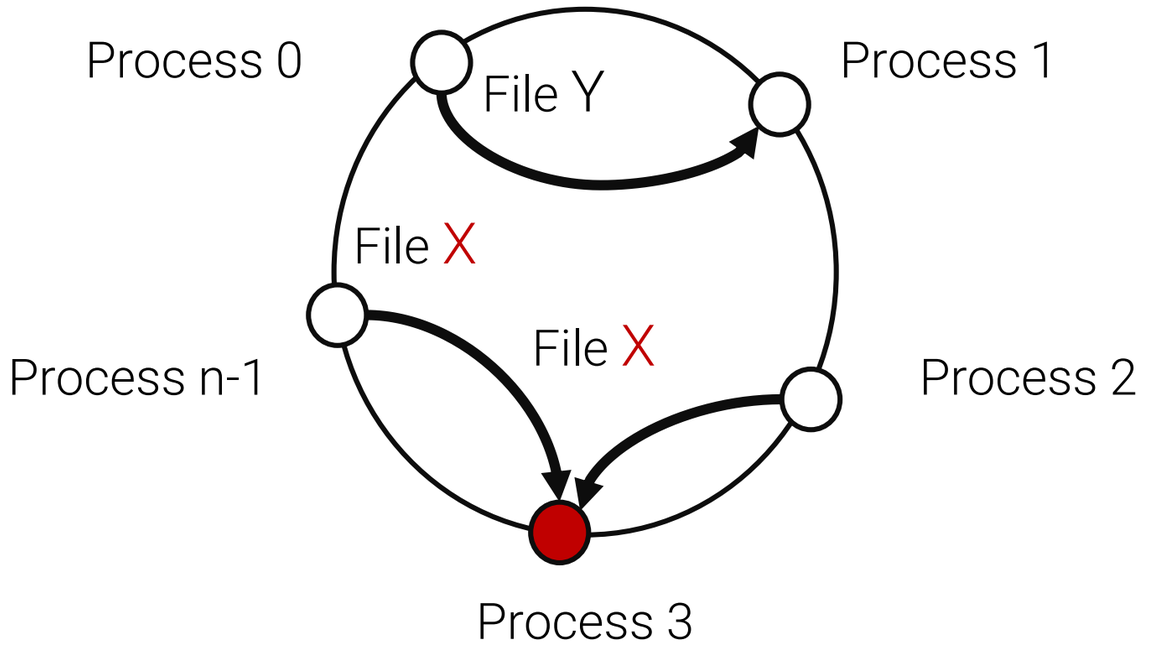


Make it Scale

This talk focuses on scalable inter-process communication

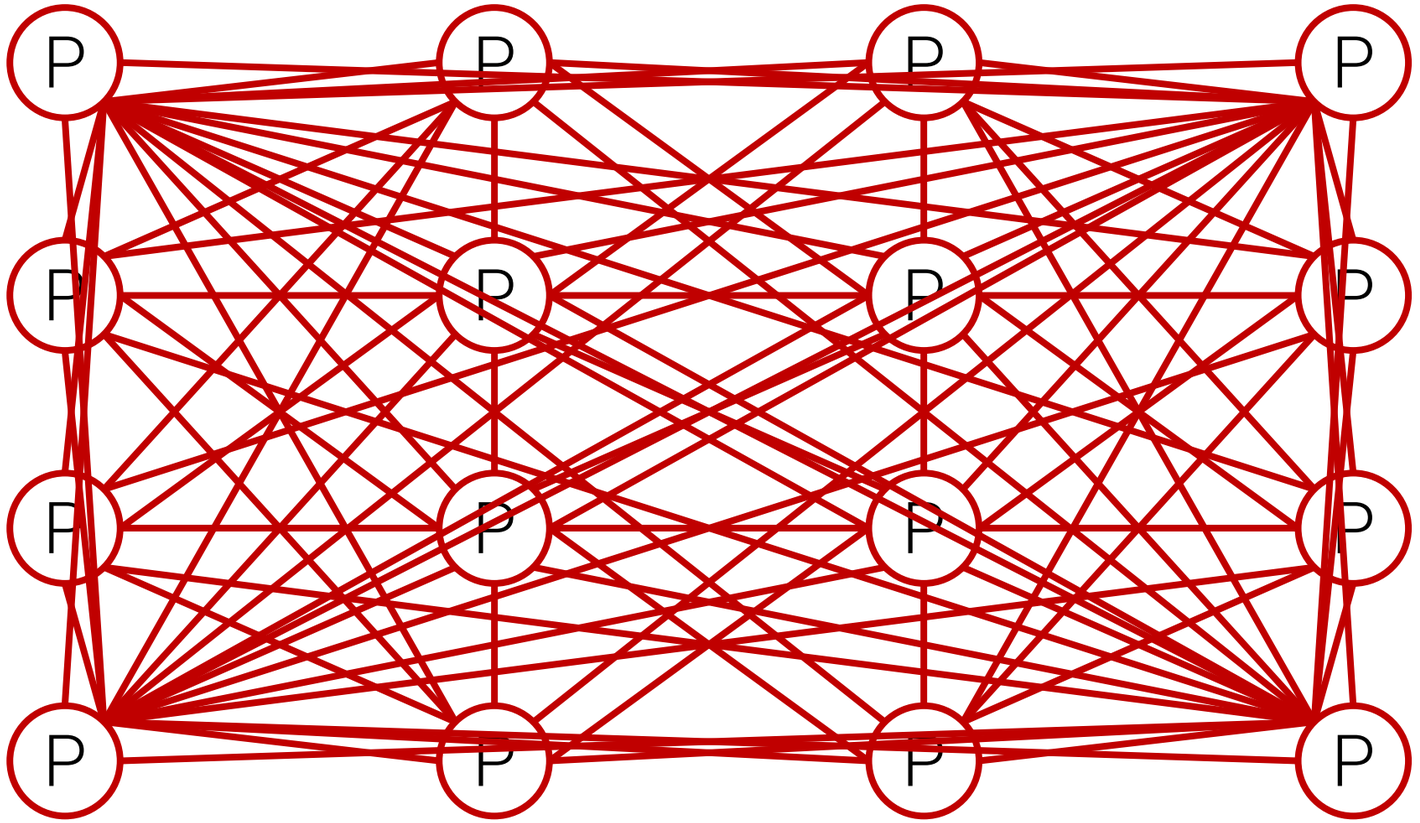
MORE INFO IN PAPER

Recall: We Partition Data Dynamically



Each query hits 1 partition

