Tableau: A High-Throughput and Predictable VM Scheduler for High-Density Workloads

Manohar Vanga, Arpan Gujarati, Björn Brandenburg

MPI-SWS, Kaiserslautern, Germany

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

MAX-PLANCK-GESELLSCHAFT

Many small VMs packed onto few cores

How to support

high-density

VM workloads

Why High Density?

Google Cloud Platform

rackspace®

Microsoft Azure

amazon web services

IBM Cloud

Competitive market driving datacenter efficiency

Why High Density?

High-Density VM Packing

Consolidating small, cheap VMs to use fewer resources.

Competitive market driving datacenter efficiency

Why High Density?

High-Density VM Packing

Consolidating small, cheap VMs to use fewer resources.

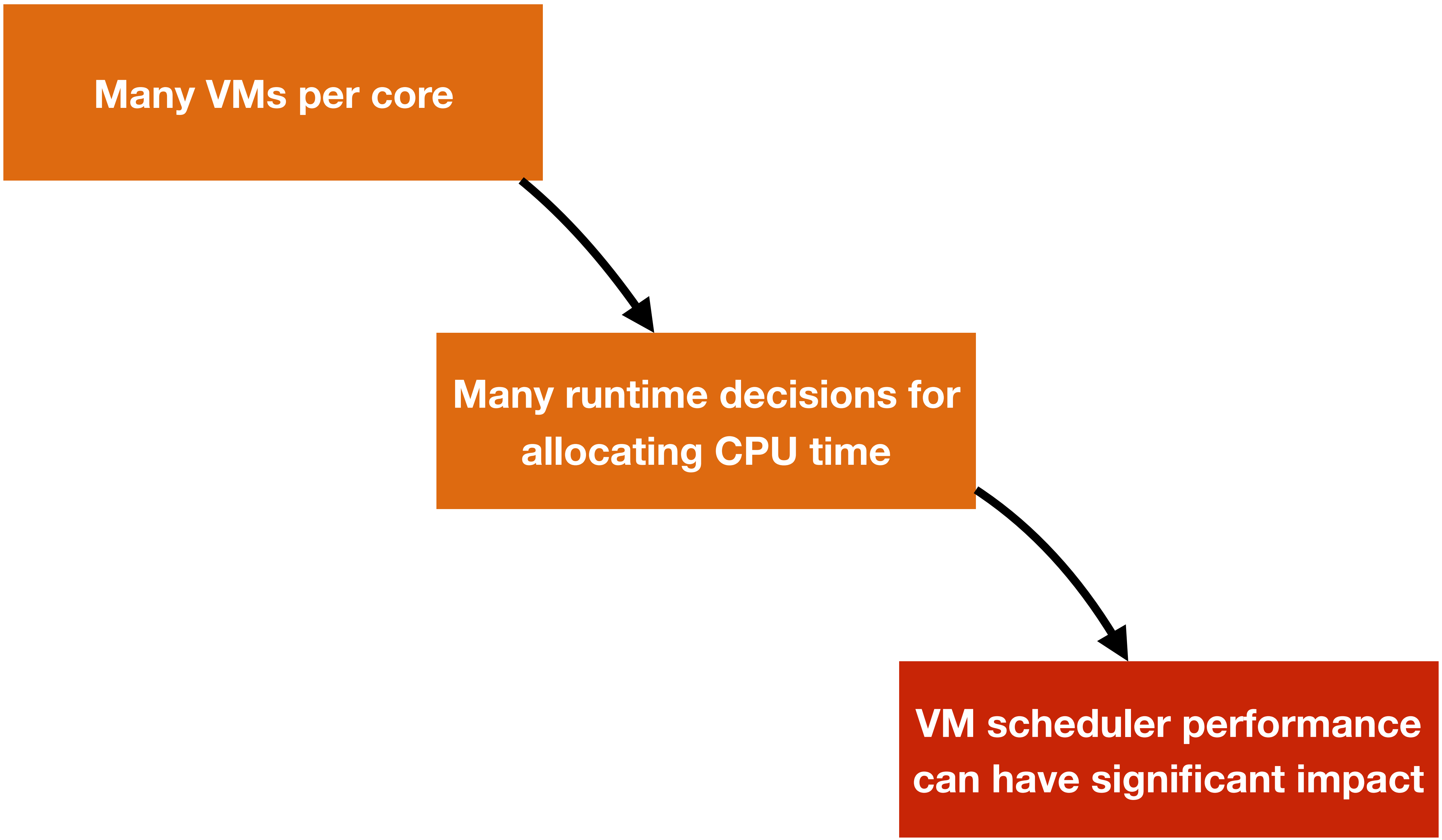
Challenge

Must continue to provide consistent throughput and predictable latency tails.

