Automatic Identification of Application I/O Signatures from Noisy Server-Side Traces

Yang Liu Raghul Gunasekaran Xiaosong Ma Sudharshan S. Vazhkudai




NORTH CAROLINA STATE UNIVERSITY
NC STATE UNIVERSITY

OAK RIDGE National Laboratory

معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
Member of Qatar Foundation

OAK RIDGE National Laboratory

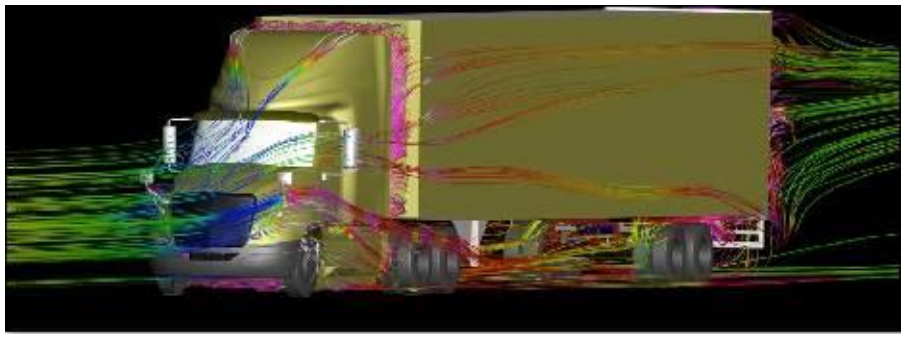
Instance of Large-Scale HPC Systems

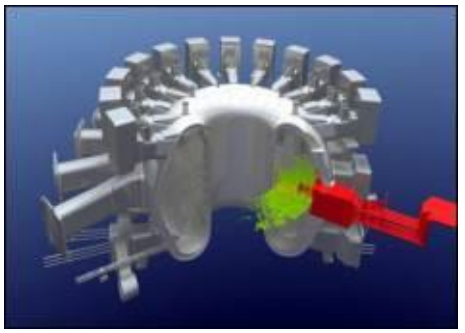
ORNL's TITAN
(World's #2 Supercomputer)

- 27.1 PF Peak performance
- 18,688 compute nodes
  - 16-core AMD Opteron
  - Nvidia Tesla GPU
  - 32 + 6 GB memory
- 3-D Torus interconnect

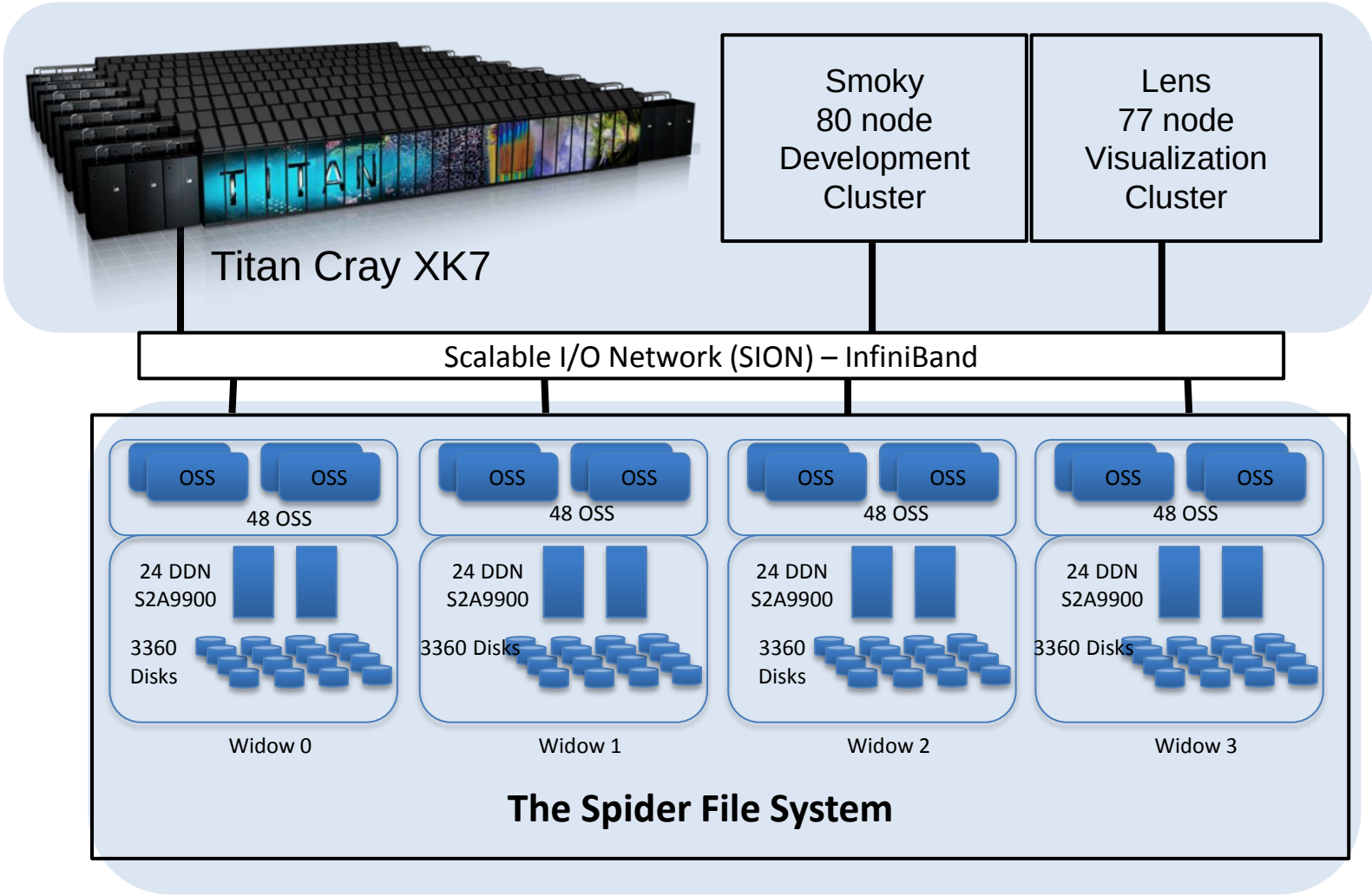
Delivering science and solving real-world problems