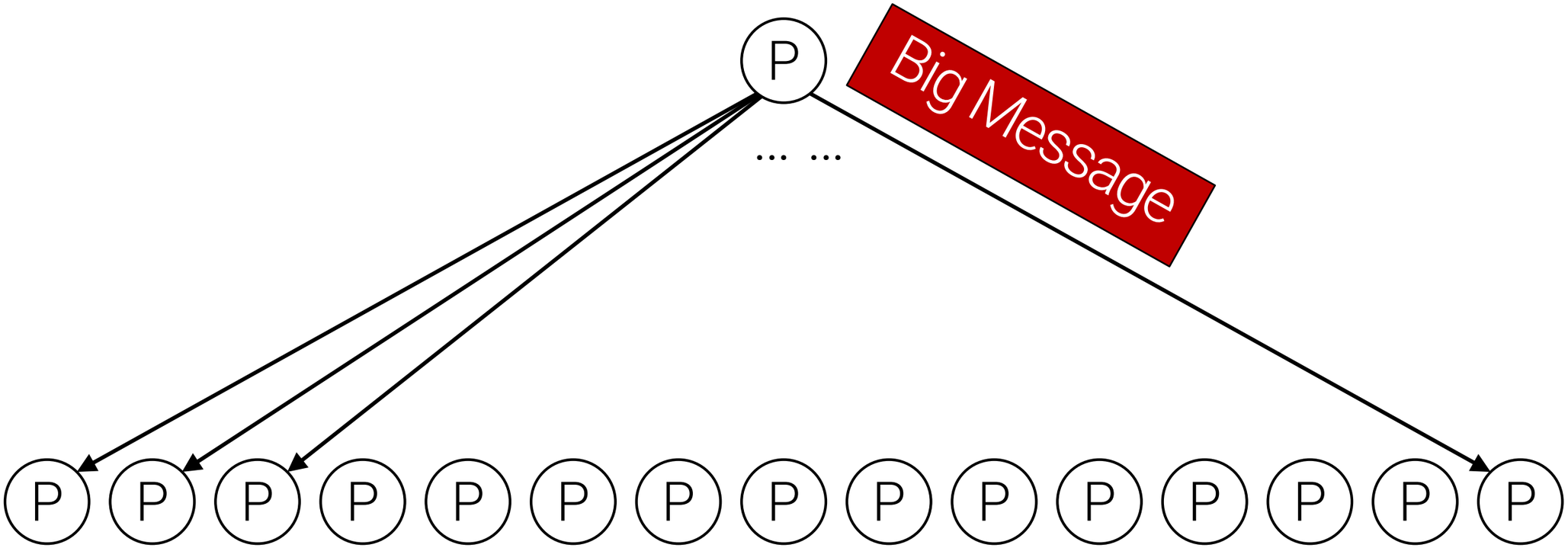
Doesn't Work at Scale



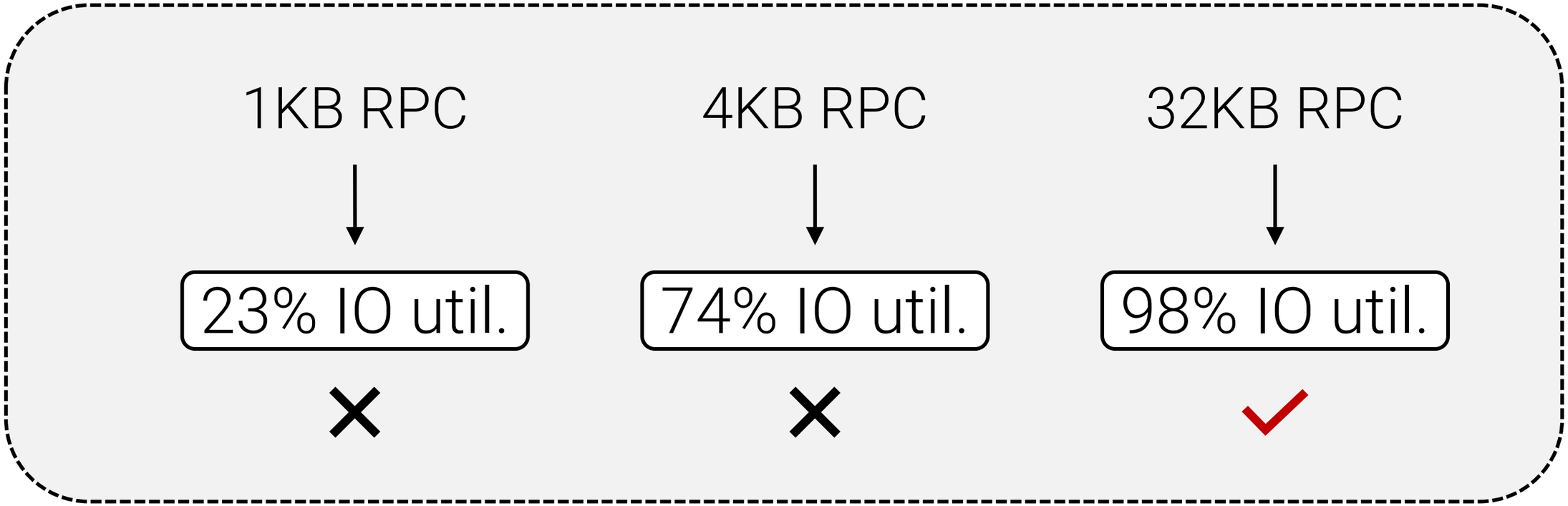
← A process running on 1 CPU core

Note: not all links are shown

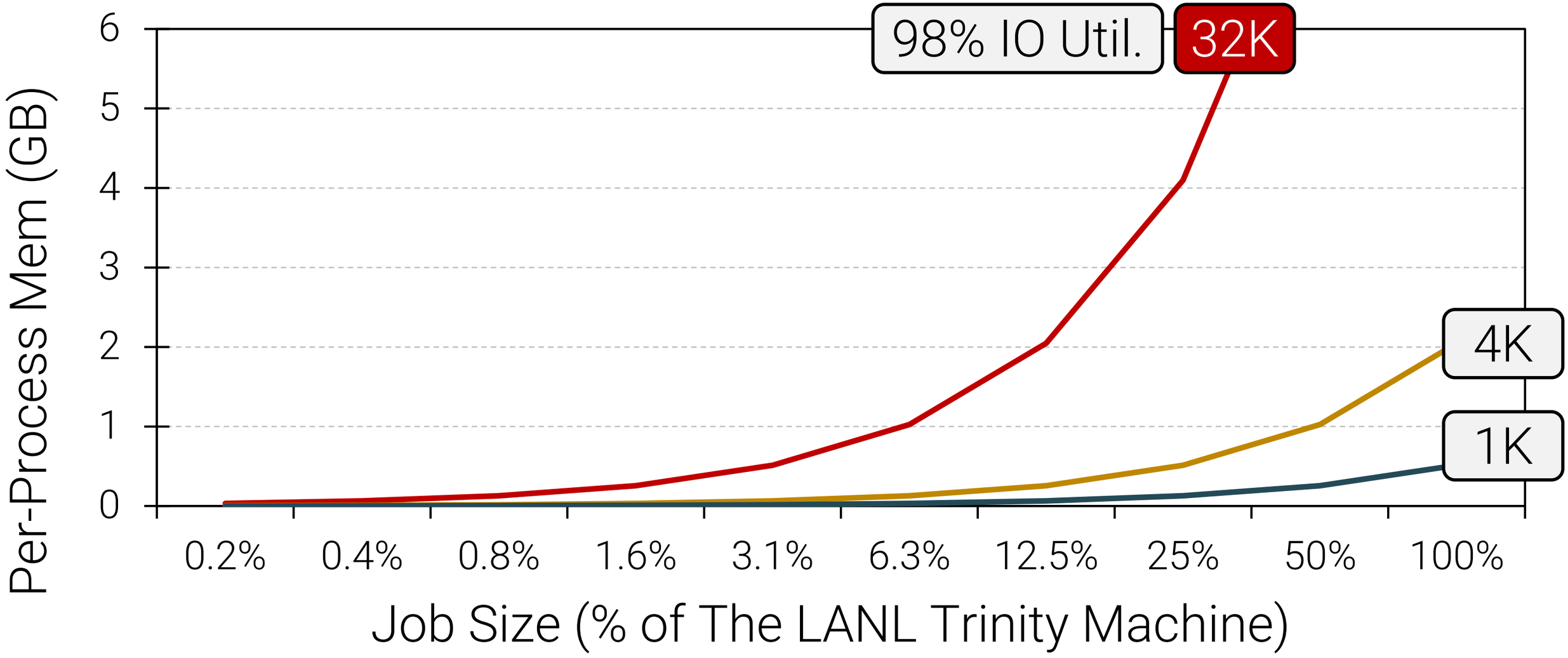
Data is Buffered for Each Destination



