Extracting Flexible, Replayable Models from Large Block Traces

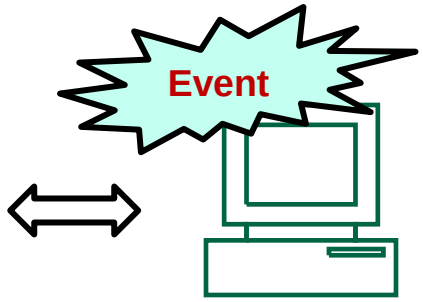
Vasily Tarasov[1], Santhosh Kumar[1], Jack Ma[2],
Dean Hildebrand[3], Anna Povzner[2],
Geoff Kuenning[2], Erez Zadok[1]

[1]Stony Brook University [2]Harvey Mudd College [3]IBM Research – Almaden

Outline

1. Traces and their problems
2. Workload models suitability
3. Design of the model extractor
4. Evaluation
5. Conclusions

Traces



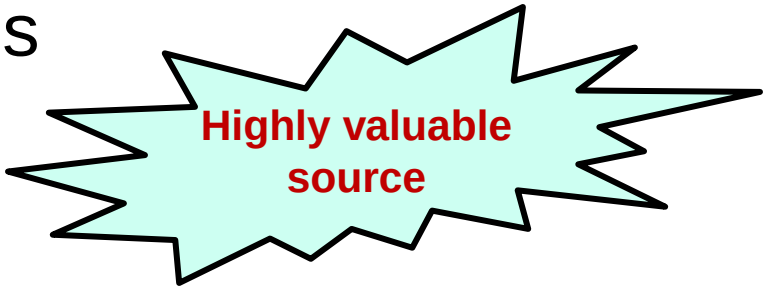
Trace record

Time-stamp	Opera-tion	I/O size	Offset
0	read	4096	0
0.5	read	4096	4096
0.7	read	4096	8192
1.3	write	8192	28762
1.5	write	8192	32768
1.6	read	4096	12288
2.0	read	4096	14384

- In general case, any event can be traced (process forking, file accesses, user logins)
- Timestamp is a common field
- Other fields depend on the specific events traced
- We used block traces
- Our approach is valid for any trace

Trace Use Cases

- Workload analysis and characterization
  - Tune existing systems
  - Design new systems



- Trace replay
  - Evaluate, compare, and validate system behavior

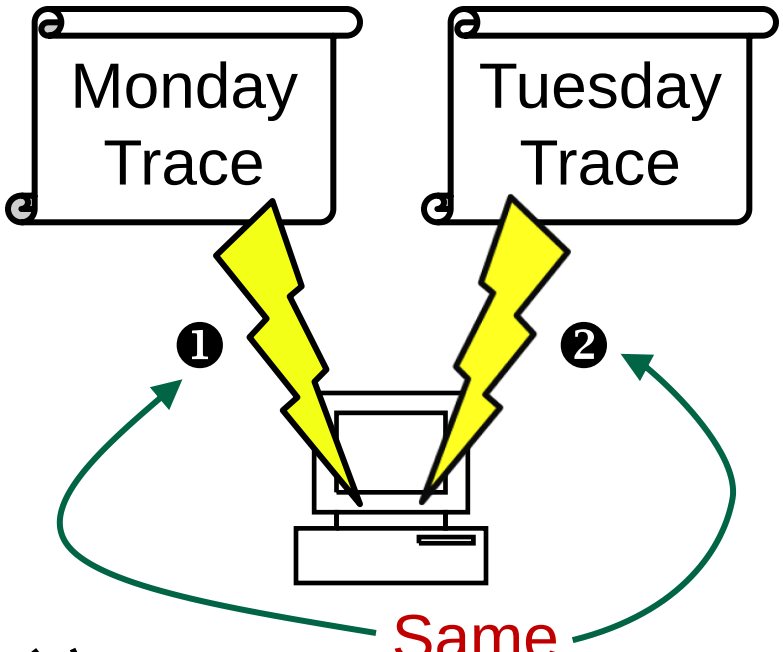
Problems with Trace Replay

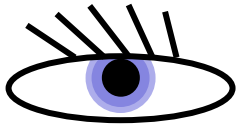
- Large in size
  - Disturb results
    - Replayer bottlenecks on I/O
    - Cache pollution
  - Hard to distribute
- Static objects
  - Hard to intelligently and systematically modify the workload
  - Not easy to compare