- 400+ scientific users
- Diverse science domains

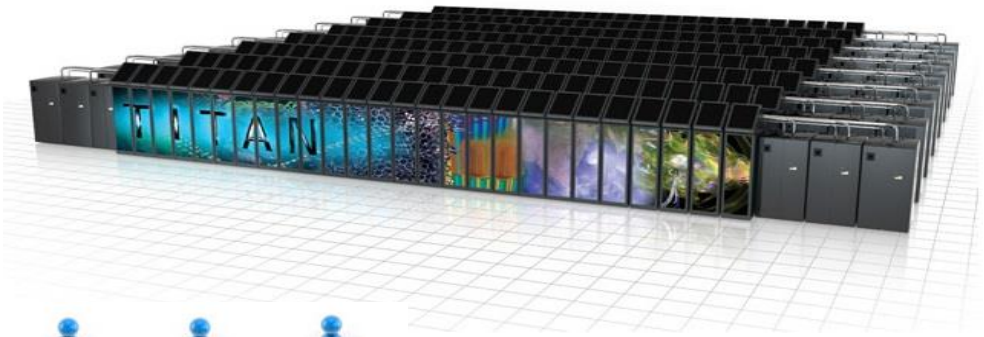




OLCF Architecture Overview

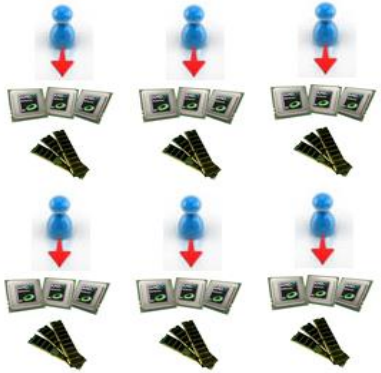


I/O Bottleneck in HPC



Processor

Memory



Interconnect network

Storage system

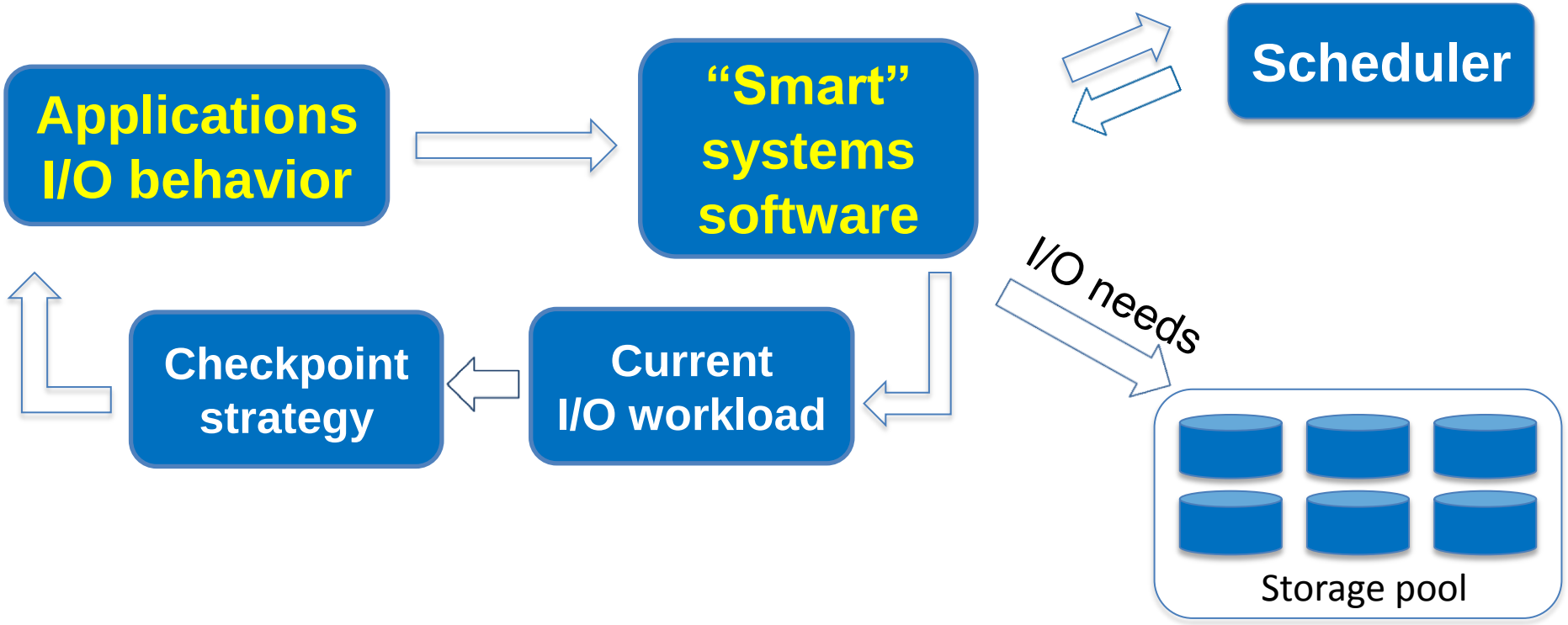
Resource contention

Longer and highly variable runtime

Lower scientific productivity

Reduced resource utilization

I/O Usage Patterns Needed to Alleviate I/O Bottleneck



We need to know application's I/O patterns and trends

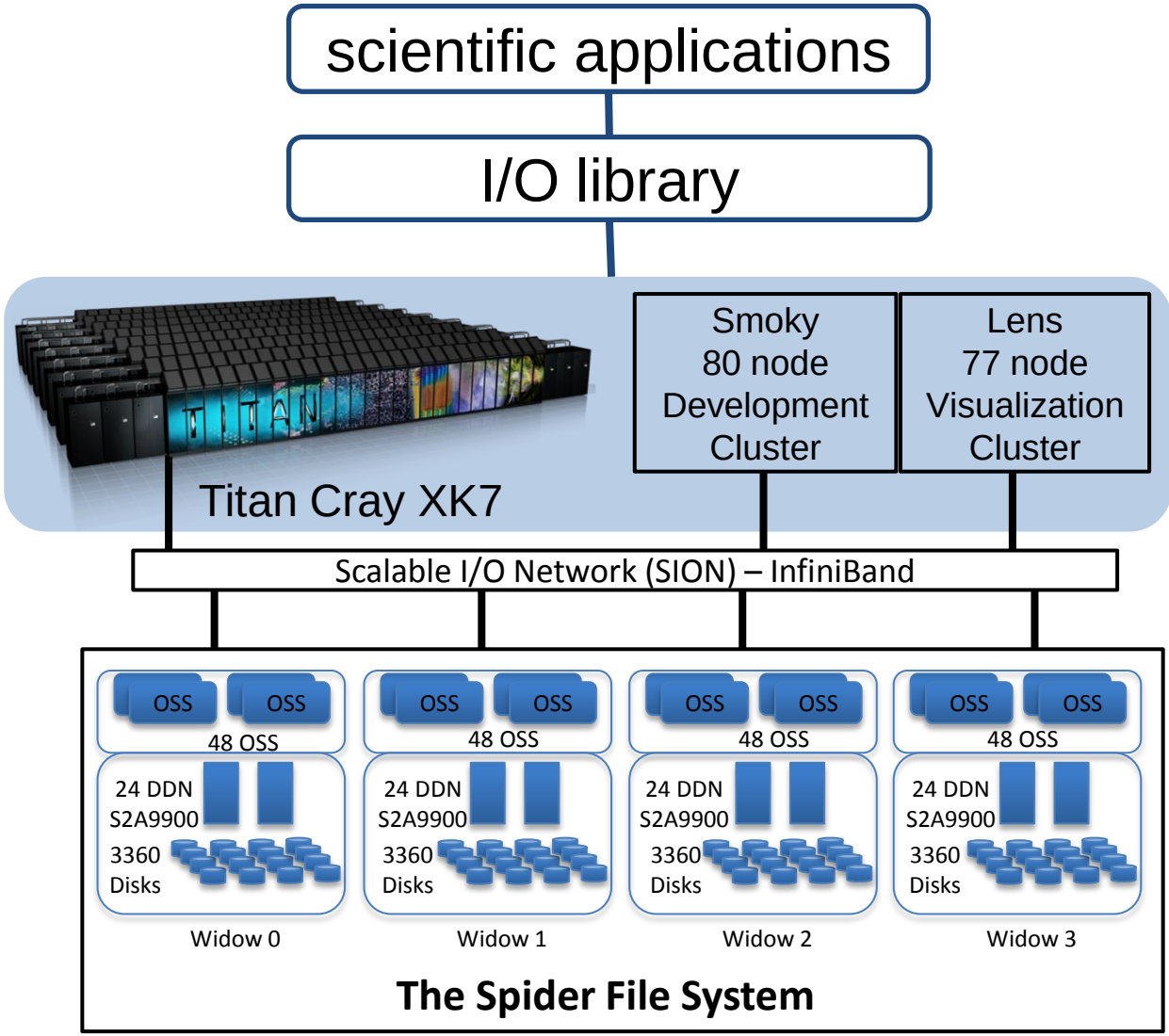
Outline

- Background
- Motivation
- Tracing approach comparison
- Problem definition
- Design
- Evaluation
- Conclusion

Client-side Tracing

Problems

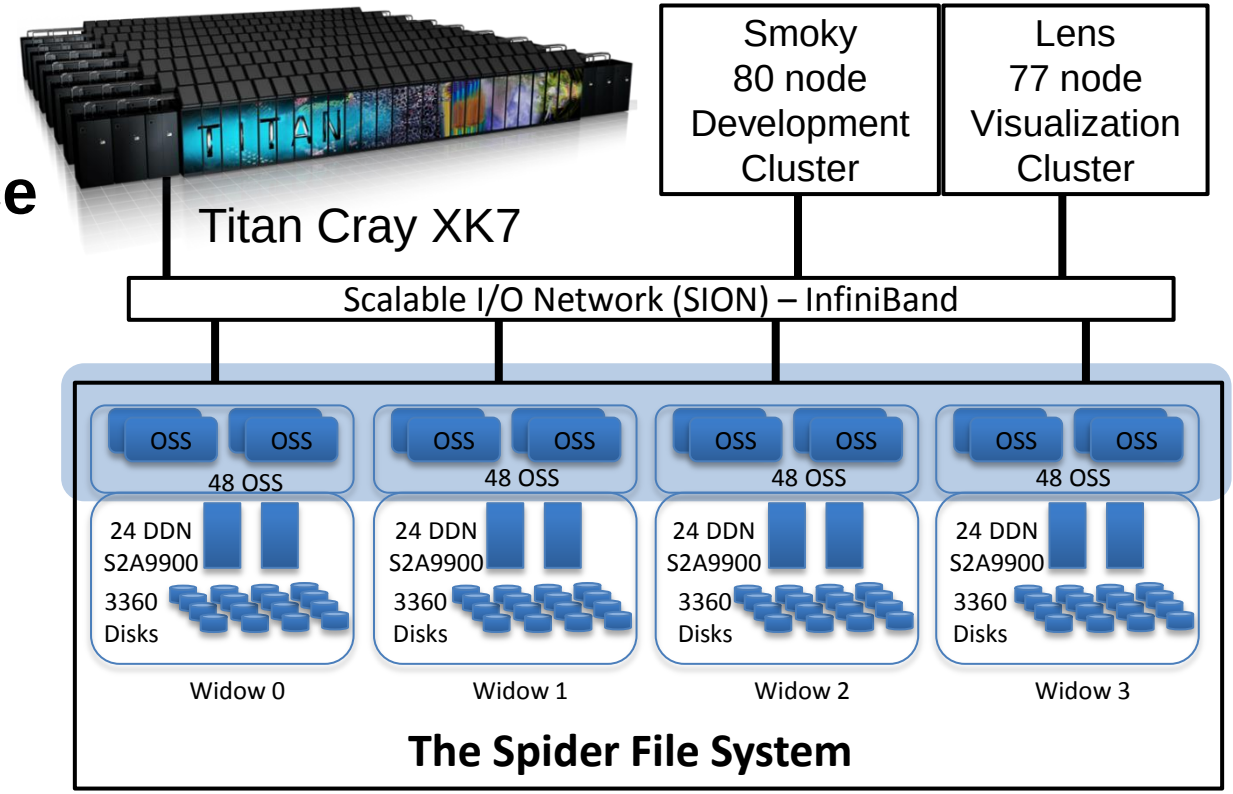
- Development effort
- Elective & Voluntary
- Performance overhead (2%-8%)
- Distinct tracing format
- Extra I/O workload



Server-side Tracing

Fine-grained RPC trace

- Detailed information
- No user effort
- Performance degradation
- Huge trace data