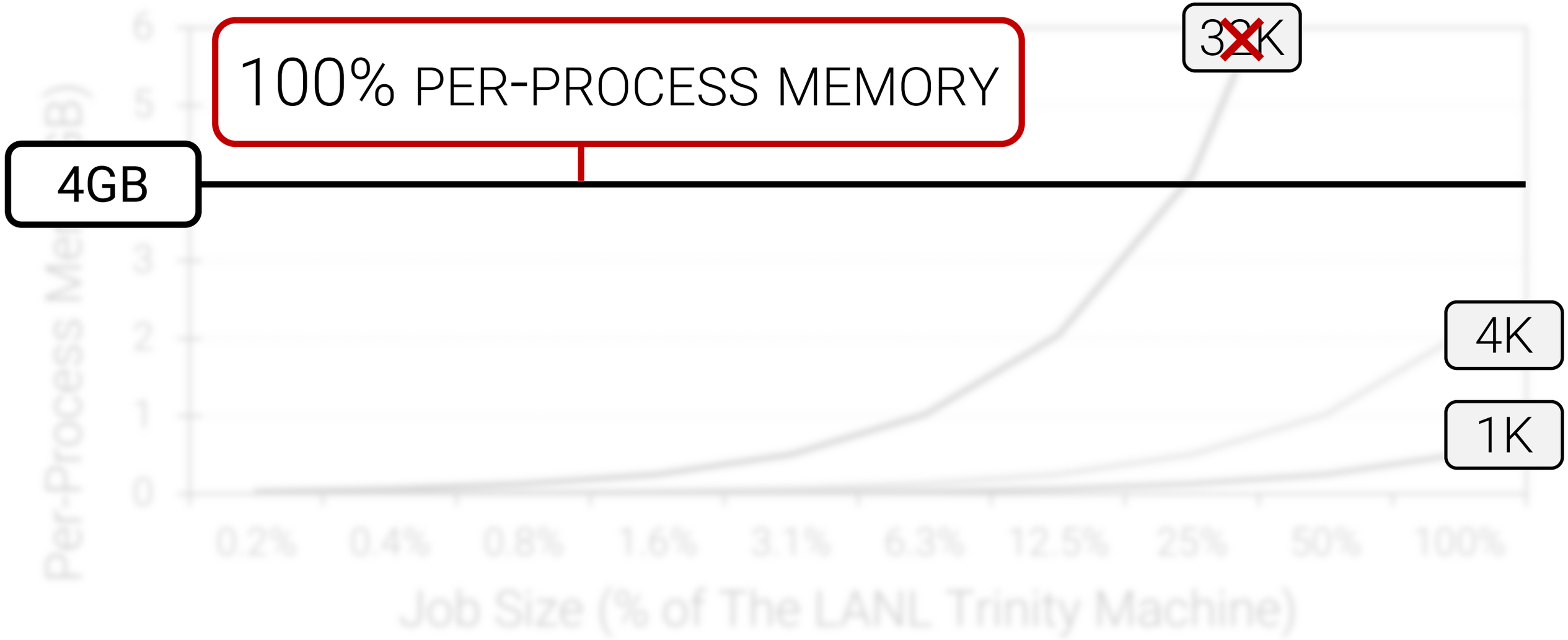
Must Use Large Buffers



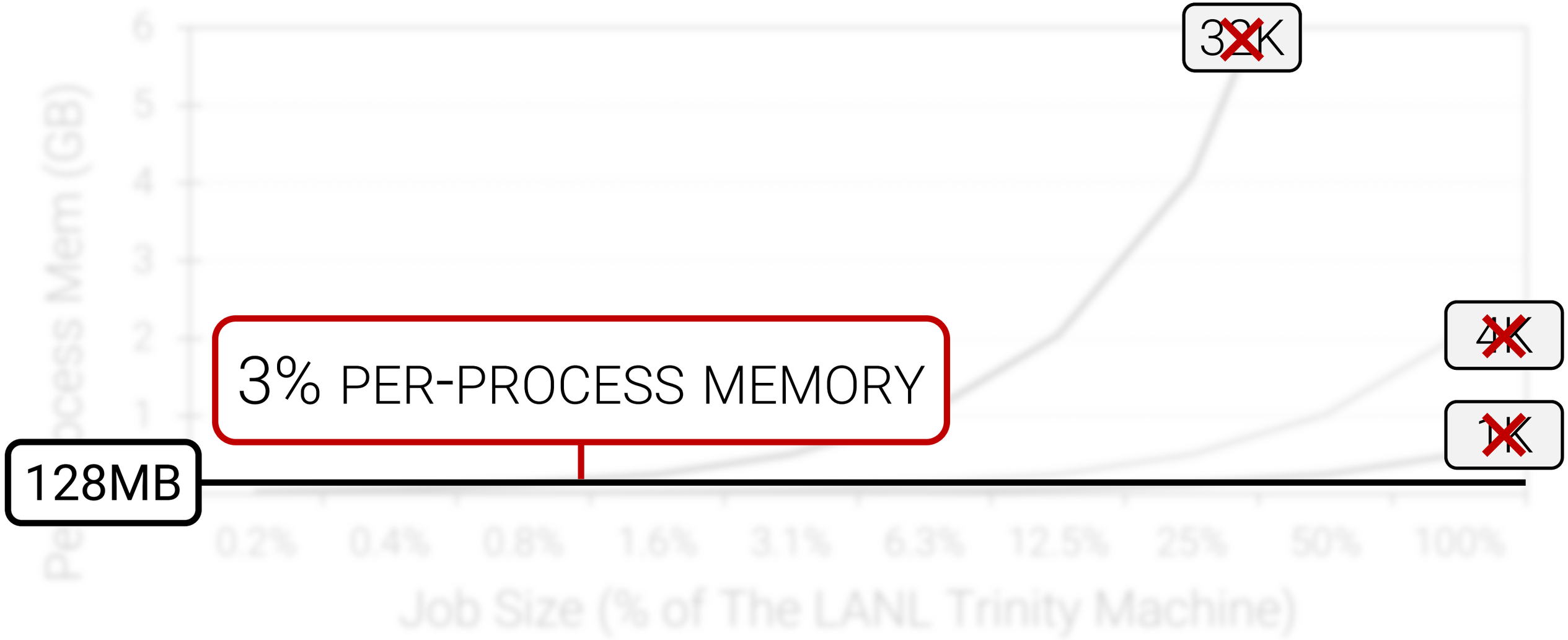
Can’t Afford The Memory



Can’t Afford The Memory

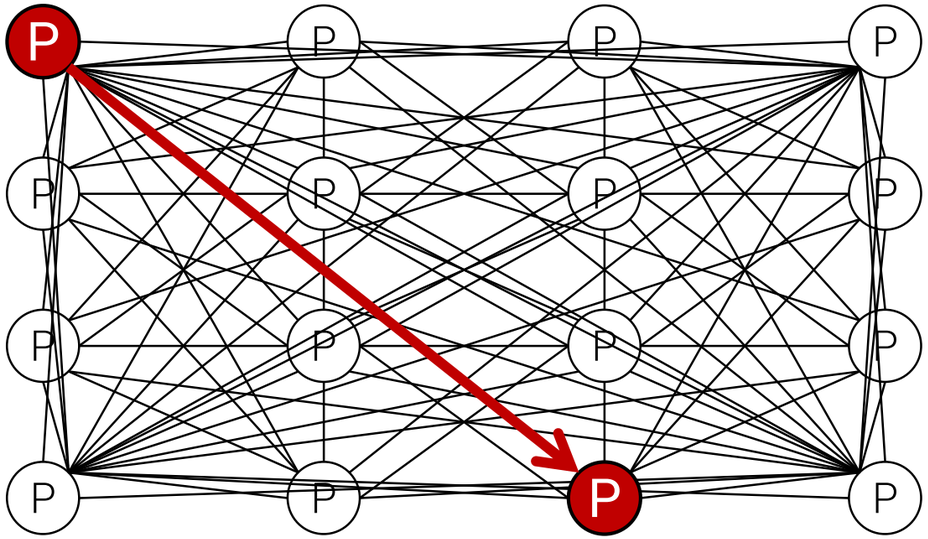


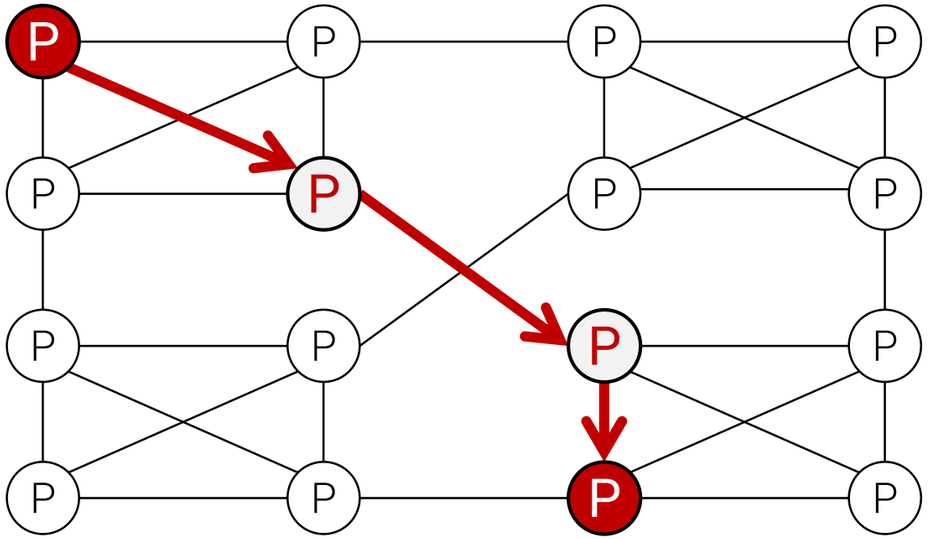
Can’t Afford The Memory