Outline

1. Traces and their problems
2. **Workload models suitability**
3. Design of the model extractor
4. Evaluation
5. Conclusions

Statistics Matter





Observe system's response:

- Latency
- Throughput
- Power
- Disk utilization

- Monday's trace is not exactly the same as a Tuesday's trace
- Responses are the same
- Statistics of the workload in the traces impact the system:
  - ♦read/write ratio
  - ♦I/O size

  } Same
- Set of statistics depends on specific system

Outline

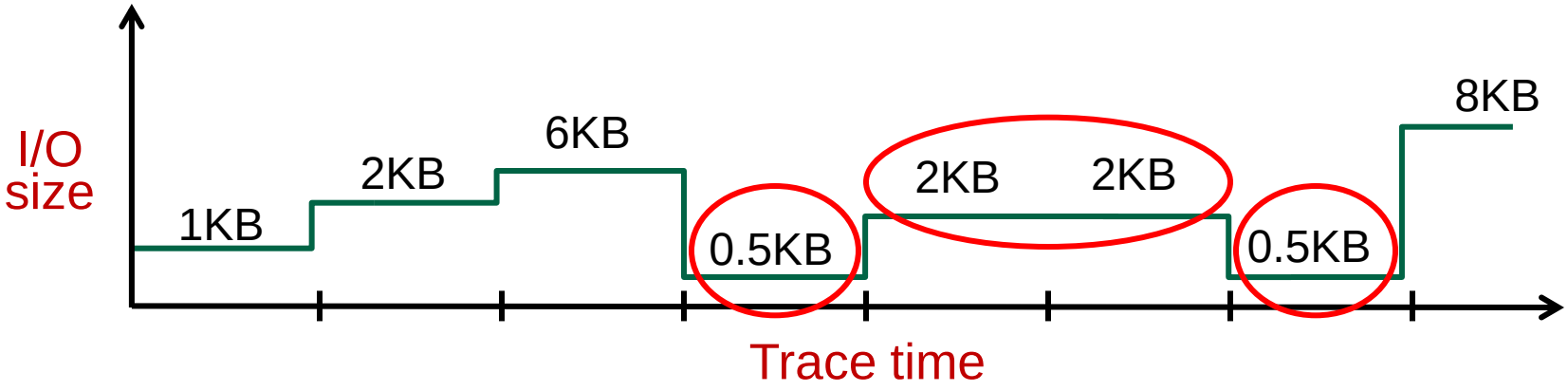
1. Traces and their problems
2. Workload models suitability
3. **Design of the model extractor**
4. Evaluation
5. Conclusions

Design Goals

- Accuracy
  - System responses match
- Conciseness
  - Small model size
- Flexibility
  - Trade model size for accuracy
  - Existing benchmarks for workload generation
- Extensibility
  - Statistics and benchmarks

Trace Chunking

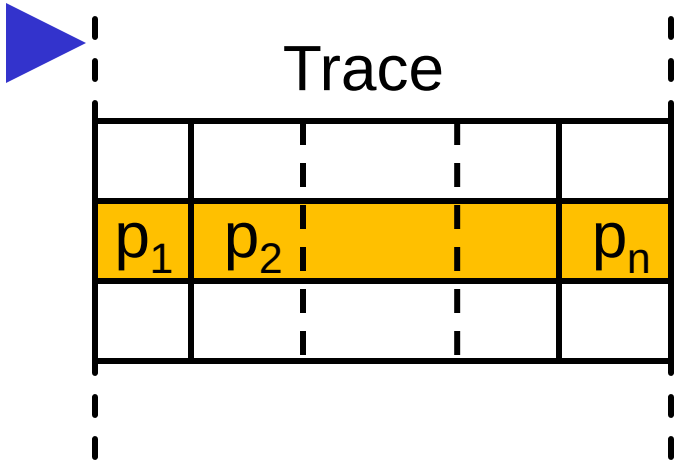
- Workload changes in the trace over time