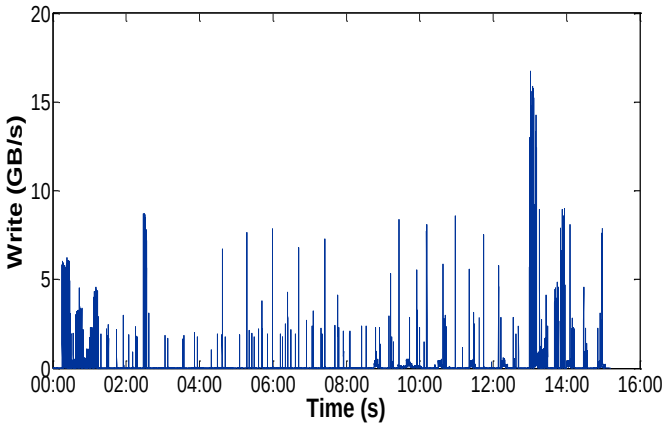
Server-side tracing tools

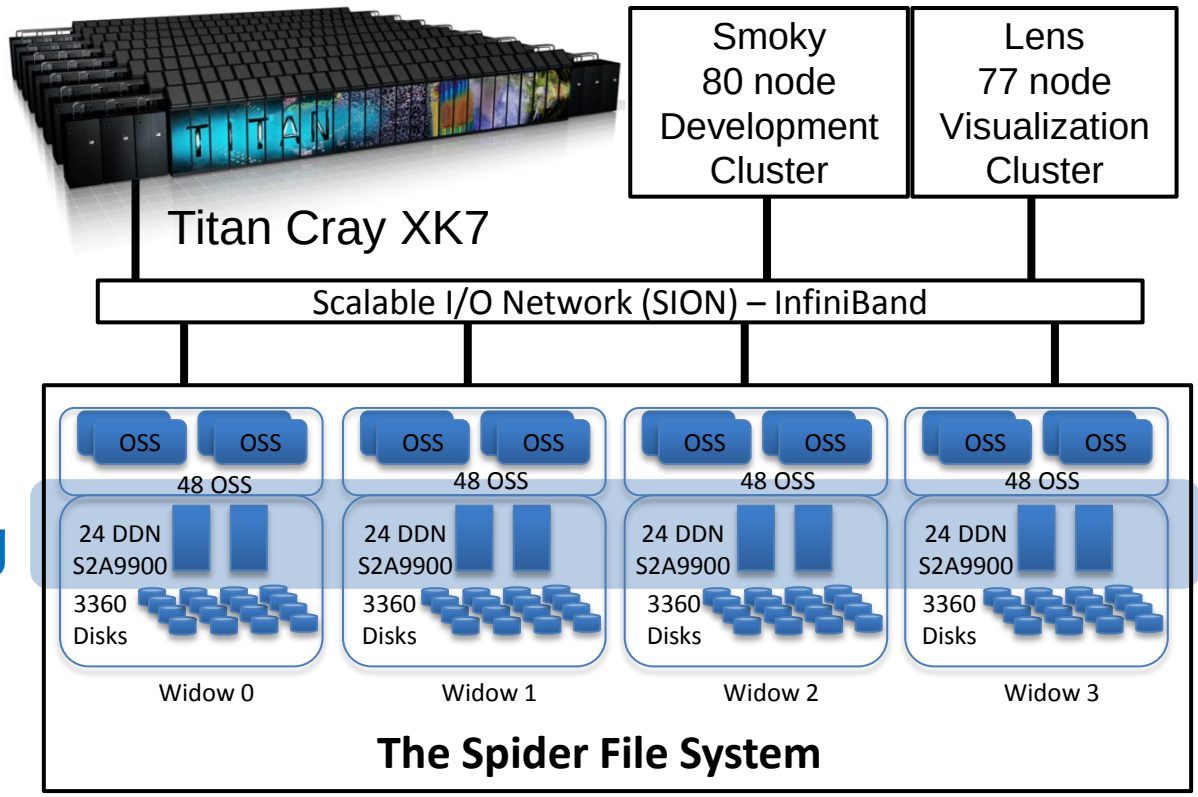
Server-side I/O Throughput Logs

I/O throughput logs

- Zero overhead
- No impact on user IO
- No user effort
- Mixed I/O traffic

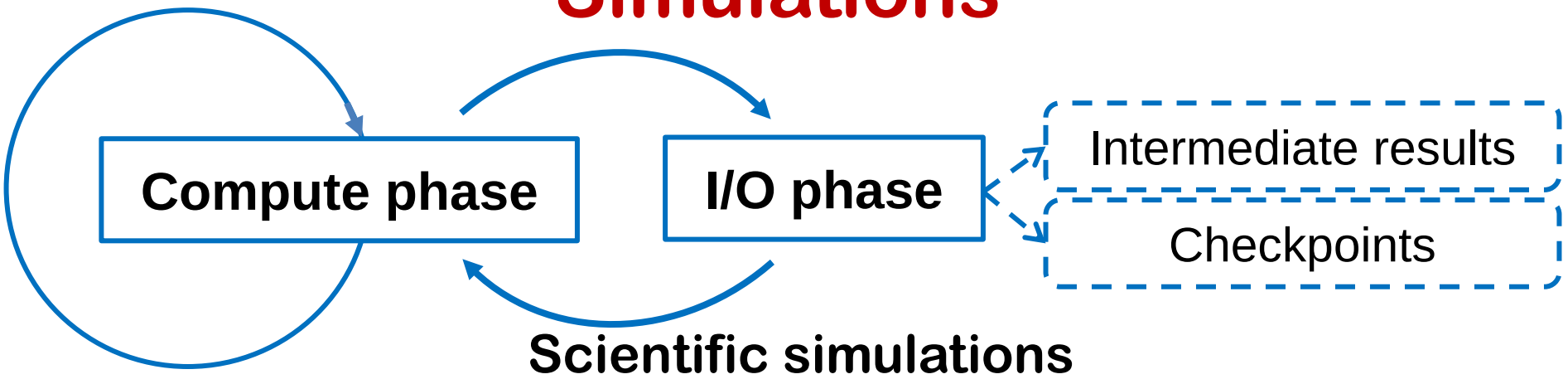
RAID controller Coarse-granule logging





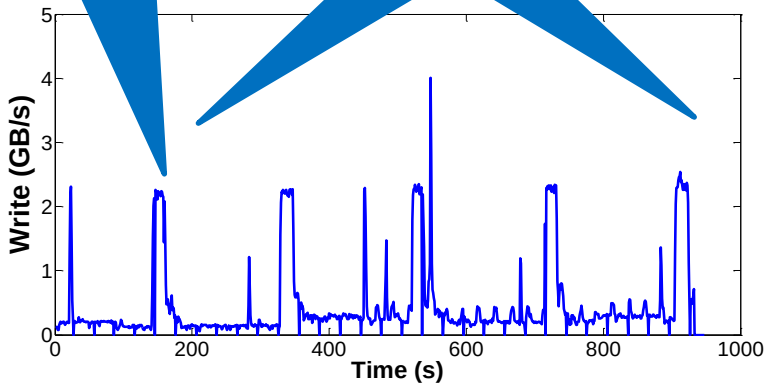
Server-side I/O throughput logs

I/O Characteristics of Scientific Simulations



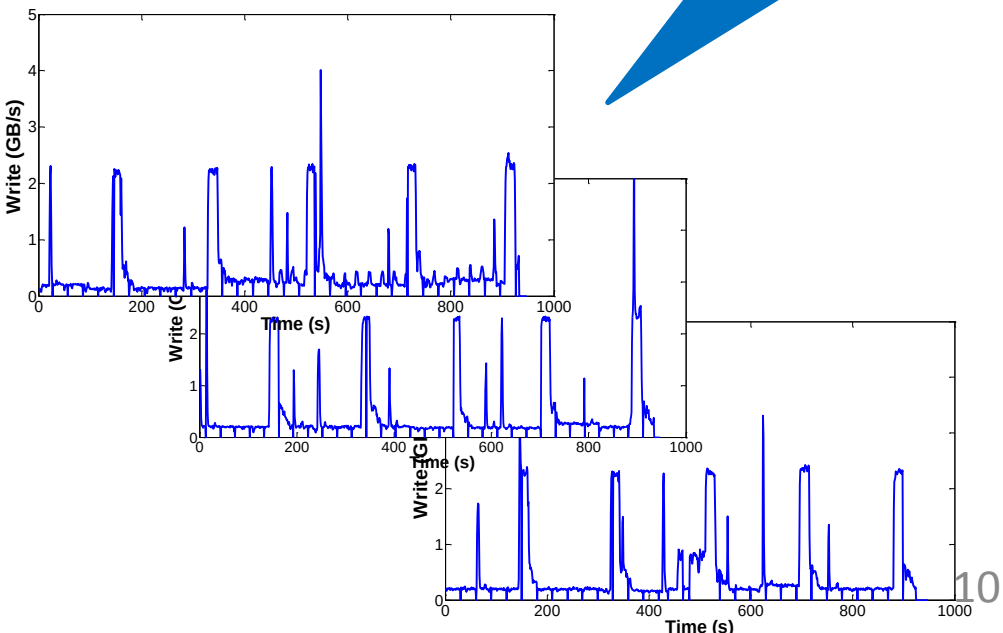
Scientific simulations

Bursty **Periodic**



Write throughput observed during one application run

Repetitive



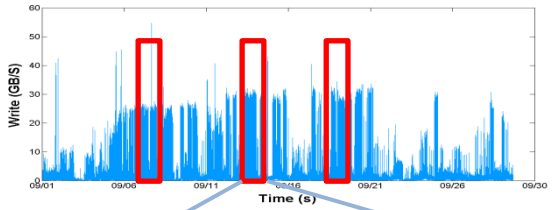
I/O Signature Identifier (IOSI)