Competitive market drivin

VM Scheduling Crucial for High-Density

VM Scheduling Crucial for High-Density

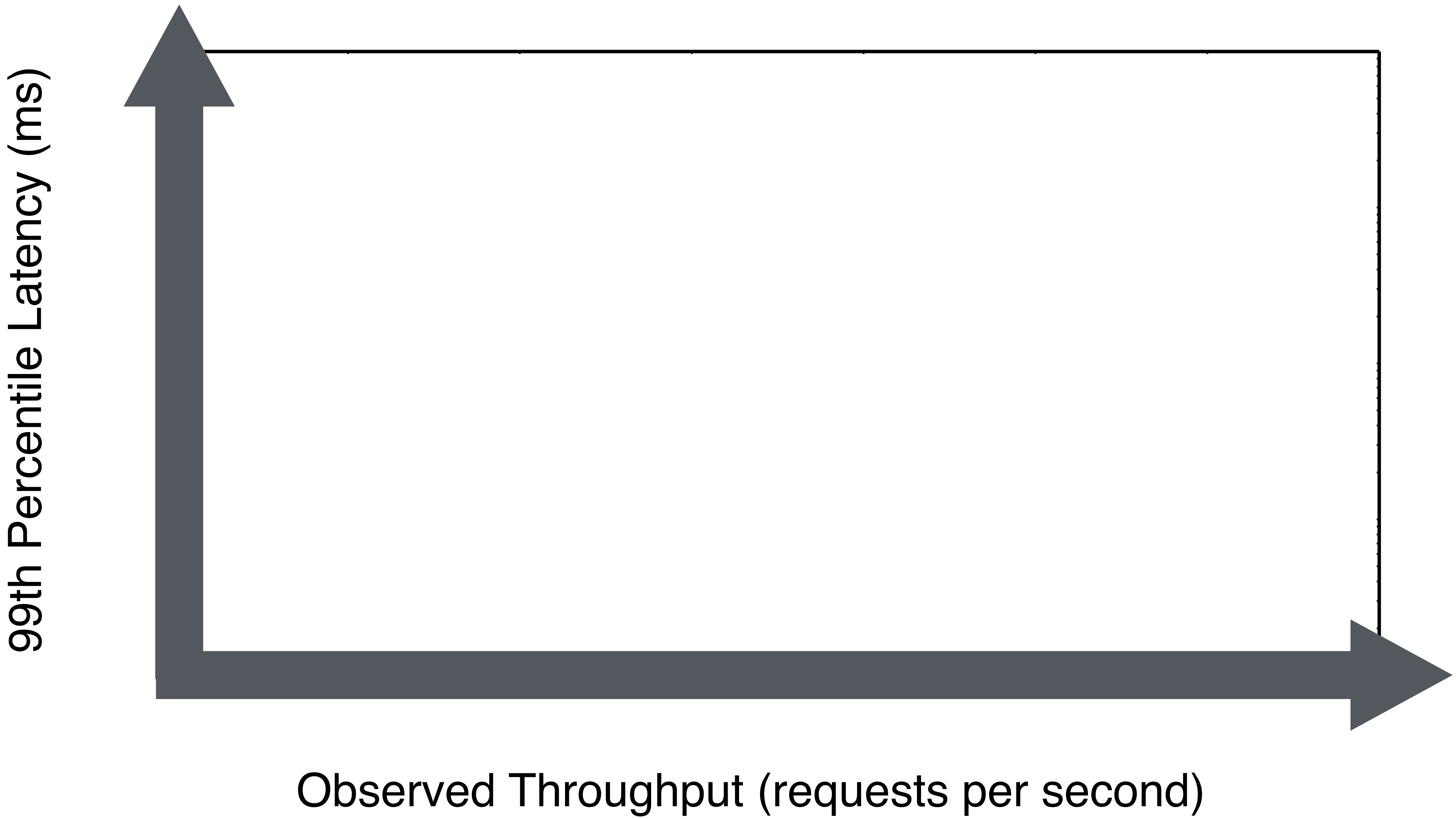


Case Study: VM Scheduling in Xen

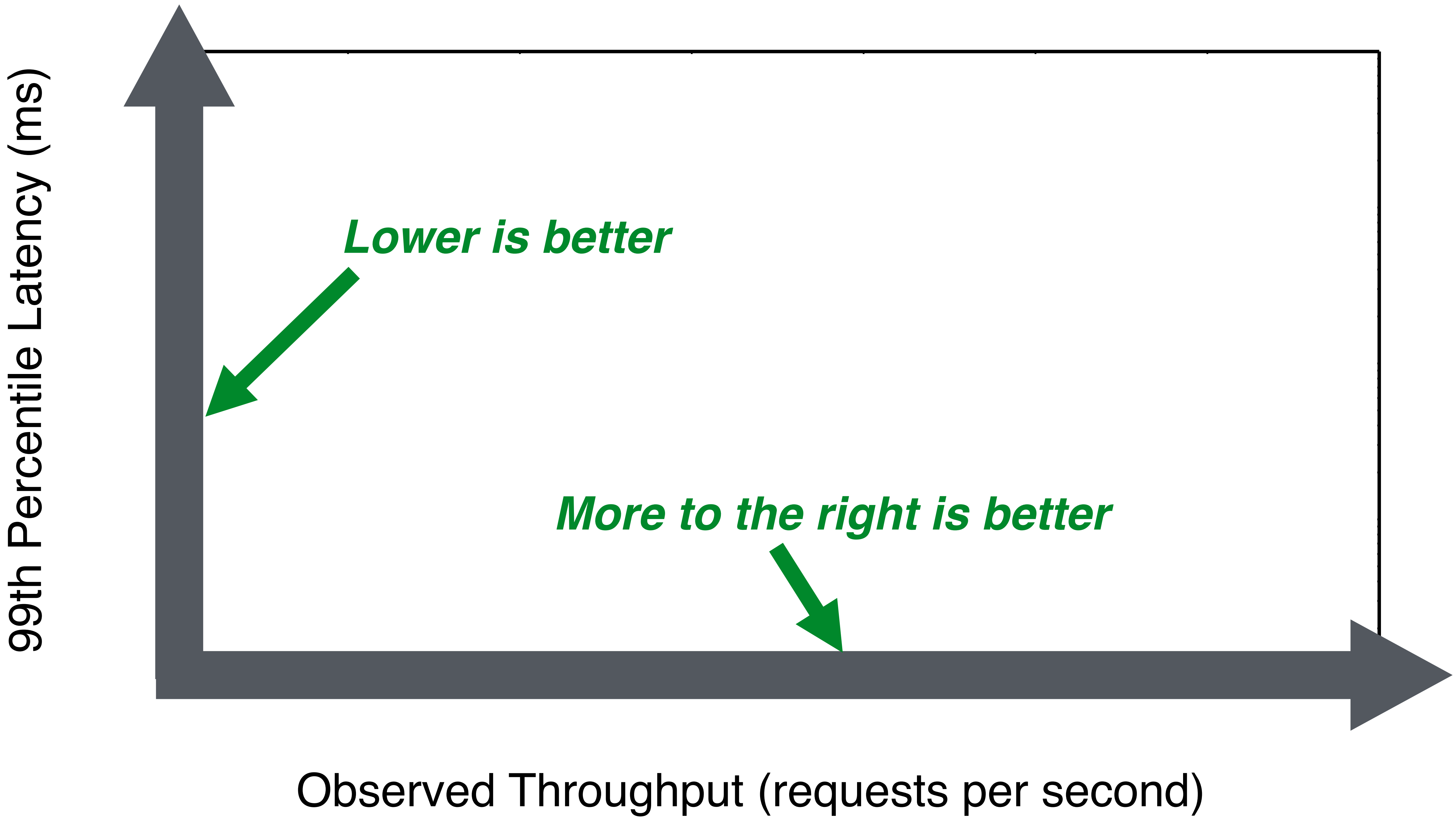
Case Study: VM Scheduling in Xen

- **Four** VMs per core, 16-core server
- Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz.
- Measure **HTTPs performance** of one VM
- All other VMs running **I/O-bound stress workload**.