- Chunk the trace:
  - Fixed chunking first
  - Then deduplicate chunks
  - This often results in variable chunking

Within a Chunk

- Assume stationary workload
- Feature functions



Trace field vector: $\vec{p} = (p_1, p_2, \ldots, p_n)$

Feature function: $f_1 = f_1(\vec{p}, s_1)$

s_1: state

Feature function vector:

$$\vec{f} = (f_1(\vec{p}, s_1), f_2(\vec{p}, s_2), \ldots, f_n(\vec{p}, s_n))$$

Put into a multi-dimensional histogram

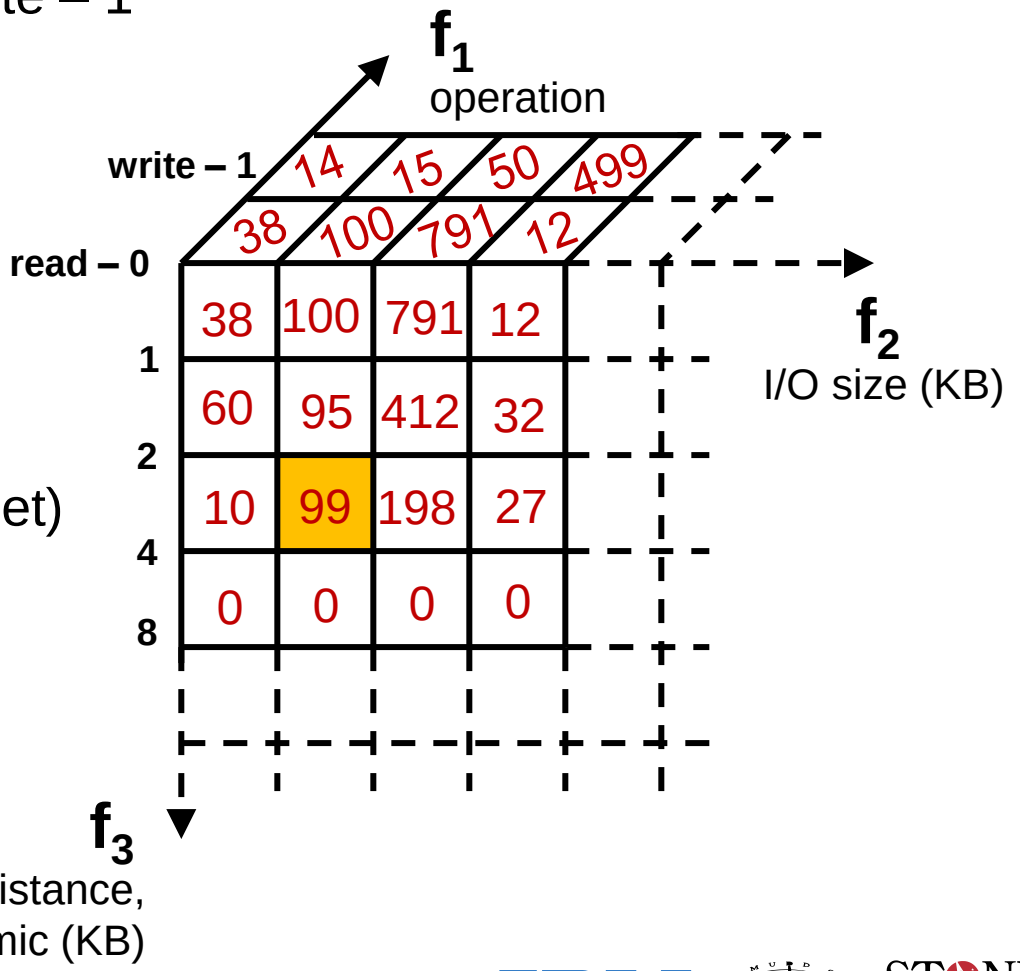
Multi-Dimensional Histogram

$\vec{p}$:

- p_1 – operation: read – 0, write – 1
- p_2 – I/O size: in KB
- p_3 – offset: in KB

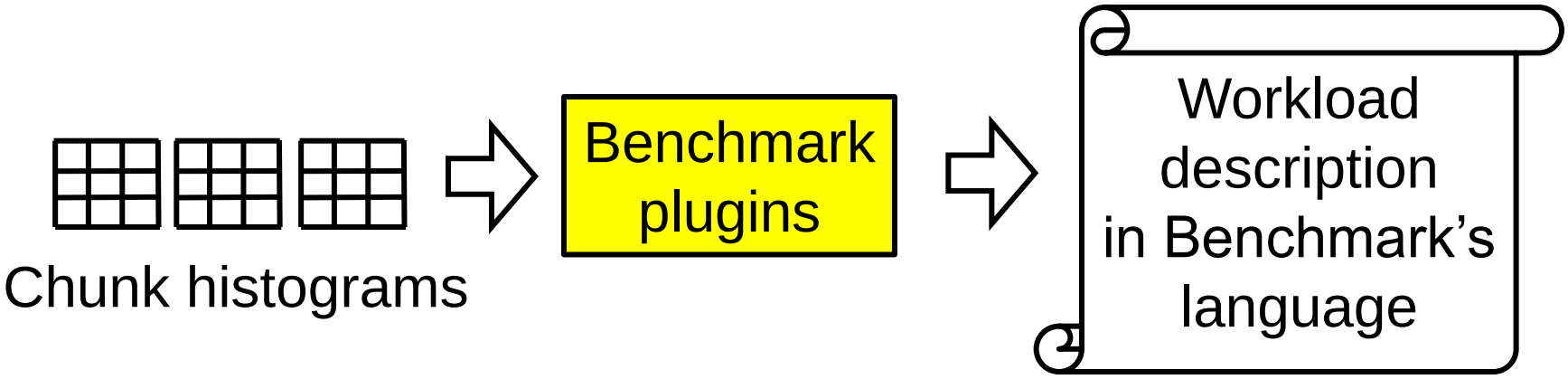
$\vec{f}$:

- $f_1 = p_1$ (operation)
- $f_2 = p_2$ (I/O size)
- $f_3 = \log(\text{offset} - s_3.\text{prev\_offset})$ (inter-arrival distance)