Target App
(User ID + App ID)

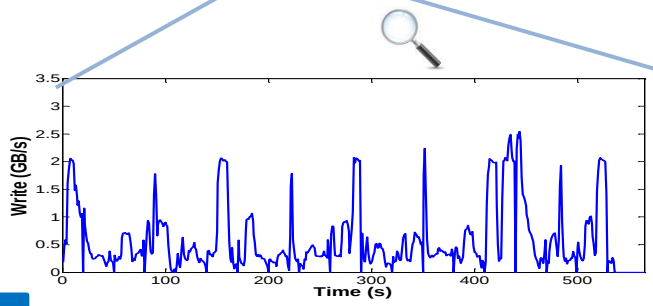
Titan's production Job scheduler logs

Start_time	End_time
2011-10-16 00:00	2011-10-16 02:01
2011-10-17 01:00	2011-10-17 04:00
2011-10-18 05:10	2011-10-18 07:20

Spider server-side Throughput logs

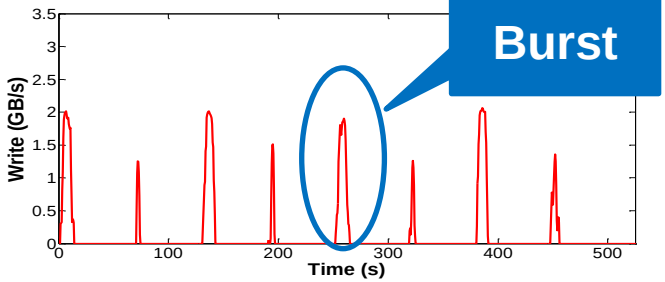


Sample



I/O throughput observed at the server side

Burst



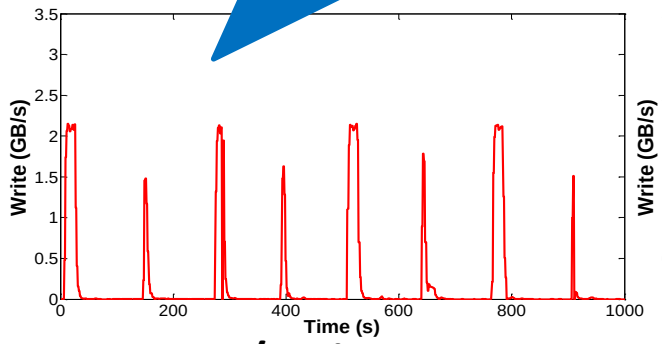
I/O signature

Commonality across multiple samples tends to belong to target application's I/O signature