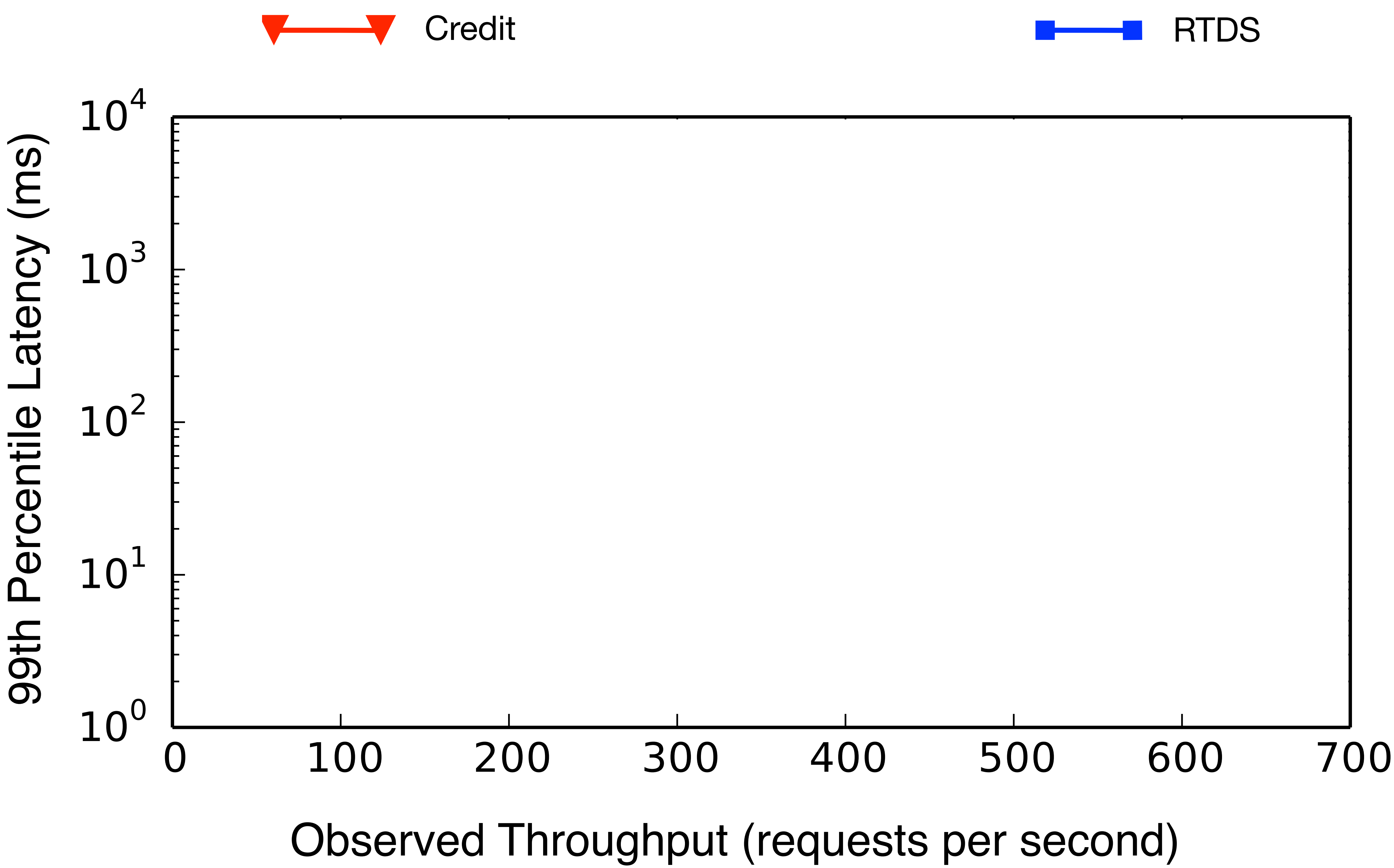
Case Study: VM Scheduling in Xen



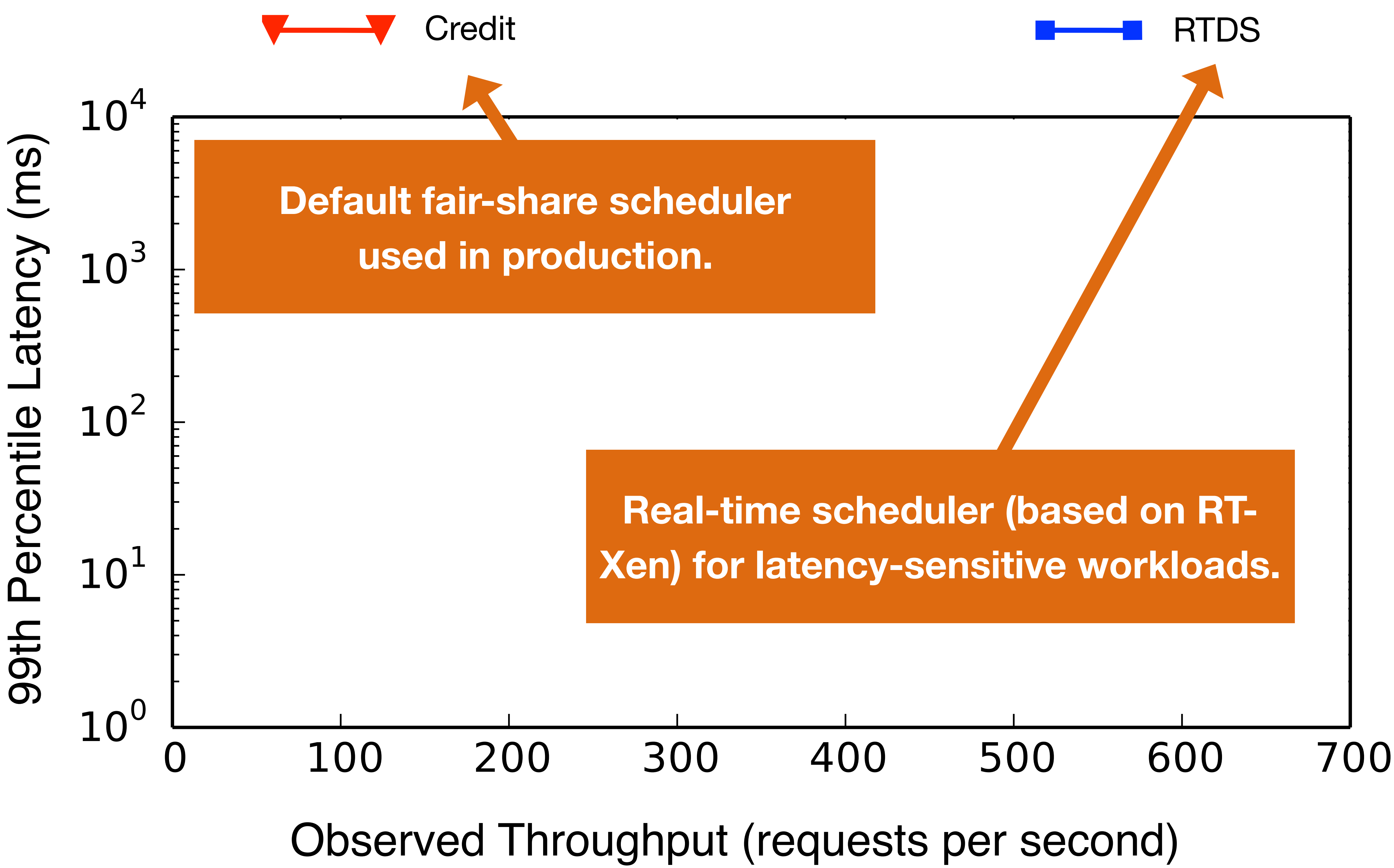