Fan-Out Control

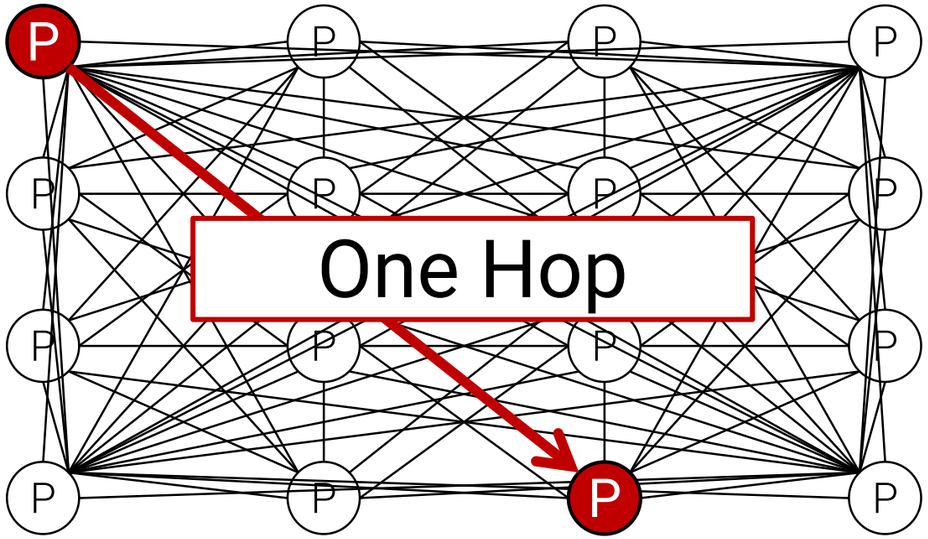
Solution: add 2 extra hops





Fan-Out Control

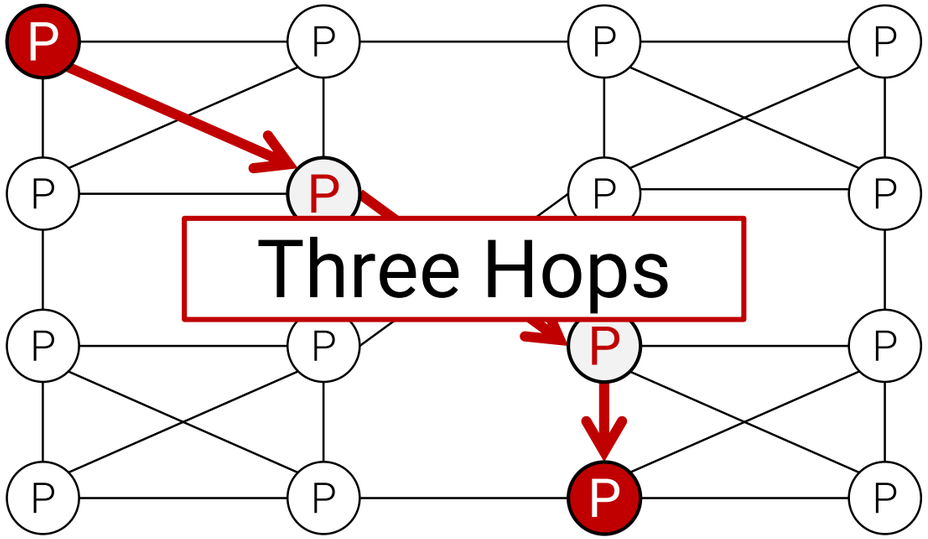
Solution: add 2 extra hops



CASE STUDY

131,072 procs: 4GB mem per-proc

(100% per-proc memory)

