



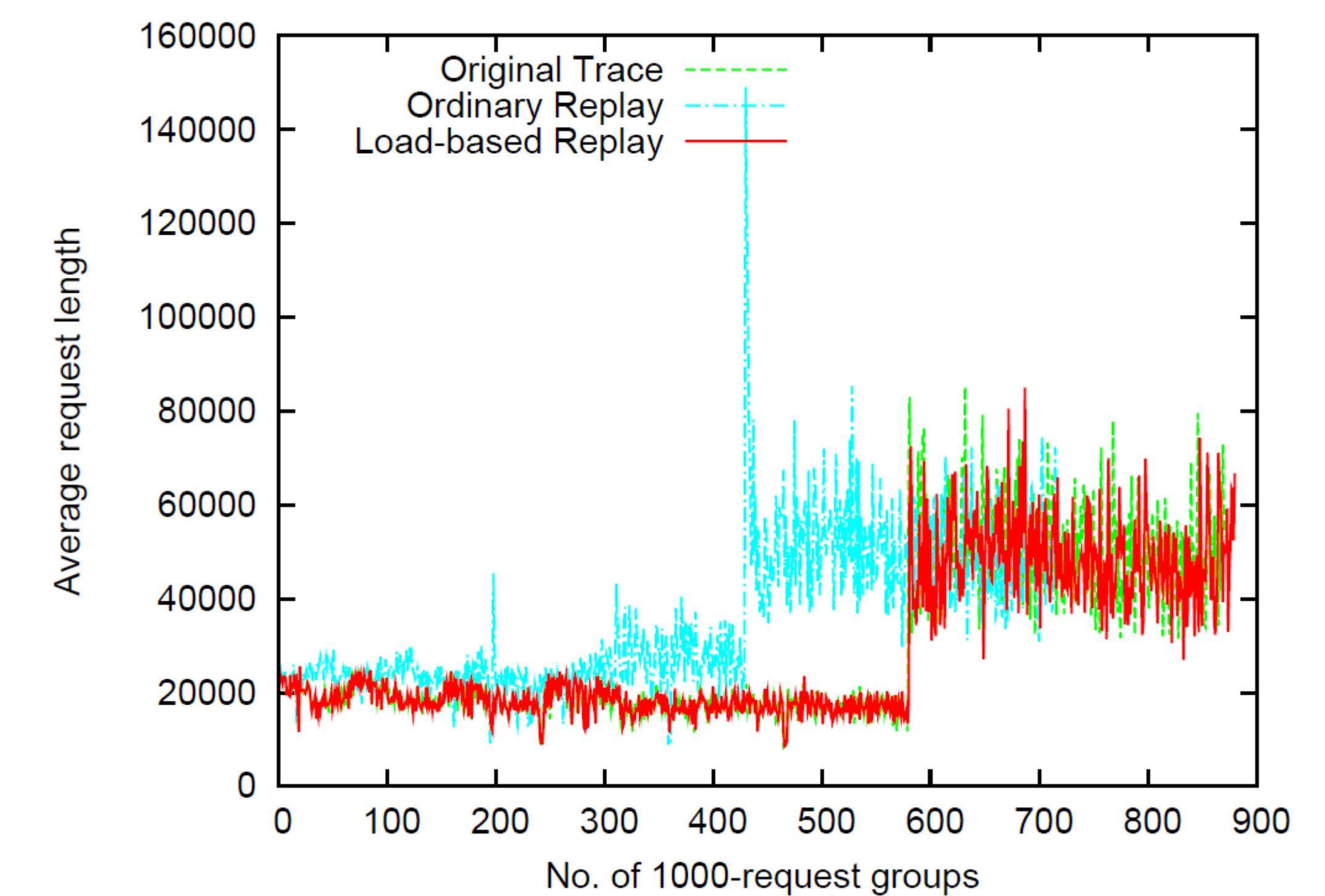