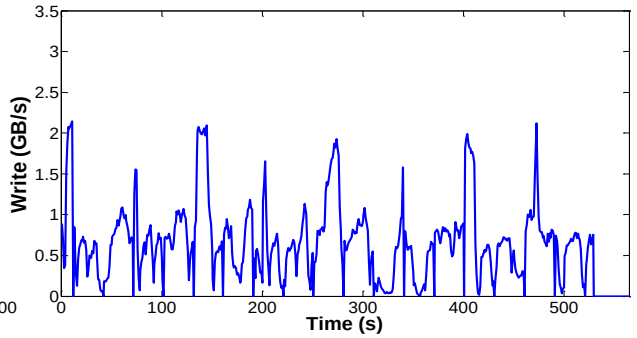
Case Study: IOR Pseudo-application

IOR: Popular parallel I/O benchmark

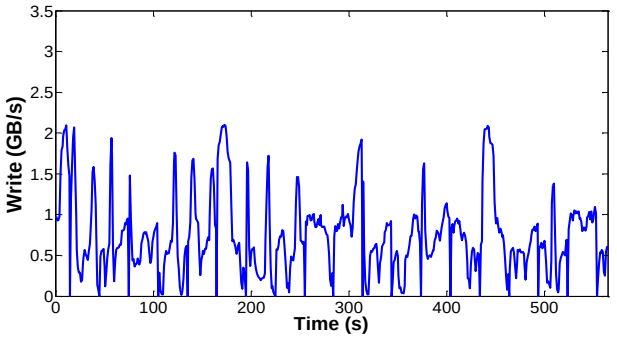
Titan's maintenance window



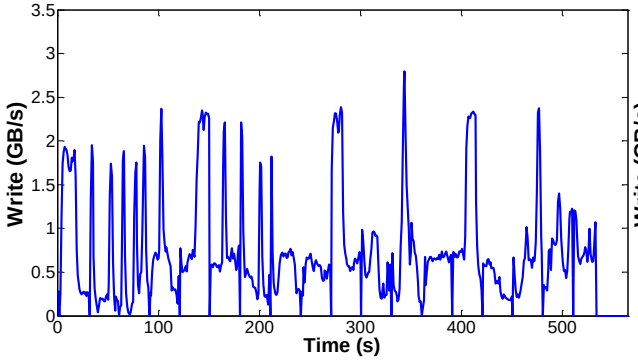
I/O Signature



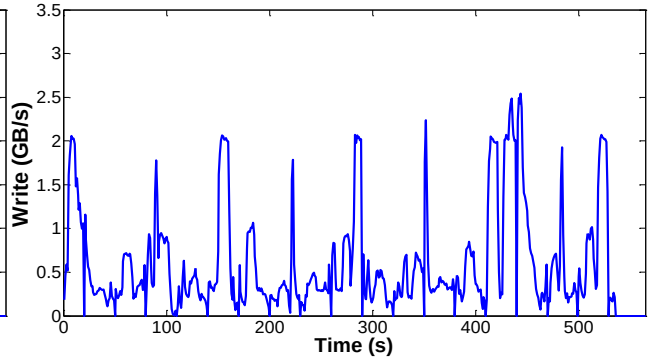
Sample 1



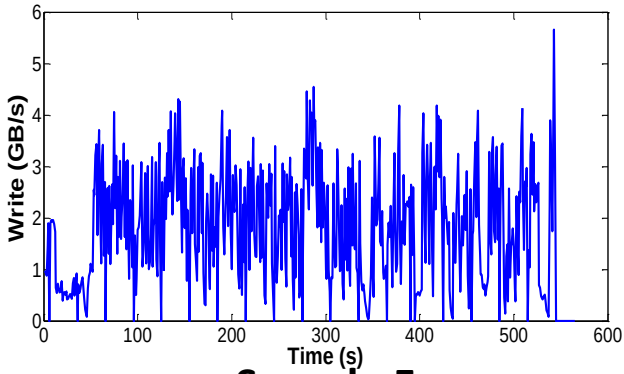
Sample 2



Sample 3



Sample 4



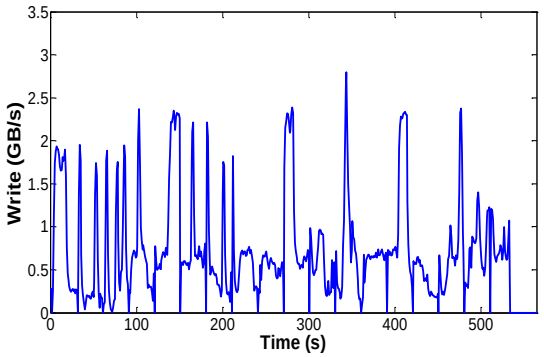
Sample 5

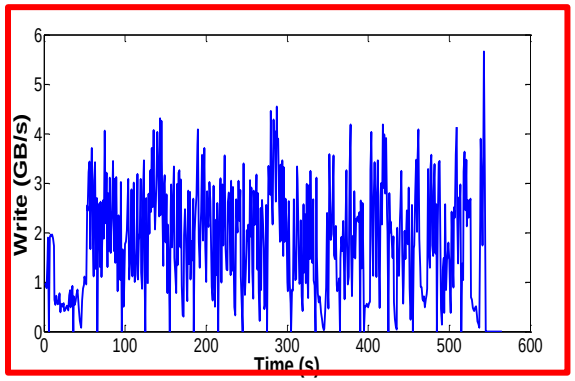
IOSI Challenges

Background noise

Random

Unpredictable

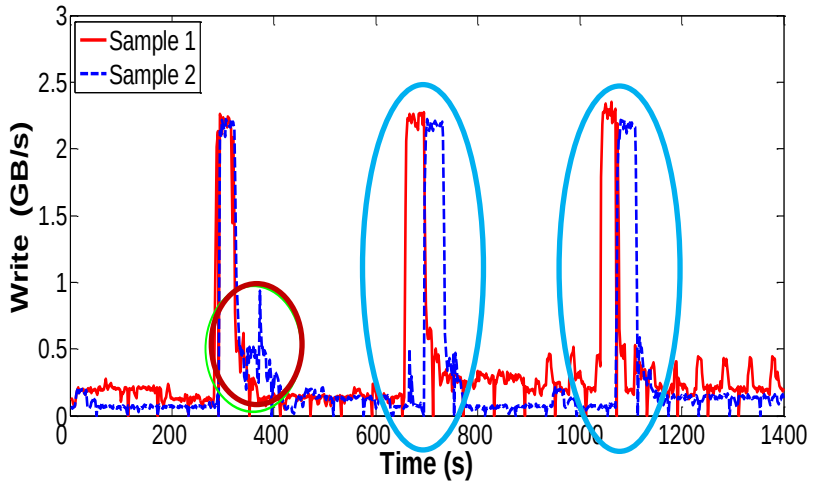




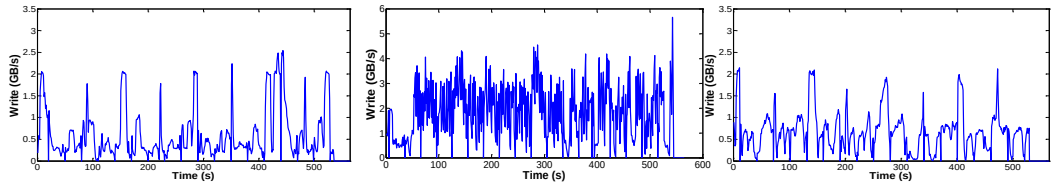
I/O drift

Distort I/O operations

Unpredictable



Processing multiple samples



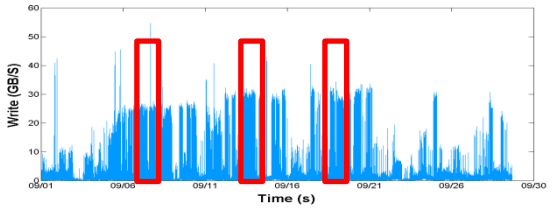
IOSI Workflow

Job scheduler logs

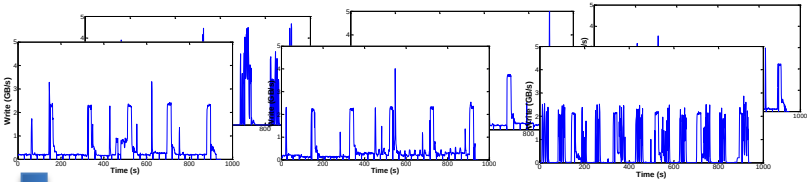
Start_time	End_time
2011-10-16 00:00	2011-10-16 02:01
2011-10-17 01:00	2011-10-17 04:00
2011-10-18 05:10	2011-10-18 07:20

Target App
(User ID + App ID)

Throughput logs



IOSI Input

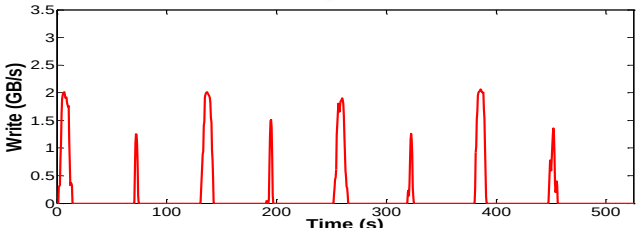


Sample set

IOSI

Data preprocessing → **Per-sample wavelet transform** → **Cross-sample I/O burst identification**

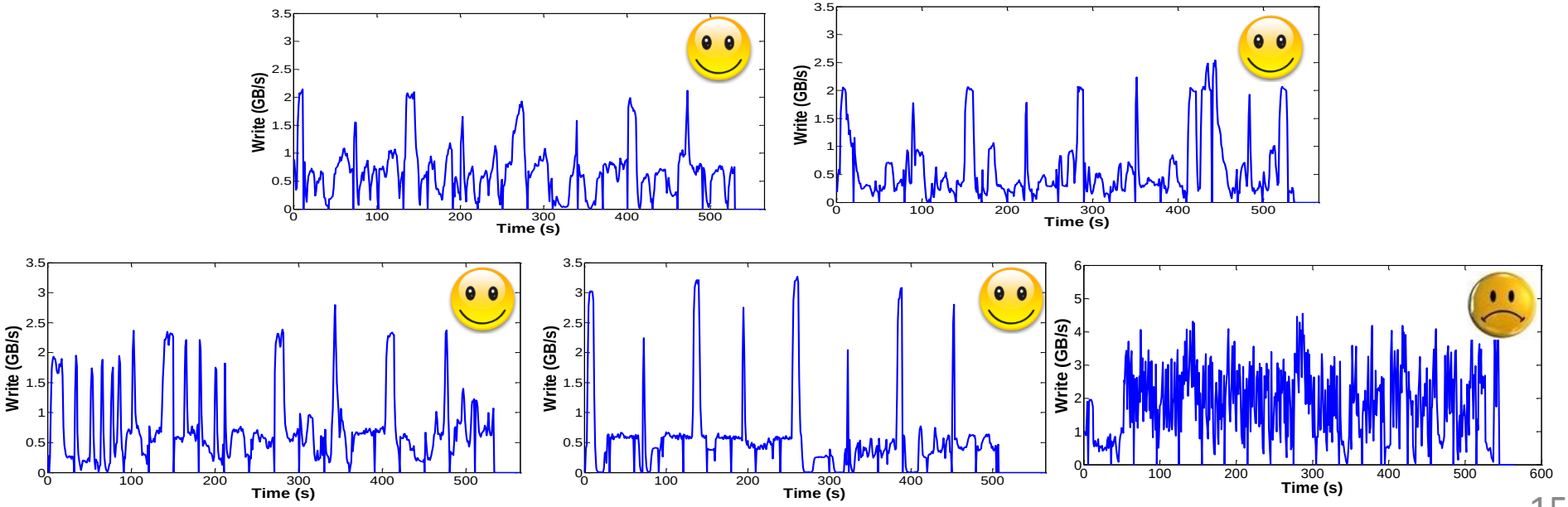
IOSI Output



Stage 1: Data Pre-processing (I)

PP1:Outlier elimination

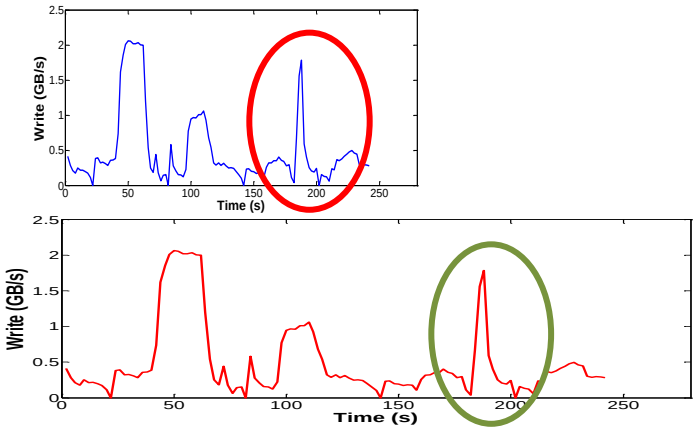
- Identify “bad samples”
- Criteria
  - Total data volume
  - Application execution time



Stage 1: Data Pre-processing (II)

PP2:Granularity refinement

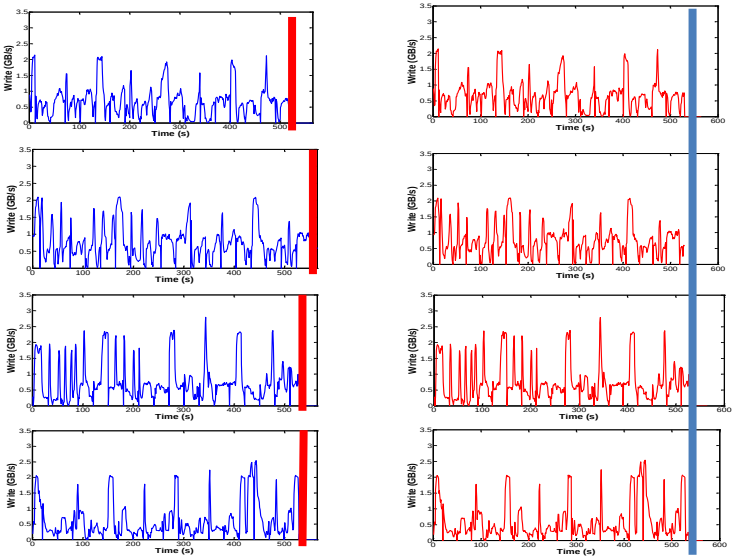
- Identify short burst
- Transform data: 2 sec -> 1 sec by inserting data points
- Linear interpolation



Linear interpolation

PP3:Duration correction

- Contention creates drifts and causes different durations
- Noise makes sample slower
- Normalize samples by discarding data points



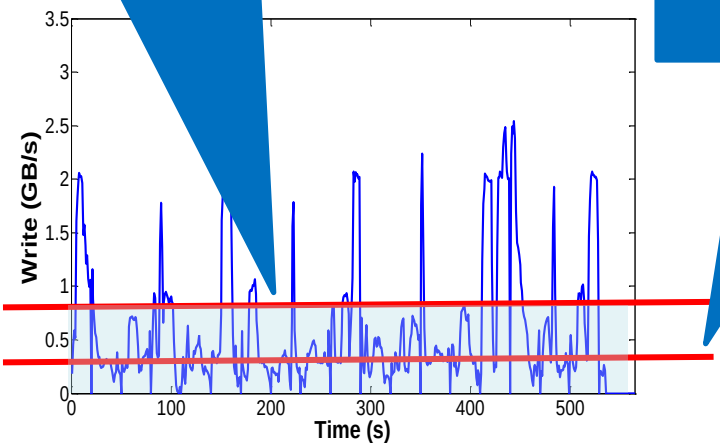
Stage 1: Data Pre-processing (III)

PP4: Noise reduction

- To remove light I/O traffic (interactive user activities, …)
- Two-level filtering