Case Study: VM Scheduling in Xen



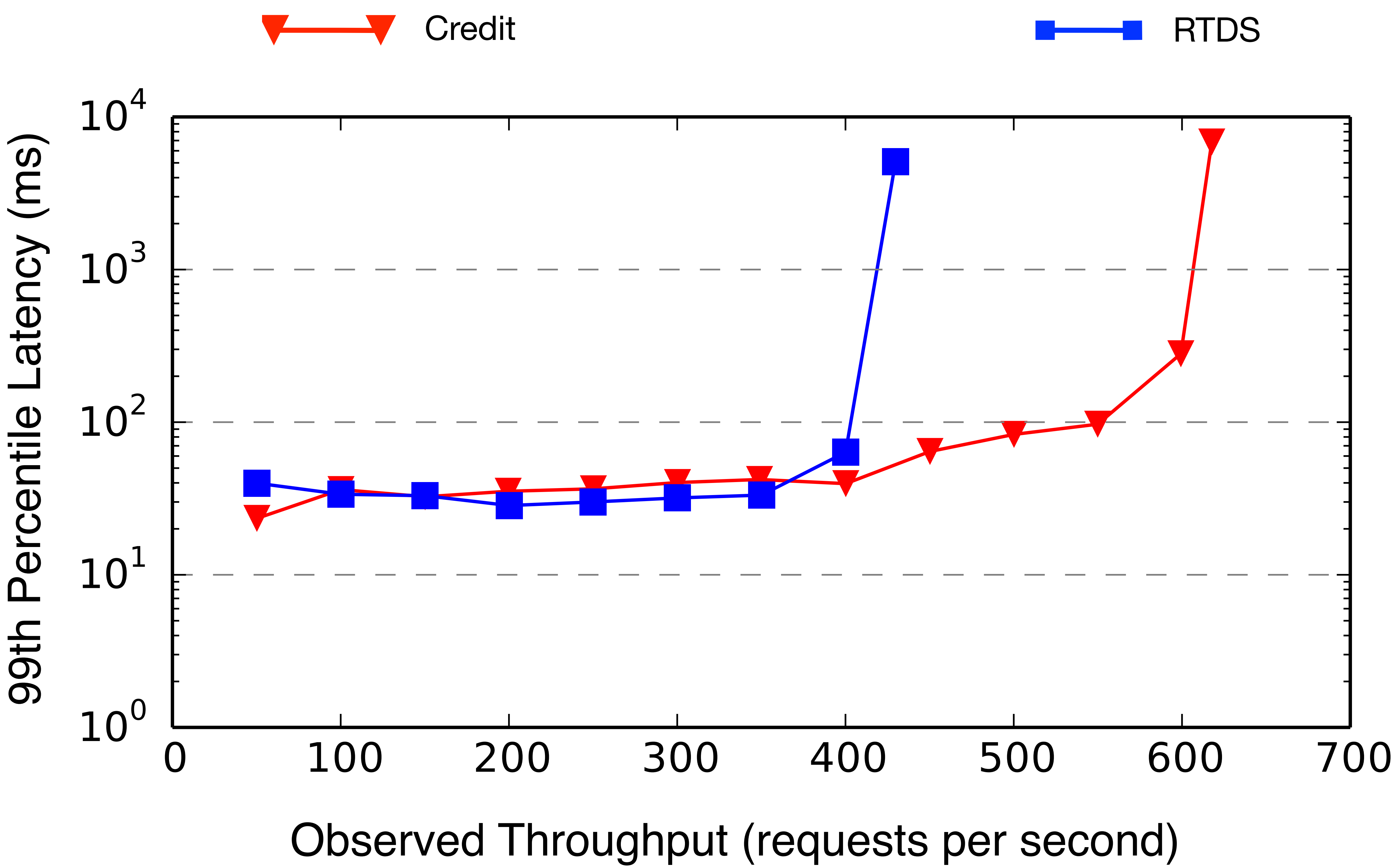
Case Study: VM Scheduling in Xen