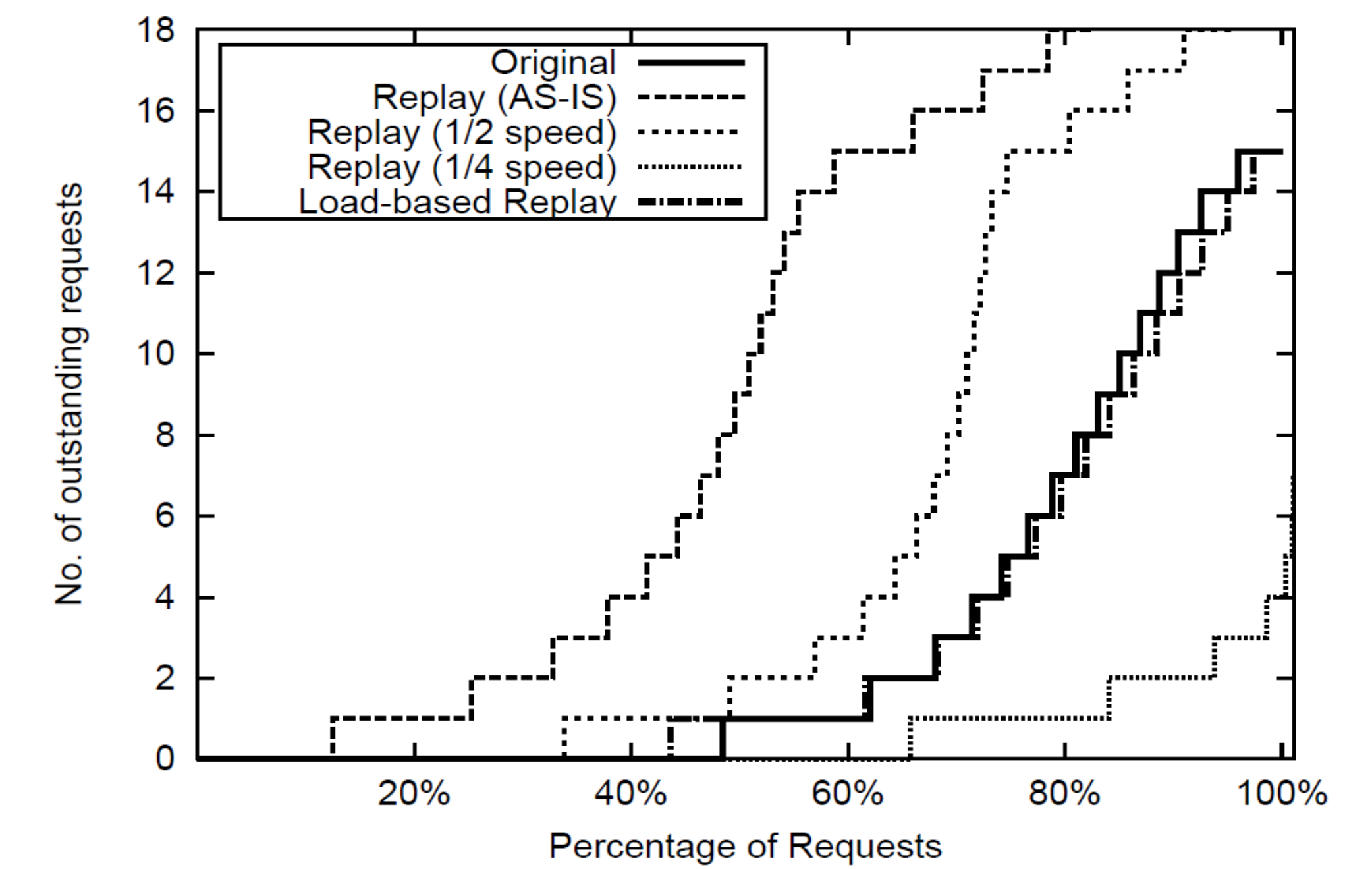
Load-Aware replay of I/O Traces