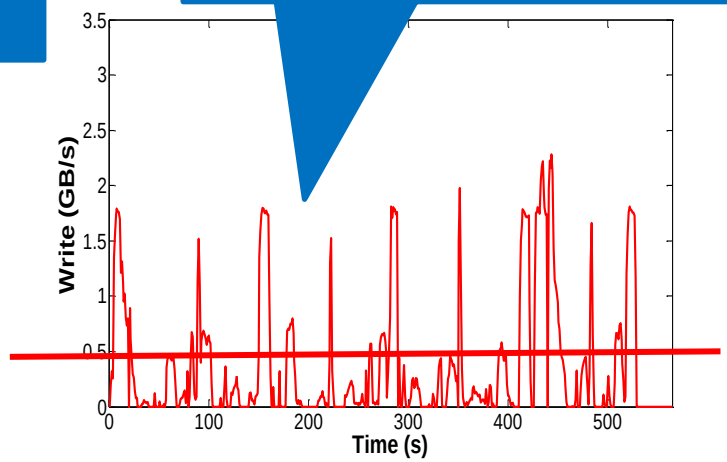
$$M = \frac{Total\ volume}{\#\ of\ data\ points}$$

$$N = \frac{Low\ points\ volume}{\#\ of\ low\ points}$$



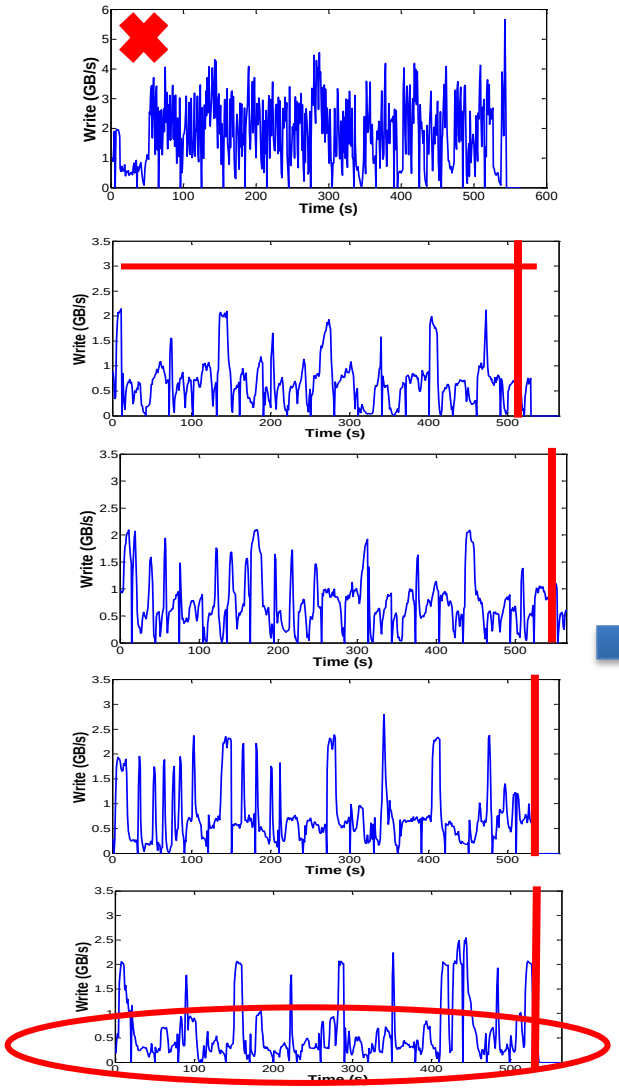
Sample before noise reduction

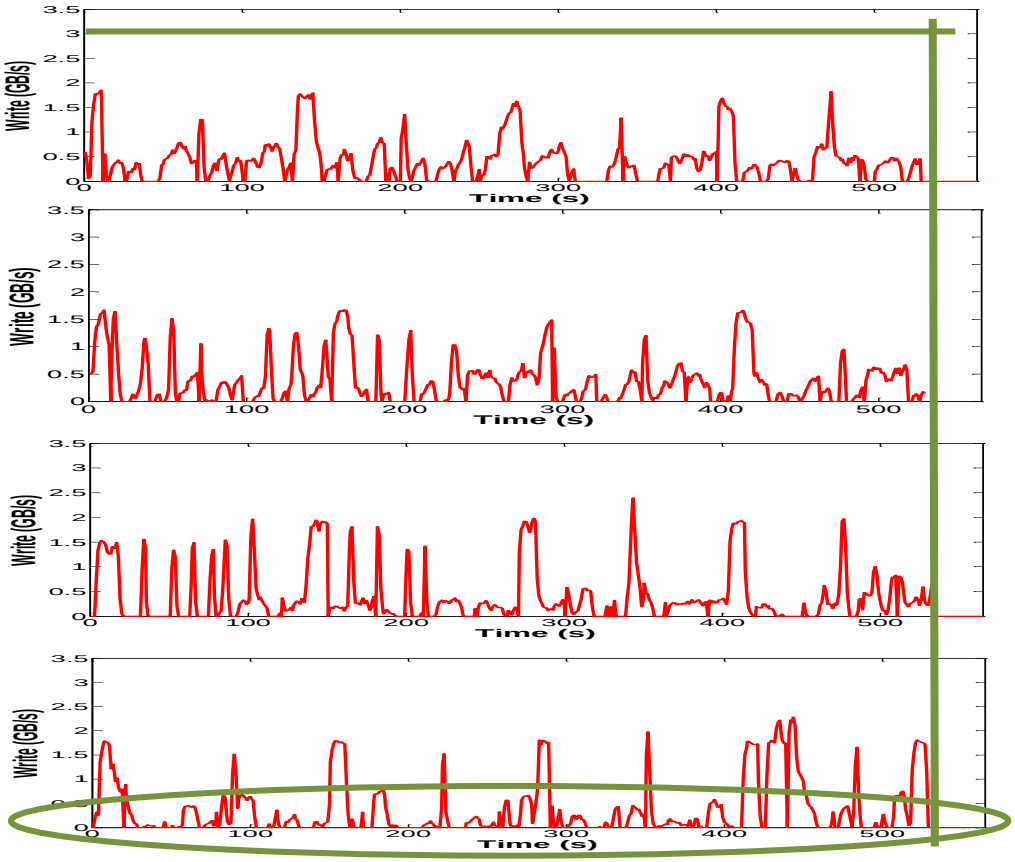
Lower all points by N



Sample after noise reduction

Effect of Data Pre-processing





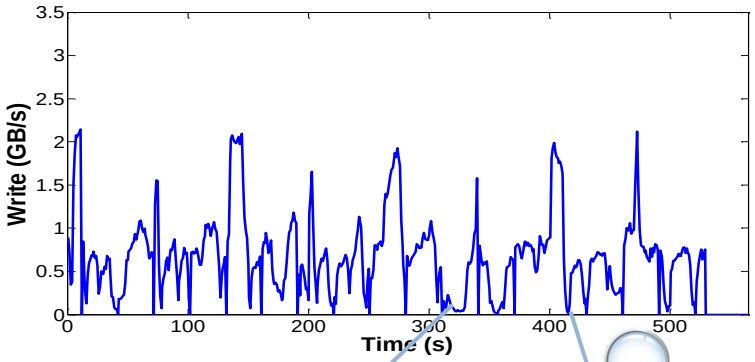
Sample **before** data pre-processing

Sample **after** data pre-processing

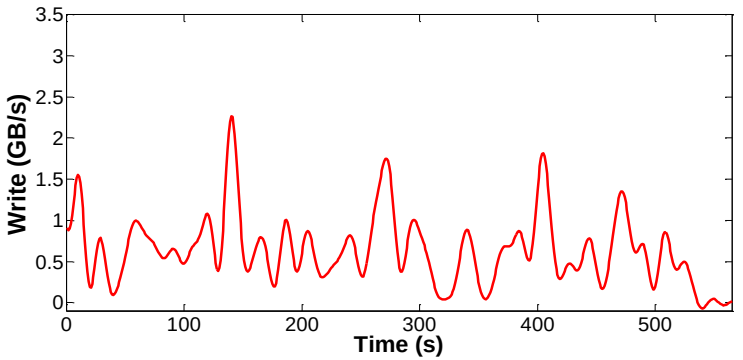
Stage 2: Per-Sample Wavelet Transform

Why wavelet transform?