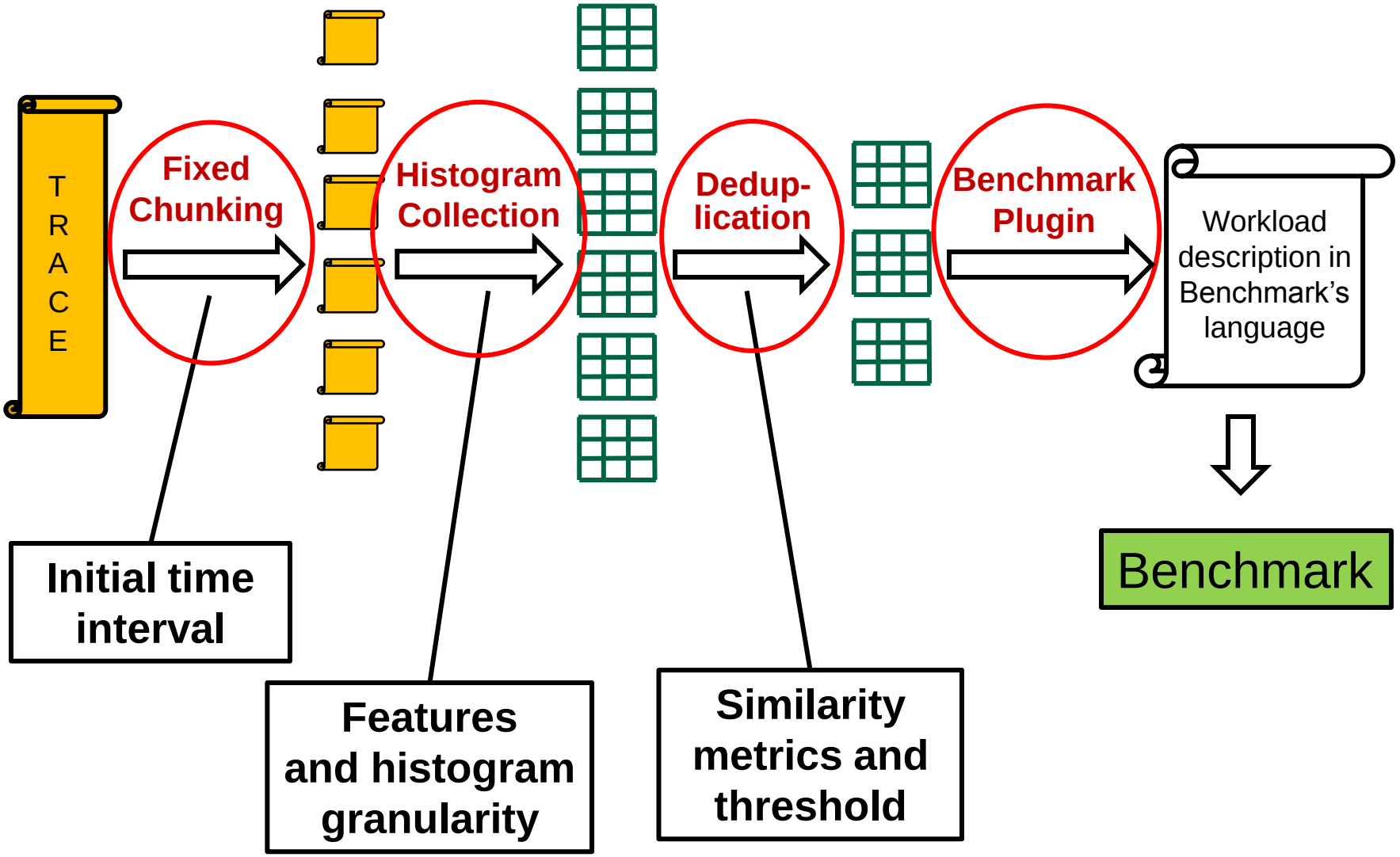
Benchmark Plugins

- Yet another workload generator?
- Use existing benchmarks instead
- Benchmark plugin:



- command line arguments for IOzone
- config files for Filebench or FIO

Overall Design



Outline

1. Traces and their problems
2. Workload models suitability
3. Design of the model extractor
4. **Evaluation**
5. Conclusions

Evaluation

1. Replayed the trace
2. Emulated workload
3. Compared response (accuracy) parameters

- Reads/sec
- Writes/sec
- Latency
- I/O Utilization
- I/O Queue length
- Request size