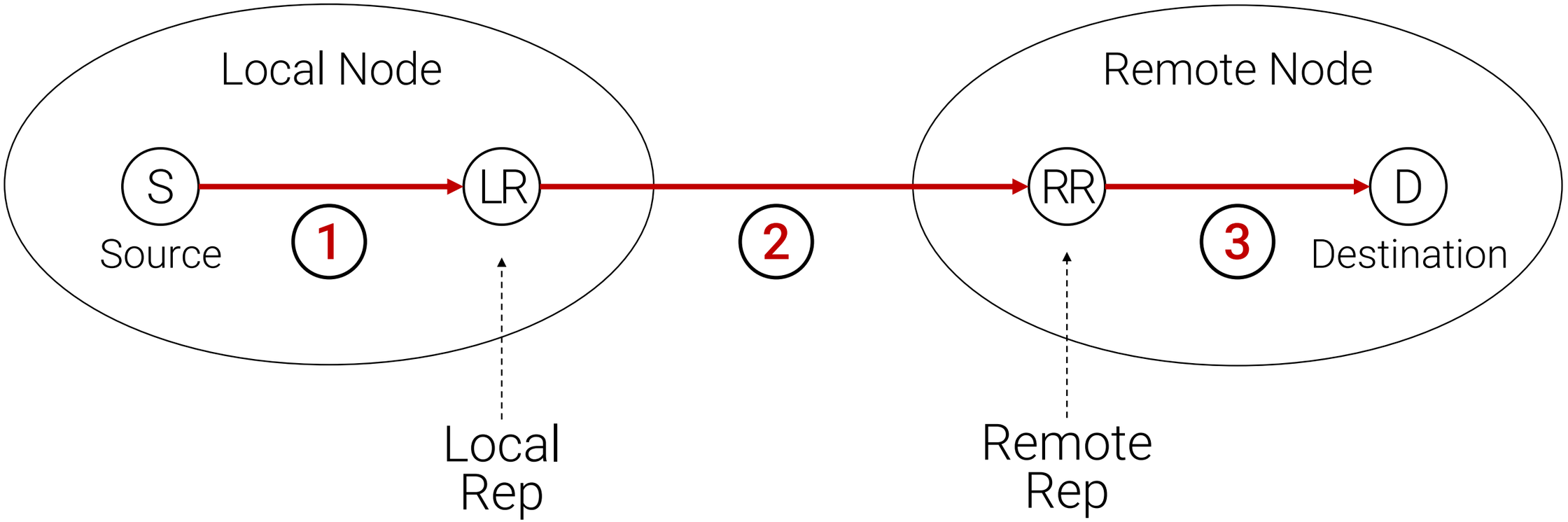


CASE STUDY

131,072 procs: 6MB mem per-proc

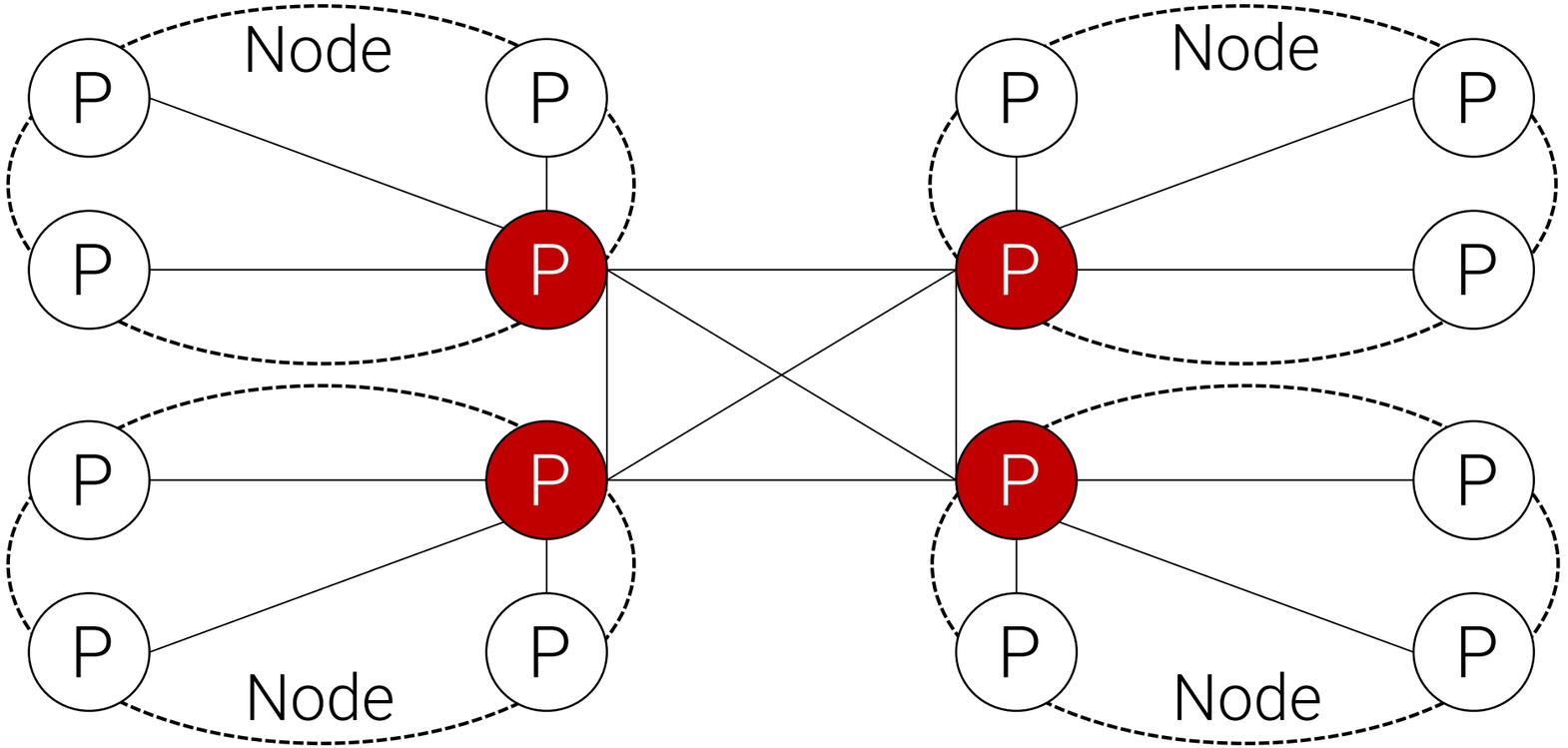
(0.15% per-proc memory)

Three-Hop Explained



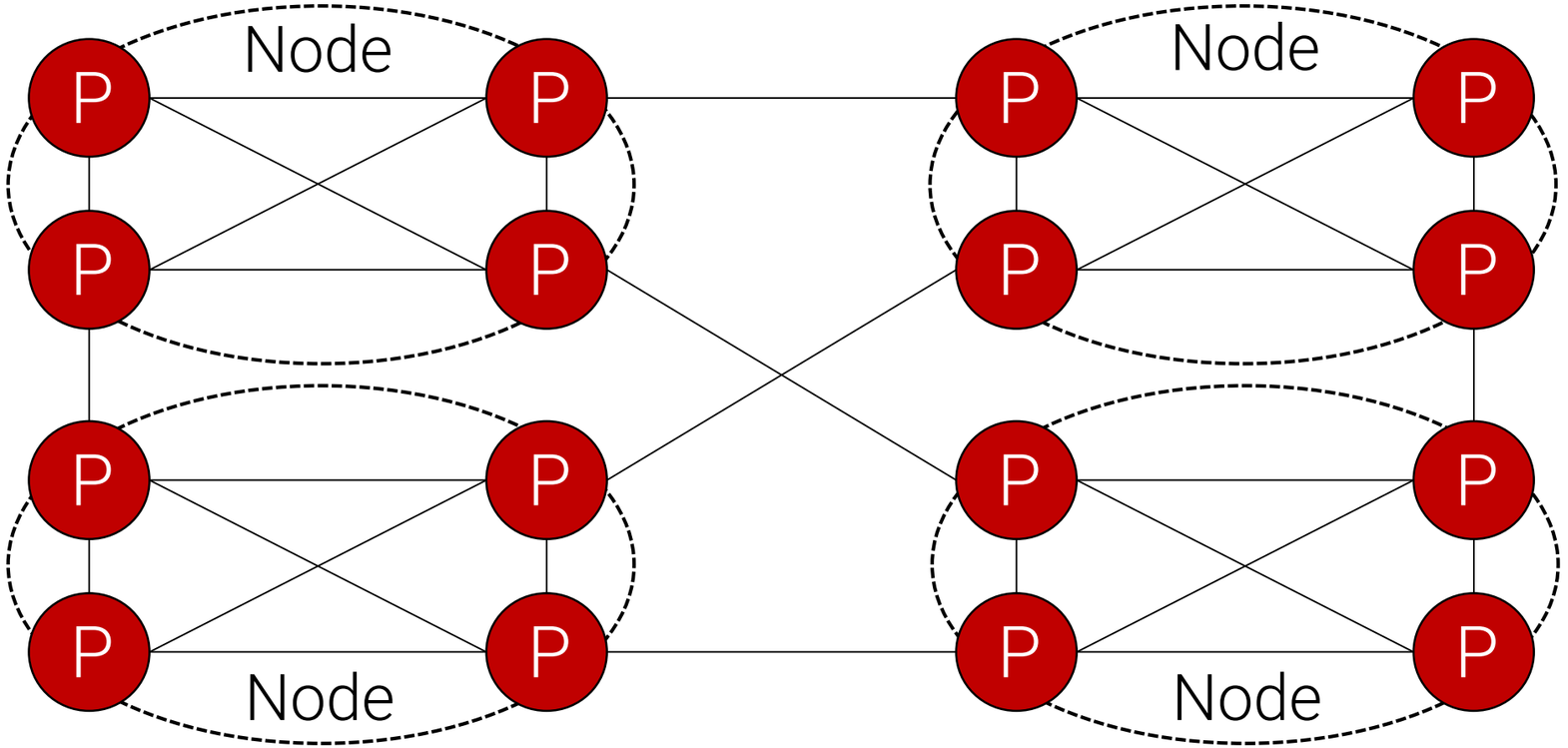
Three-Hop In Action

Transform core-to-core communication to node-to-node



Load Balance

Each process manages a subset of remote nodes



Three-Hop Takeaway

For **N** total processes, and **k** processes per node

One Hop

O(N)

links per process

Three Hops

O(N/k^2)