Case Study: VM Scheduling in Xen

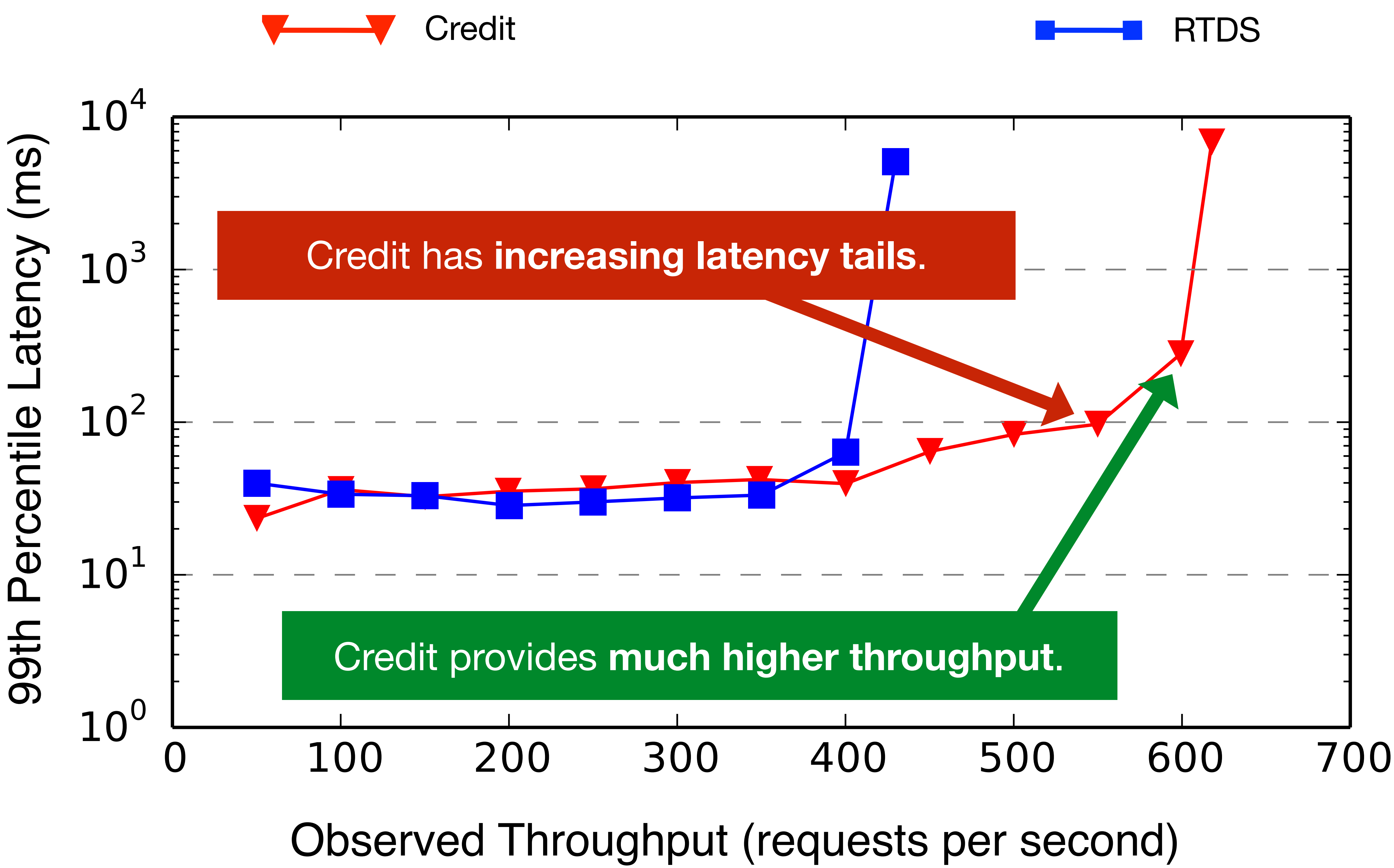


Case Study: VM Scheduling in Xen