Sankaran Sivathanu*, Jinpyo Kim+, Devaki Kulkarni+, Ling Liu*

*Georgia Institute of Technology
+VMware Inc.,



Evaluation



Advantages

Applications	Representativeness
	Ease of Use
Trace Replay	Representativeness
	Ease of Use
Synthetic benchmarks	Representativeness
	Ease of Use

- Closer to running real applications compared to synthetic benchmarks
- Easier to collect and replay I/O traces than running real applications
- Preserves timing and access pattern
- Does not require additional tuning to preserve load characteristics
- Adapts to underlying hardware characteristics

Applications

- Evaluating I/O load-balancing algorithms for large scale cluster system
- Sizing experiments
- Evaluating systems for peak load scenarios while maintaining access pattern of specific applications

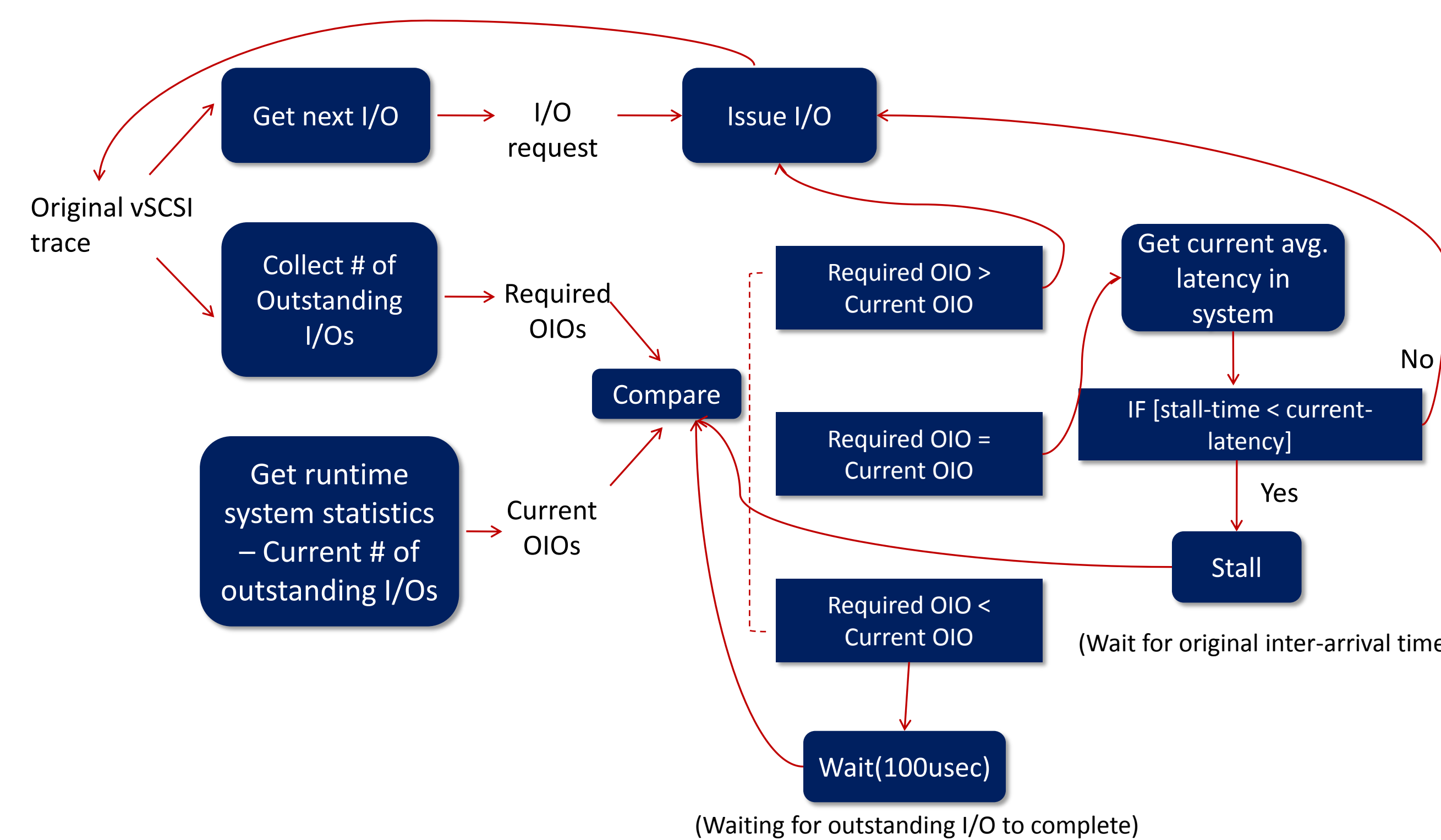
Contact

sankaran@cc.gatech.edu
jkim@vmware.com
dkulkarni@vmware.com
lingliu@gatech.edu

Overview

- Overall goal of I/O trace replay is to be representative of running real applications in a specific system environment
- I/O traces are mostly not collected and replayed in the same system environment
 - Example: Differences in OS types and versions, capacity of storage systems and servers
- Workload reaching the storage system may be entirely different during replay
- Load-aware replay is a trace replay mode, where the pace of request issues can be dynamically adjusted in order to reproduce same load characteristics in the storage system during replay