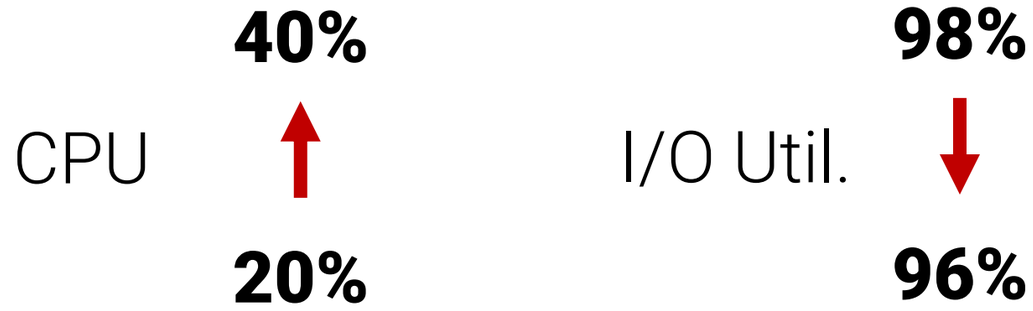
links per process

MORE INFO IN PAPER

Cost of Extra Hops

Negligible because storage is the dominant bottleneck

Results from LANL clusters



MORE INFO IN PAPER

More Techniques in our Paper

MORE INFO IN PAPER

One more thing: DeltaFS is built w/ composable services

Enabling Data Services for HPC

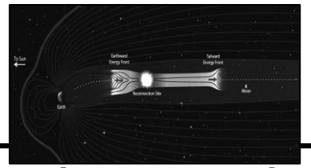
Jerome Soumagne

Philip Carns, Kevin Huck, Johann Lombardi, Manish Parashar