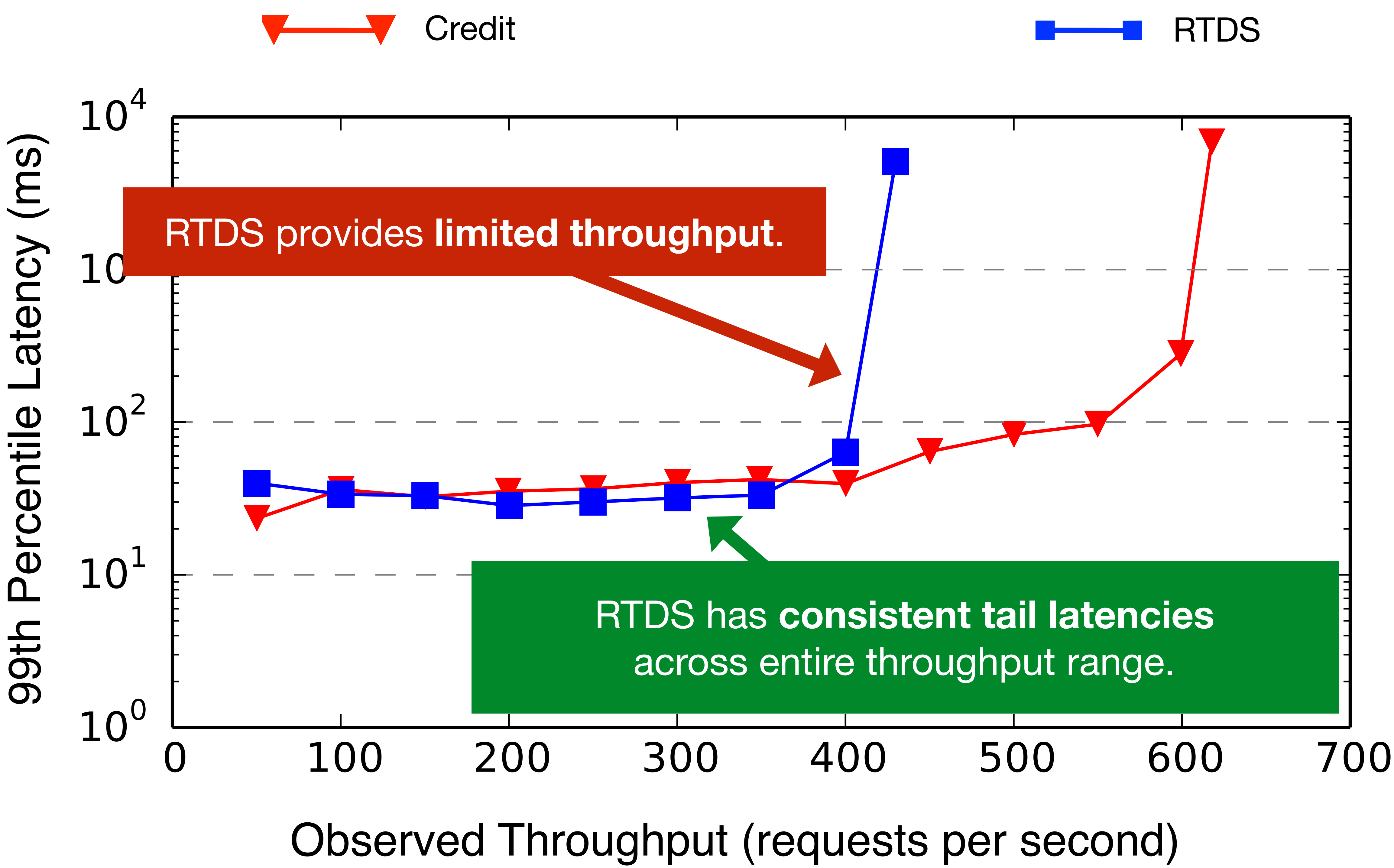
Requesting random 100K-sized files, with I/O background workload