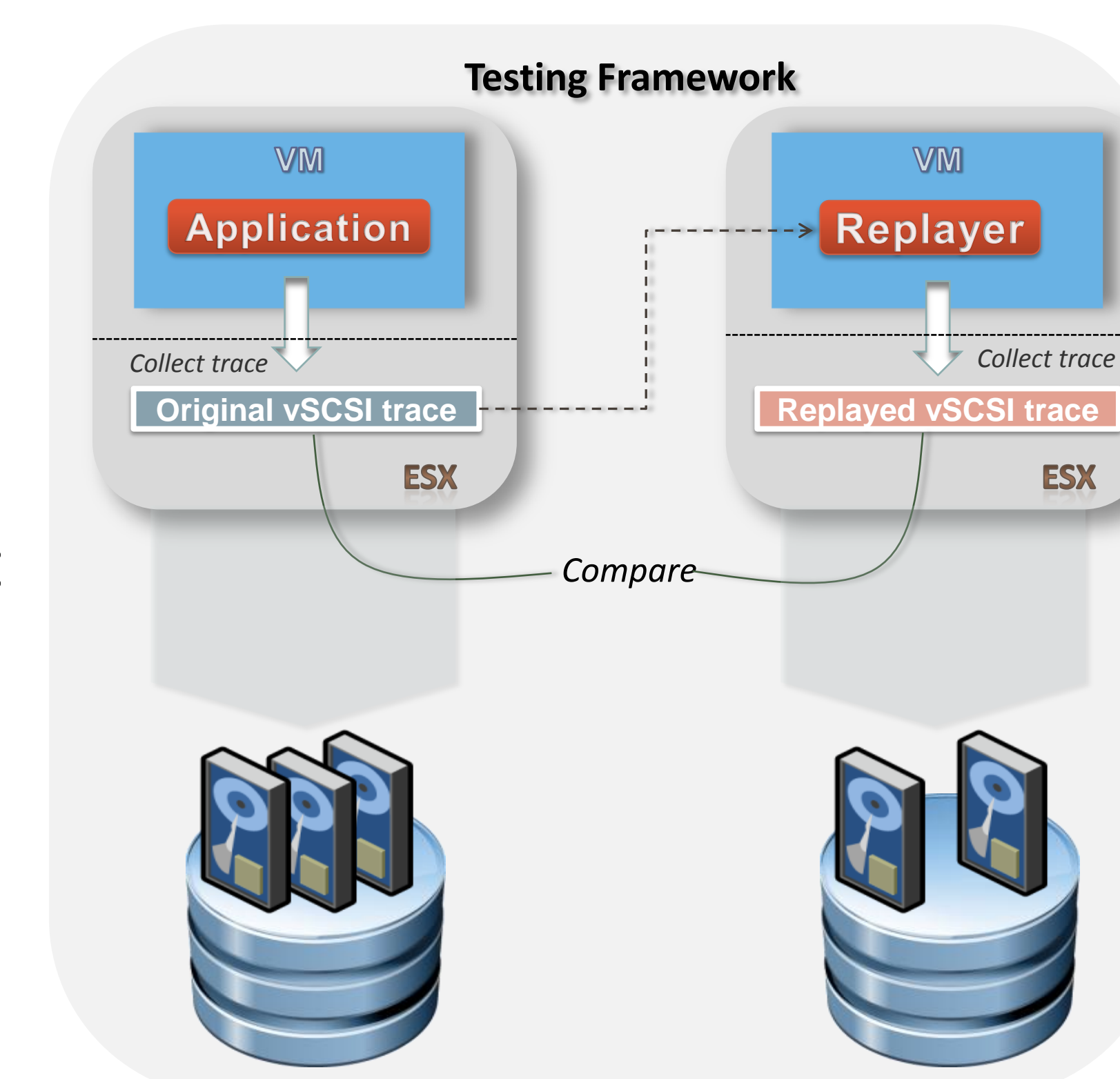
Load-based Replay



- Existing trace-acceleration/deceleration techniques are not heuristic-based.
- Goal is to reproduce load profile of source trace in any machine
- Primary focus here is to reproduce application's load profile rather than accurate timing information
- Metric considered : # of Outstanding I/O requests (OIO)
- Example use-case : Evaluation of a load-balancer in varied system setups

Challenges

- Trace collection & replay in different hosts/storage
- Per-I/O comparison between original and replay trace is difficult
 - Requests arrive out-of-order because of different OS scheduling policies
 - Requests merge or split during the replay
- Trade-off timing accuracy for representative load characteristics



Future Work

- Identify application's I/O dependencies from traces
- Develop algorithms that reproduce those dependencies irrespective of the system in which it is replayed
- Enable simultaneous replay of I/O traces from multiple hosts.

Thanks

Contributors:-Emre Celebi, Mustafa Uysal, Irfan Ahmed
Poster Shepherd:- James S. Plank