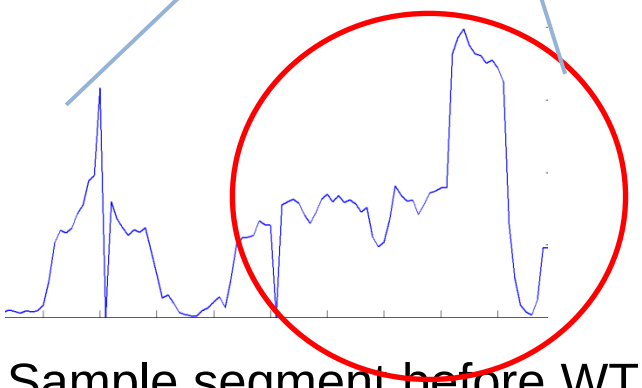
- Use burst as basic unit of signature identification
- Isolate individual burst from background noise



Sample before WT

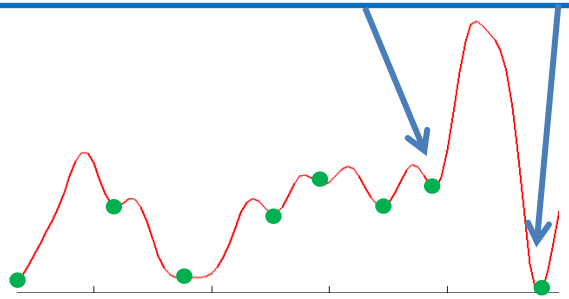


Sample after WT



Sample segment before WT

Burst: Area between two adjacent low points

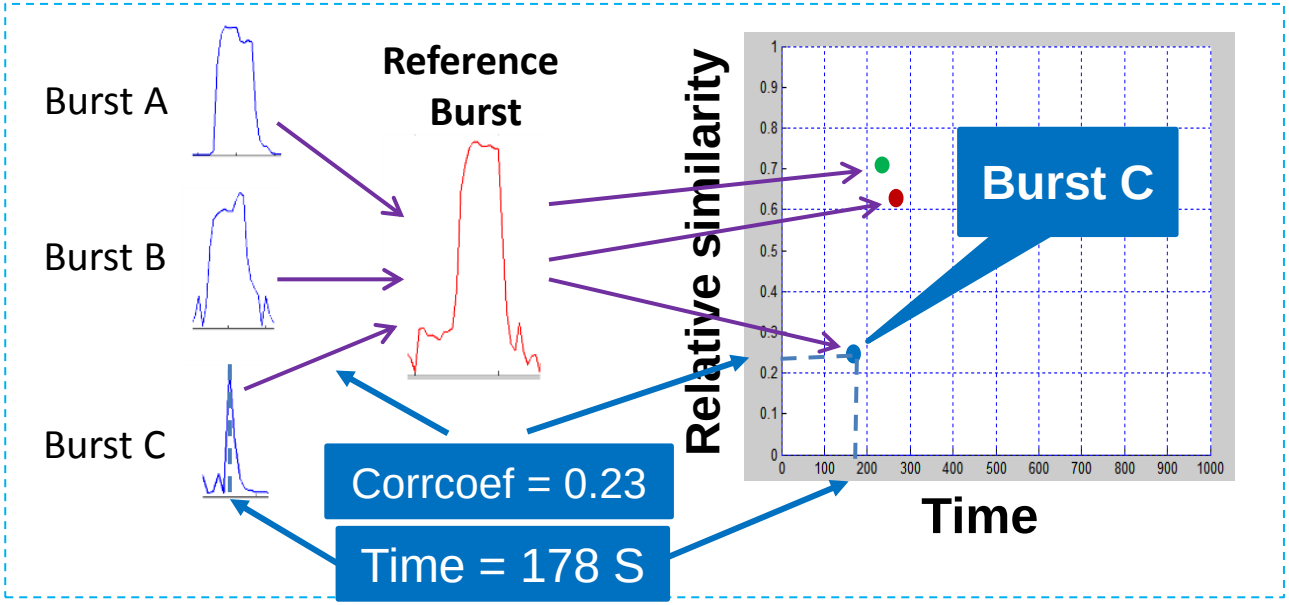


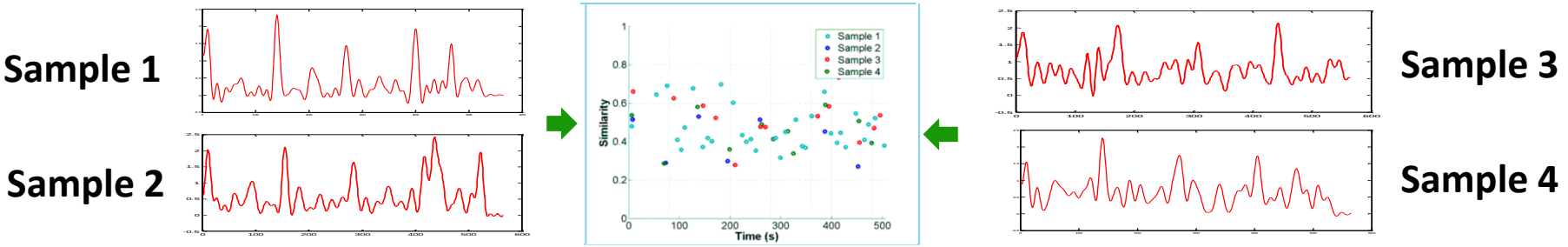
Sample segment after WT

Stage 3: Cross-sample I/O Burst Identification

Extended CLIQUE clustering *[Agrawal:SIGMOD'98]*

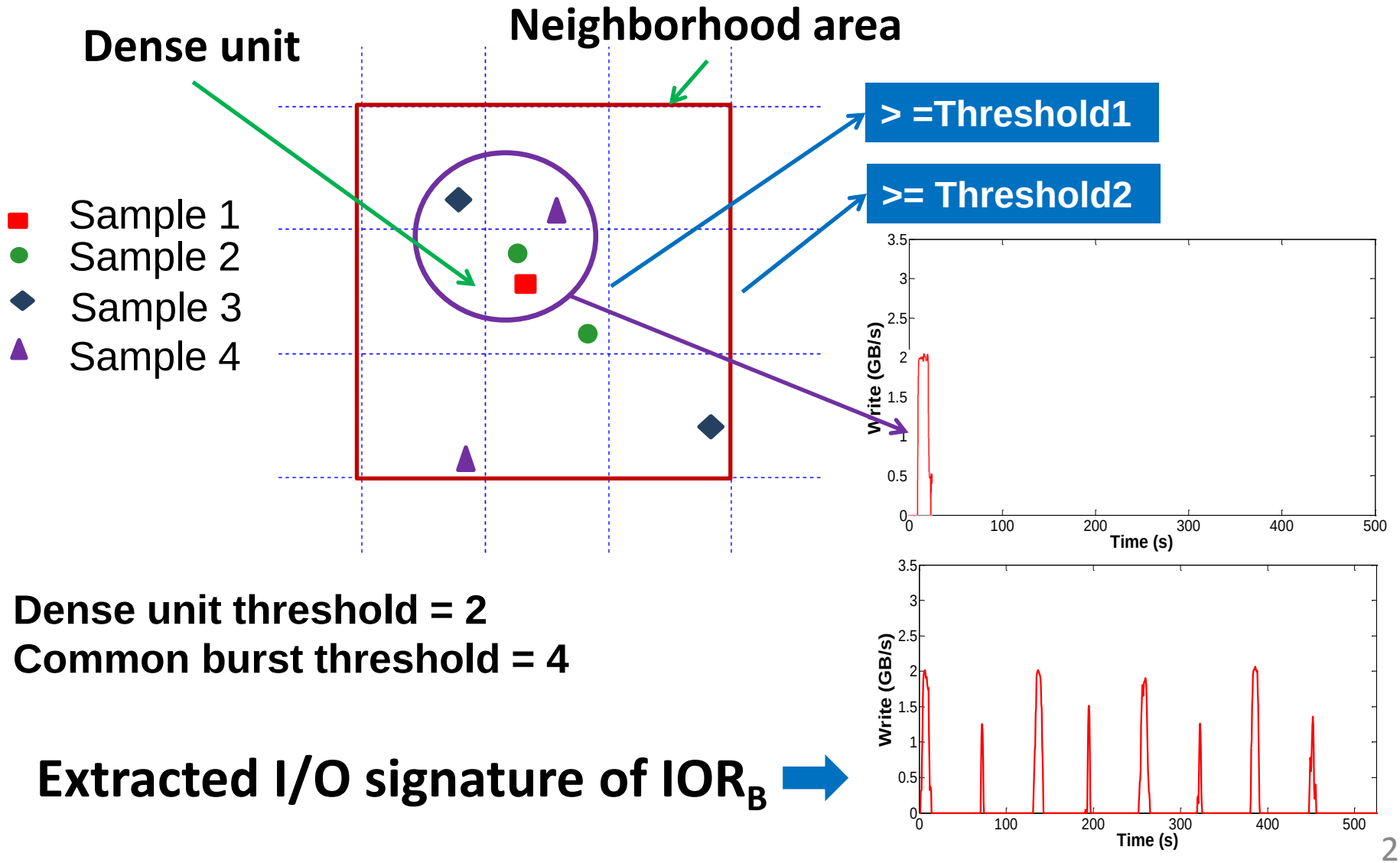
- Density-based and grid-based clustering





Map multiple samples into a 2D clustering space

Clique Clustering Example



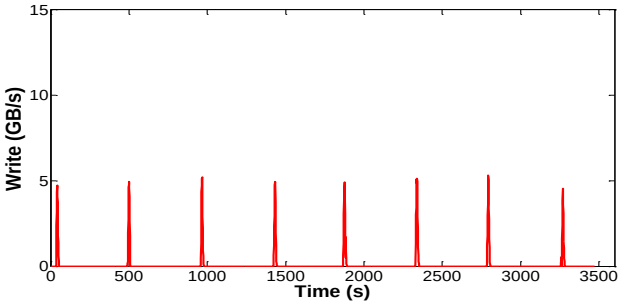
Dense unit threshold = 2
Common burst threshold = 4

Extracted I/O signature of IOR_B

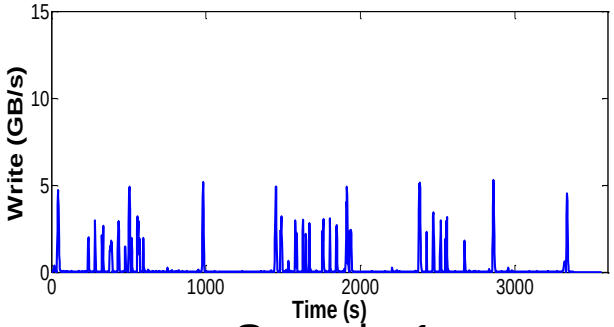
Evaluation

- IOR
  - Configurable I/O benchmark
  - Write-intensive pseudo-applications
  - Smoky, 256 processes
- S3D
  - Simulation of turbulent reacting flows
  - Titan, 18000 processes