Tue / 5:15pm / C141

The Trinity Experiment

Experimental Settings



Particle Simulation

Compute Nodes

SSD

Burst-Buffer

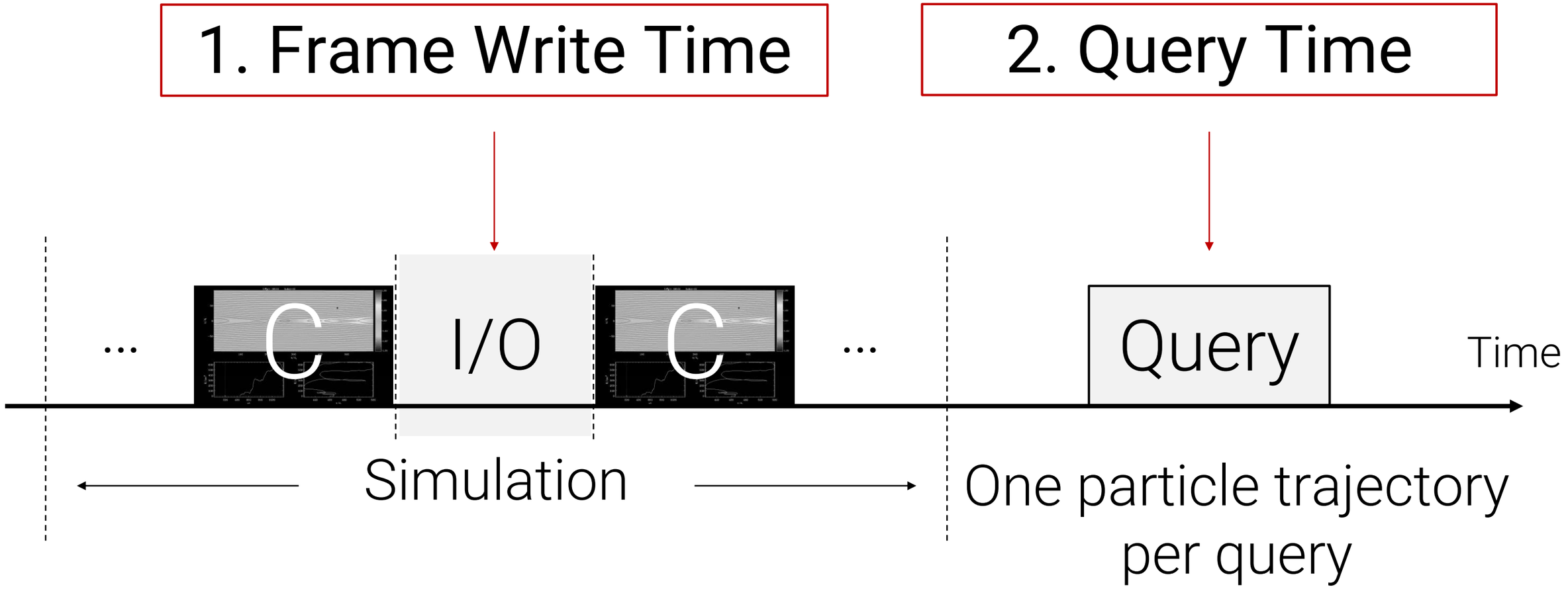
HDD

Storage

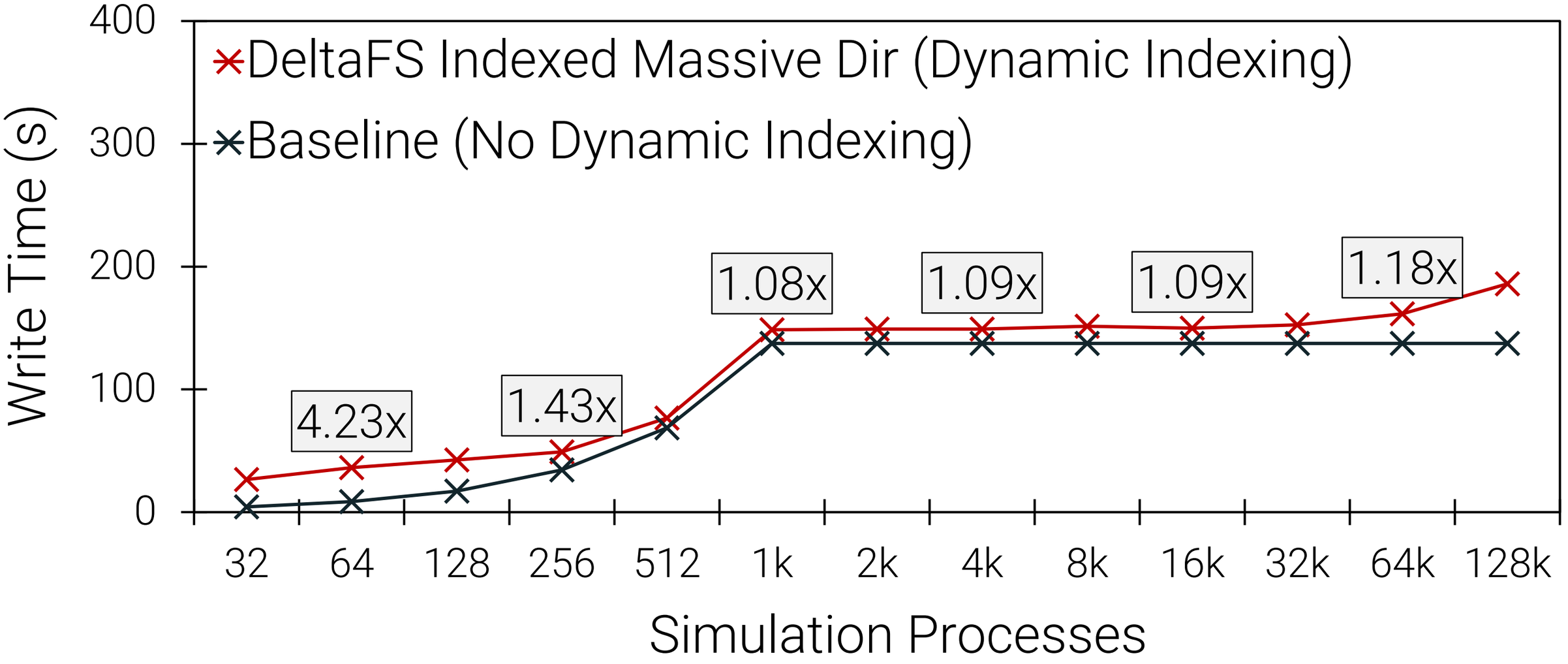
Queries

Up to 4096 compute nodes, 131,072 cores, 2 trillion particles

Measurements

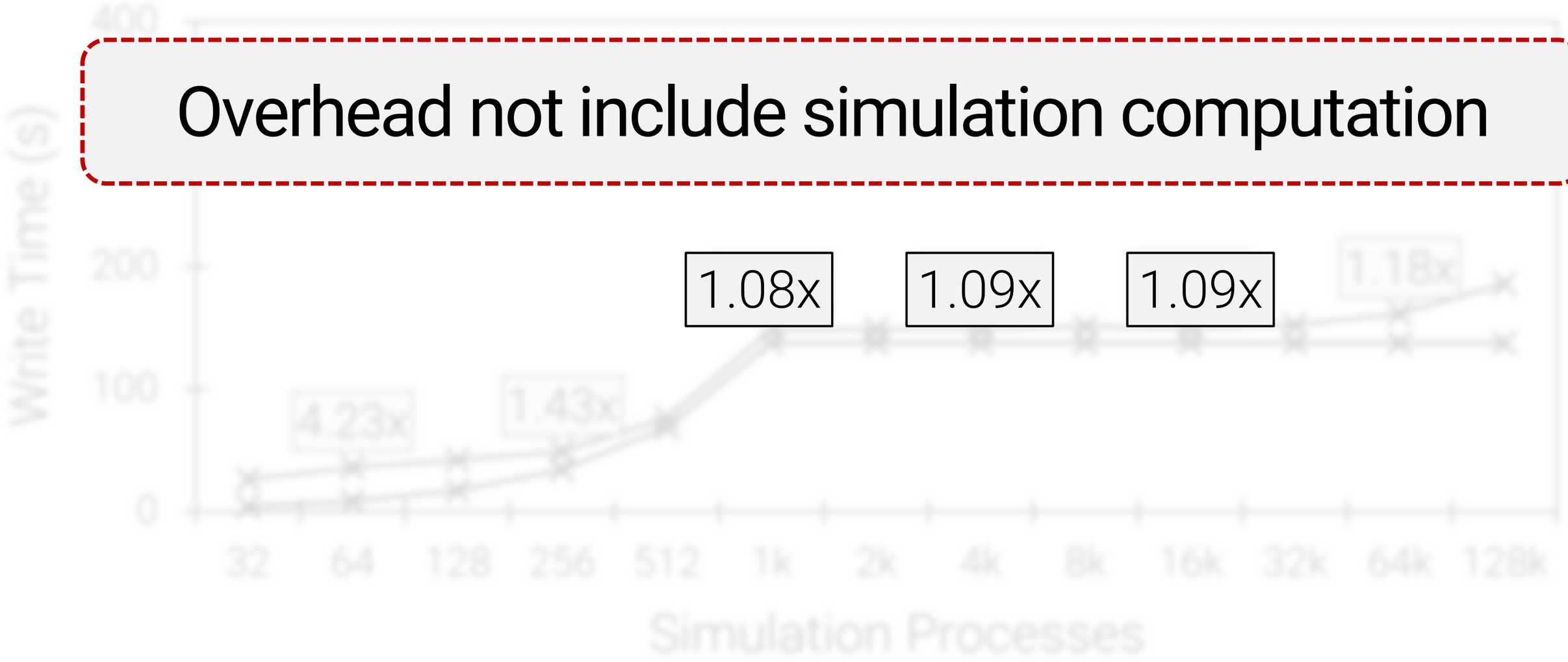


Minimal Write-Time Overhead

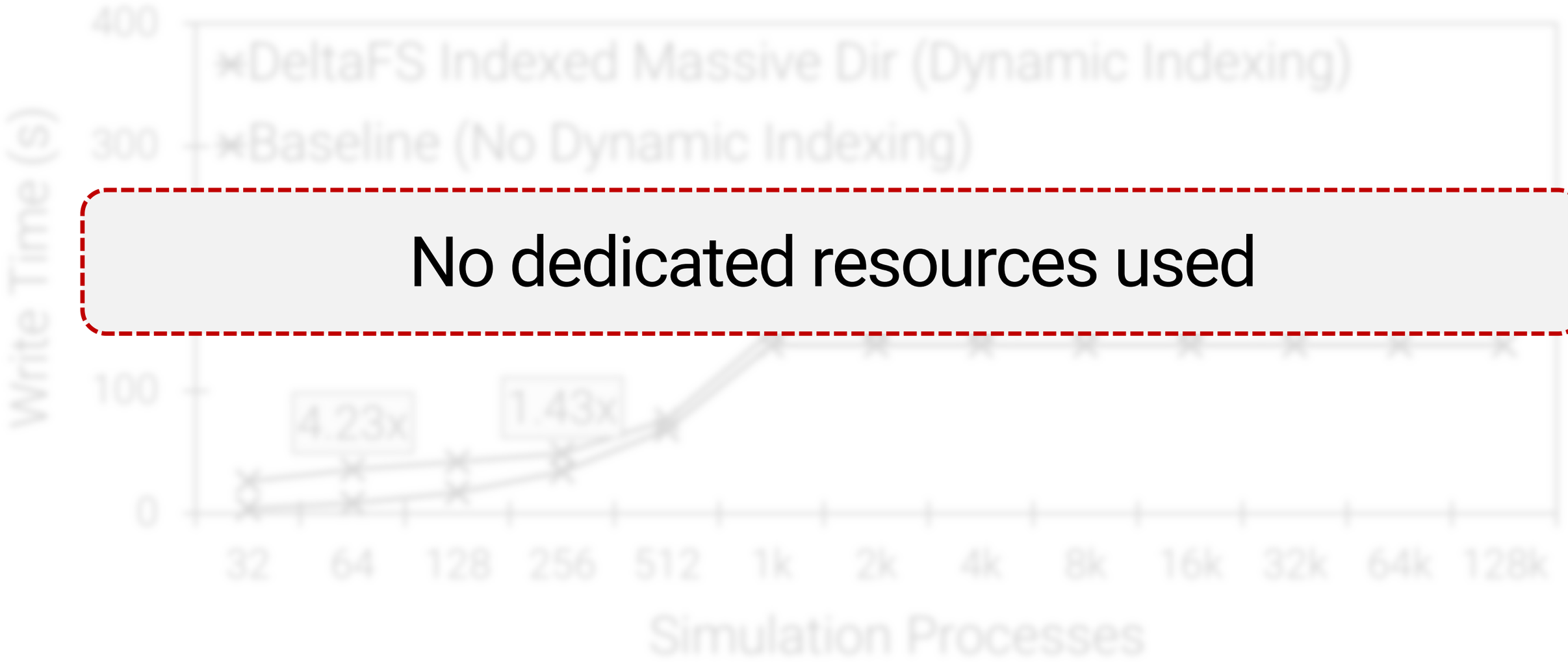


Minimal Write-Time Overhead