Case Study: VM Scheduling in Xen



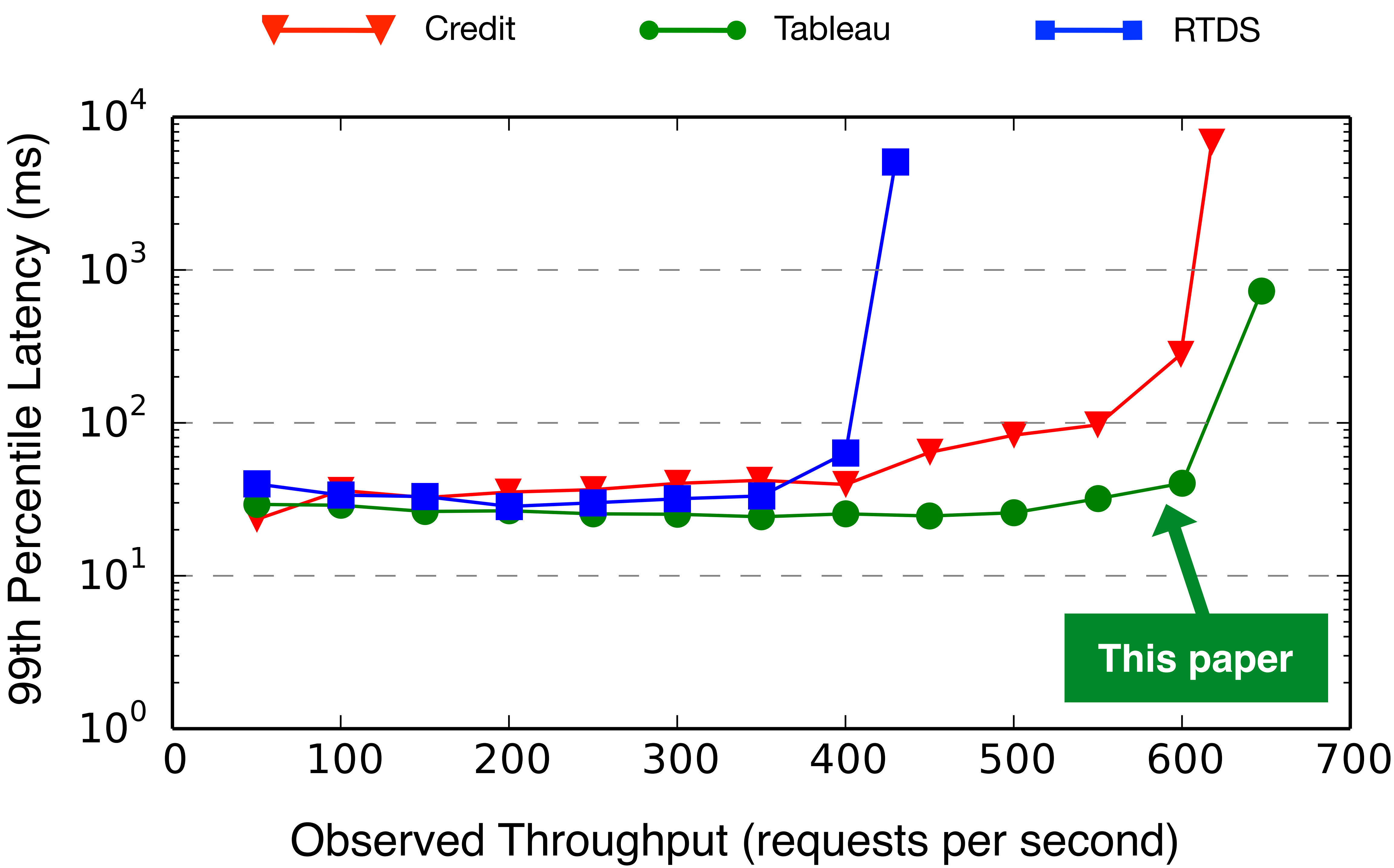
Requesting random 100K-sized files, with I/O background workload

Case Study: VM Scheduling in Xen



Requesting random 100K-sized files, with I/O background workload

The Tableau VM Scheduler



Requesting random 100K-sized files, with I/O background workload

Contributions

Tableau

An **unorthodox scheduling approach** tailored for high-density public clouds.