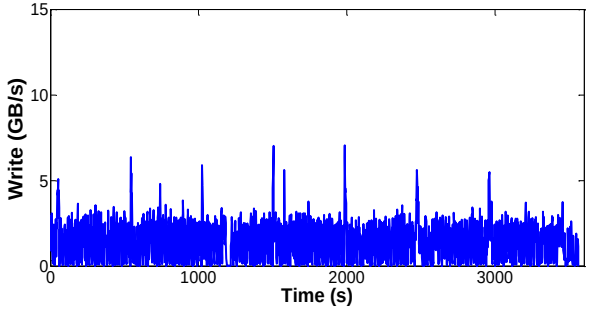
Result Comparison in S3D



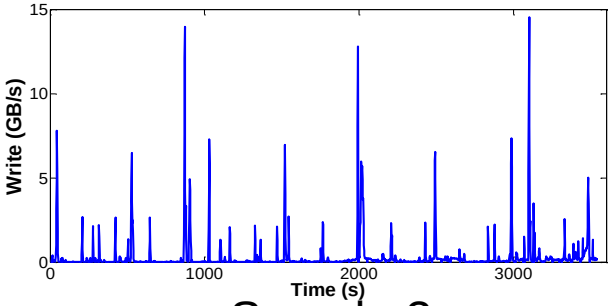
I/O Signature



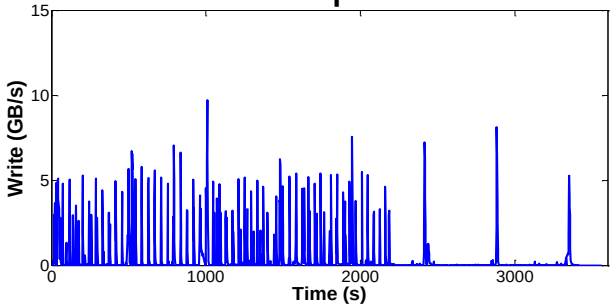
Sample 1



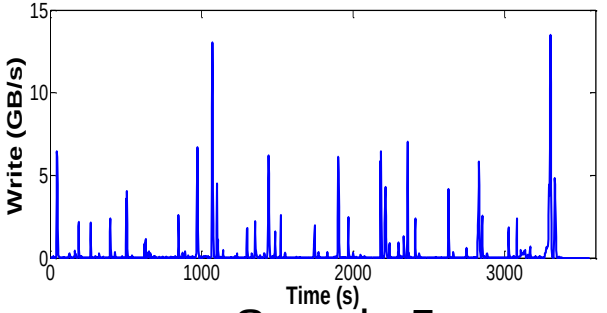
Sample 2



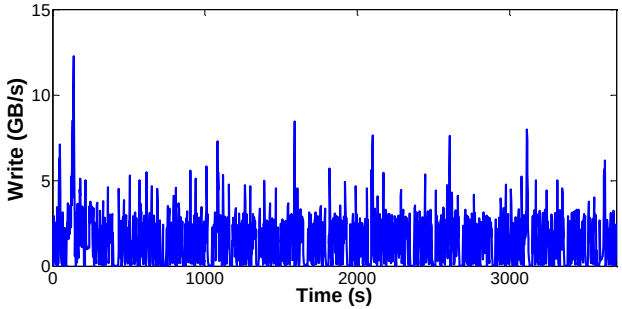
Sample 3



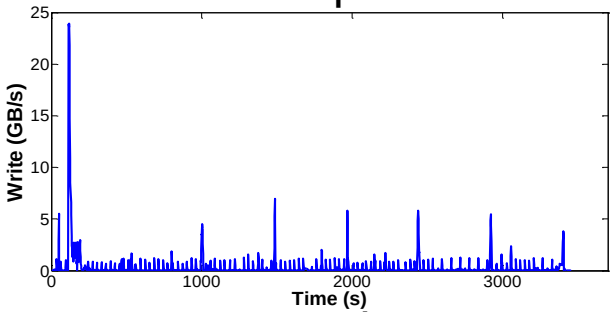
Sample 4



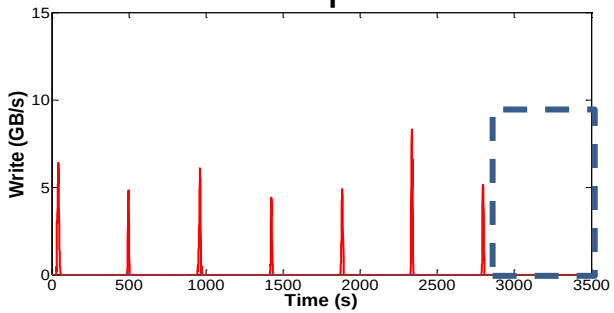
Sample 5



Sample 6



Sample 7

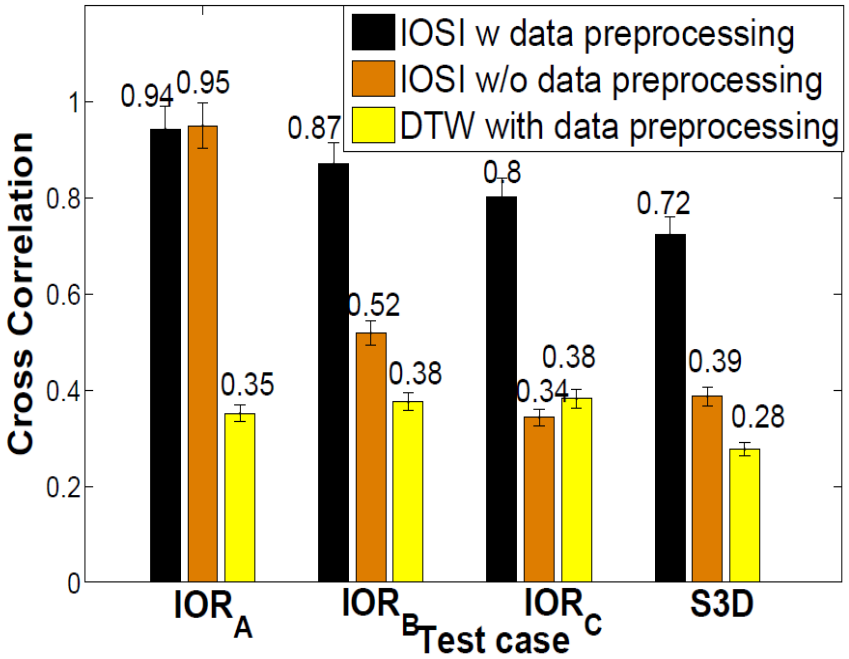


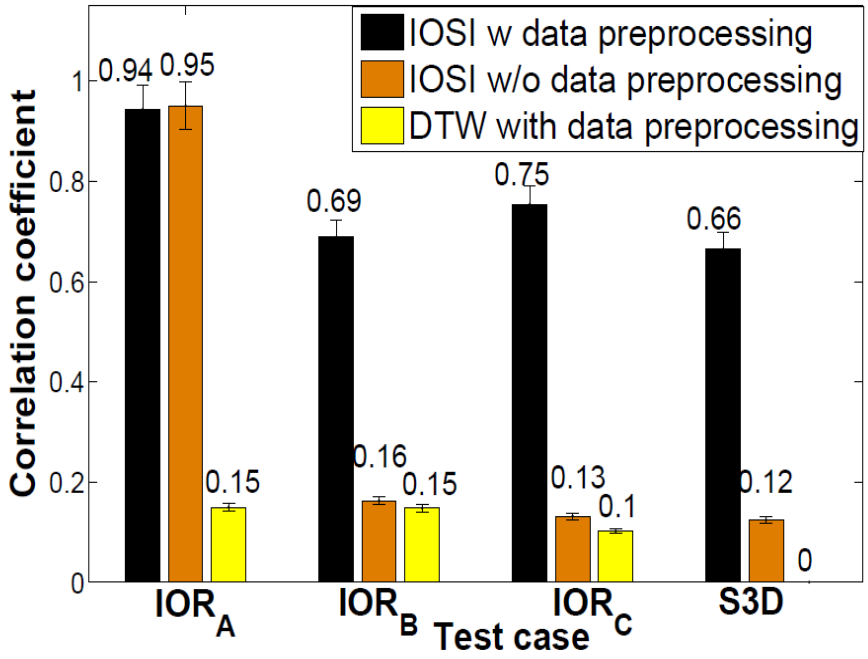
Extracted I/O signature by IOSI with data preprocessing

Accuracy Comparison

Dynamic Time Warping (DTW)

- Pair-wise data alignment technique
- Widely used in speech pattern matching





Accuracy comparison between alternative approaches

More evaluation in the paper

Closing Remarks

- Feasible and cost-effective to identify individual I/O-intensive applications' bandwidth demands
  - From aggregate, course-granule traffic logs
  - Without involving user effort

- Applicable to commercial settings?
  - Useful in addressing “I/O smear” from VMs
  - Key requirement: repeated I/O pattern
    - MapReduce, data analytics, sequential scientific computing
  - Extra challenges
    - Signal vs. noise level
    - Complexity brought by caching
      - Signatures more resource-dependent

Q&A

Yang Liu
yliu43@ncsu.edu
North Carolina State University

Reference

- [1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD’ 98), 1998.