Overhead not include simulation computation

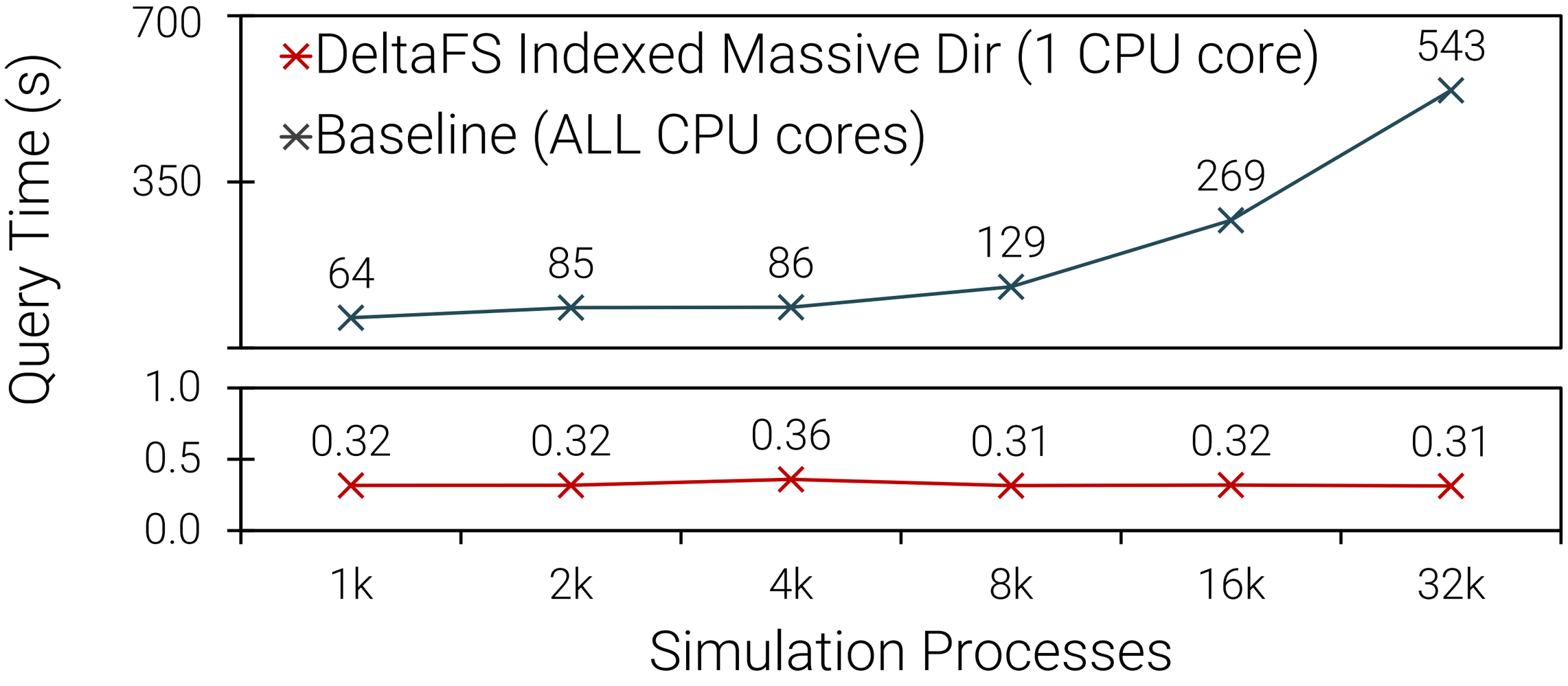


Minimal Write-Time Overhead

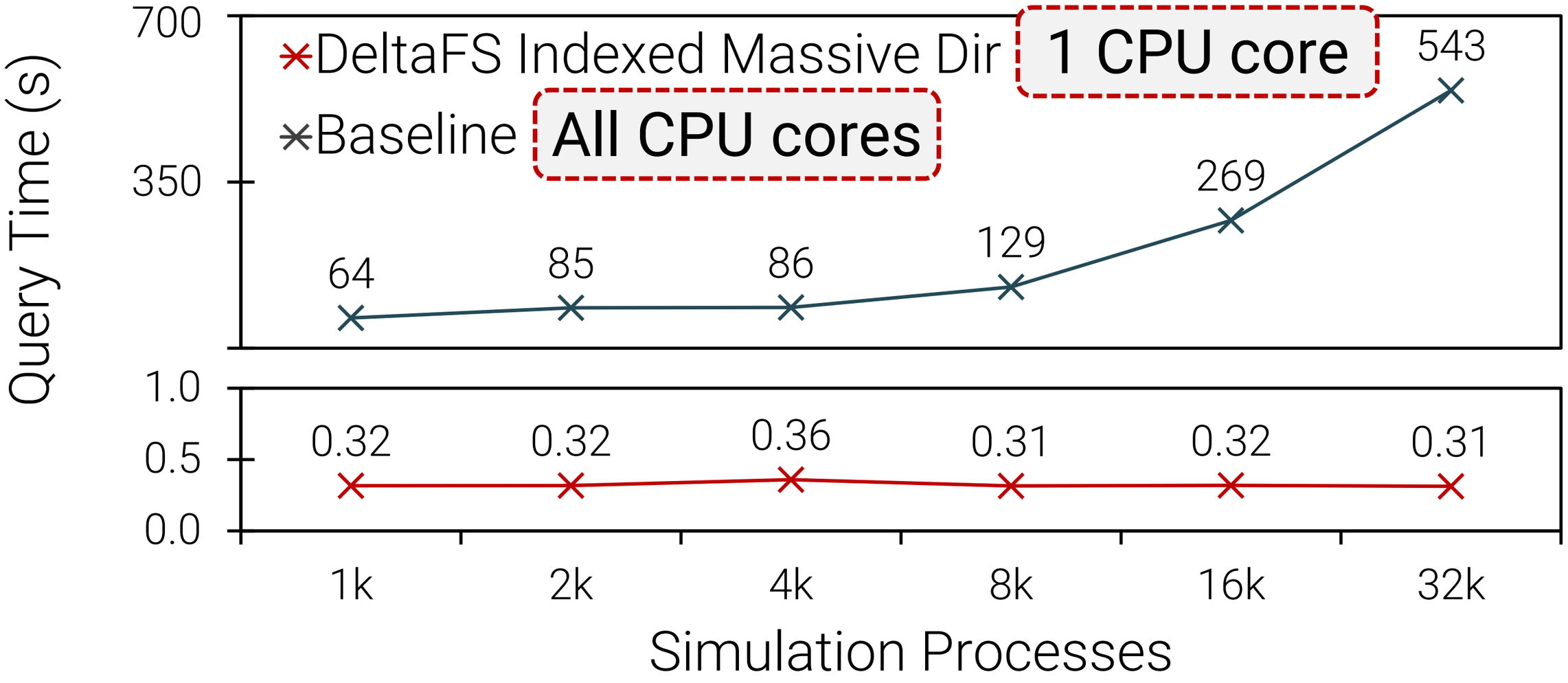


No dedicated resources used

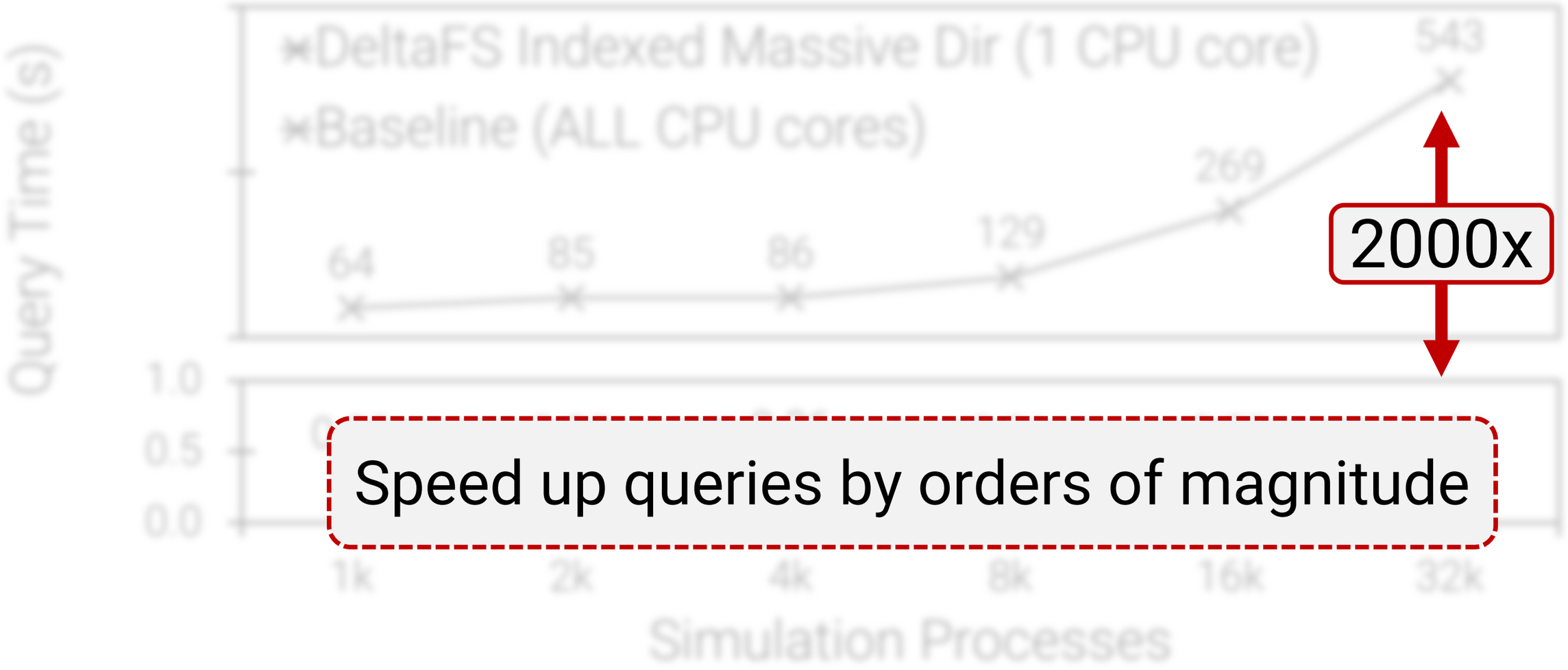
Faster Queries



Faster Queries



Faster Queries



Speed up queries by orders of magnitude

Summary

Processing data in-situ drastically improves time-to-insight

You can do it using only idle CPU cycles

MORE INFO IN PAPER

qingzhen+sc18@andrew.cmu.edu