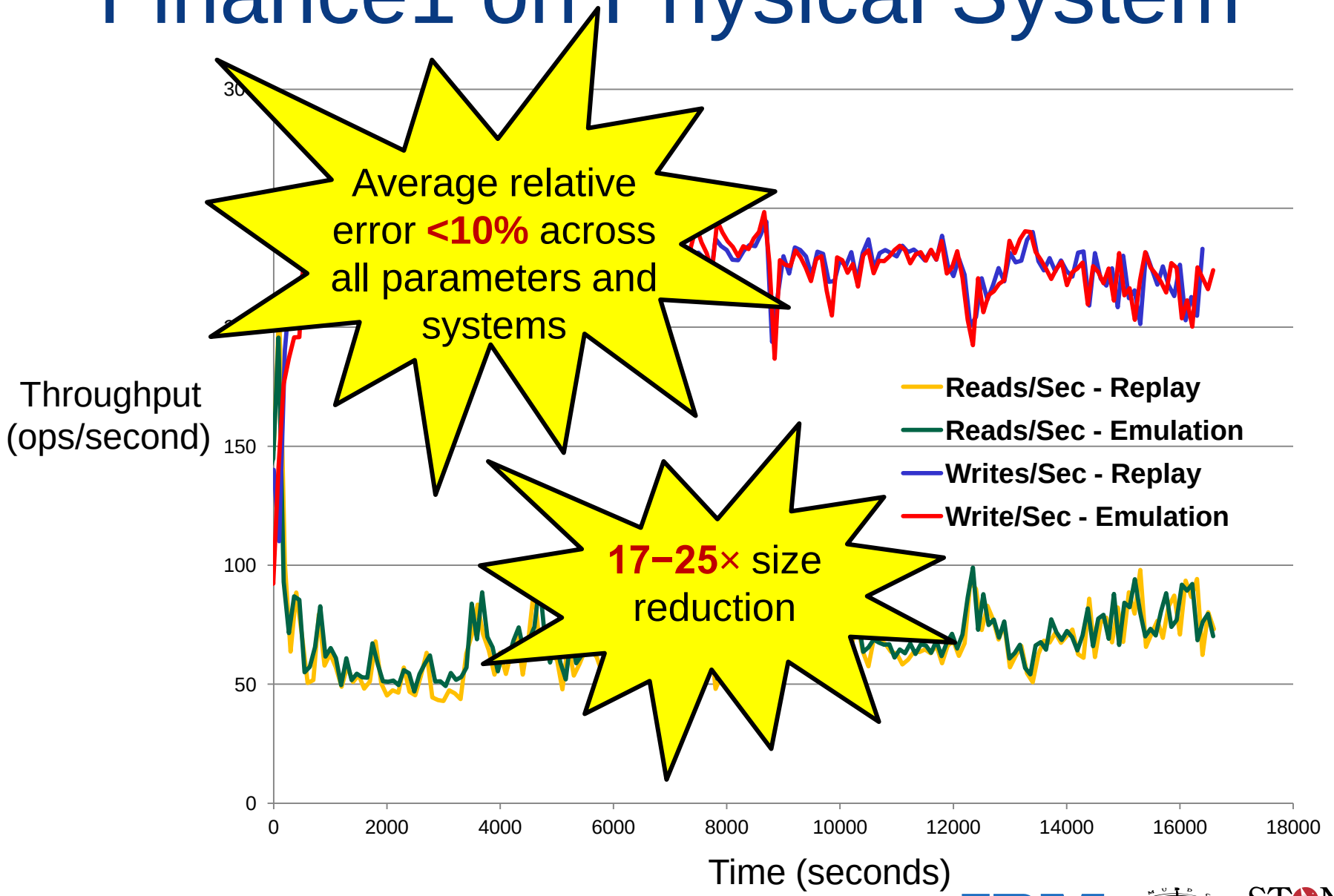
- CPU Utilization
- Memory consumption
- Interrupts
- Context Switches
- Wait Processes
- Power

Evaluation setup

- Physical Setup
  - single node with physical disk drives
- Virtual Setup
  - VM with disk image on remote GPFS server

- Finance1
  - OLTP applications
- MS-WBS
  - Microsoft build server

Finance1 on Physical System



Outline

1. Traces and their problems
2. Workload models suitability
3. Design of the model extractor
4. Evaluation
5. **Conclusions**

Conclusions

- Extractor of workload models from traces
- Multi-dimensional histograms of feature functions
- Trace chunking
- Trade off accuracy for size reduction
- Standard benchmarks

Future work

- More of everything
  - accuracy parameters, systems, traces
- File system traces
- Automatic selection of parameters
  - chunking interval, matrix granularity
- Operations on models

Extracting Flexible, Replayable Models from Large Block Traces

http://goo.gl/yFdrG

Q&A

Vasily Tarasov[1], Santhosh Kumar[1], Jack Ma[2],
Dean Hildenbrand[3], Anna Povzner[2],
Geoff Kuenning[2], Erez Zadok[1]

[1]Stony Brook University [2]Harvey Mudd College [3]IBM Research – Almaden