Contributions

Tableau

An **unorthodox scheduling approach** tailored for high-density public clouds.

Efficient

Incurs low overheads

Predictable

Accurate control over scheduling latency.

High-throughput

Provides high SLA-aware throughput.

This Talk

- Tableau
- Evaluation
- Conclusion

This Talk

- **Tableau**
- Evaluation
- Conclusion

What Do We Want From a VM Scheduler?

- **Requirement 1**: Be as "invisible" as possible.
- **Requirement 2**: Guarantee utilization and ensure predictable scheduling latency for every VM.

What Do We Want From a VM Scheduler?

Low overheads

- **Requirement 1**: Be as "invisible" as possible.
- **Requirement 2**: Guarantee utilization and ensure predictable scheduling latency for every VM.

What Do We Want From a VM Scheduler?

- **Requirement 1**: Be as "invisible" as possible.
- **Requirement 2**: Guarantee utilization and ensure predictable scheduling latency for every VM.

Requirement 2 is a **non-trivial** problem!

What Do We Want From a VM Scheduler?

- **Requirement 1**: Be as "invisible" as possible.
- **Requirement 2**: Guarantee utilization and ensure predictable scheduling latency for every VM.

Requirement 2 is a **non-trivial** problem!

Attempting to enforce requirement 2 **at runtime** conflicts with requirement 1.

What Do We Want From a VM Scheduler?

- **Requirement 1**: Be as "invisible" as possible.
- **Requirement 2**: Guarantee utilization and ensure predictable scheduling latency for every VM.

Requirement 2 is a **non-trivial** problem!

Attempting to enforce requirement 2 **at runtime** conflicts with requirement 1.

How do we overcome these conflicting requirements?

The Tableau Approach

Exploit one key property of VM environments

VM churn on a single server is low [1]

[1] Cortez et al., *Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms*, SOSP 2017

The Tableau Approach

Requirement 1
As invisible as possible.
Fast, Low overhead

Requirement 2
Guarantee utilization and
scheduling latency

The Tableau Approach

Requirement 1
As invisible as possible.
Fast, Low overhead

Requirement 2
Guarantee utilization and
scheduling latency

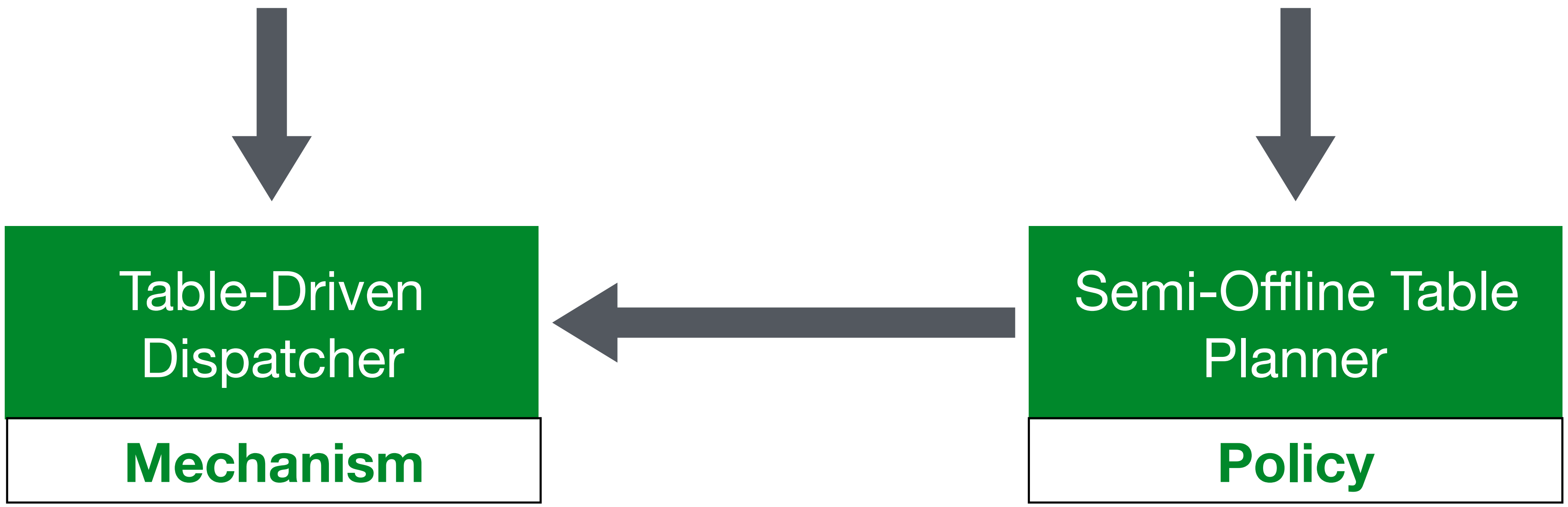
Table-Driven
Dispatcher

The Tableau Approach

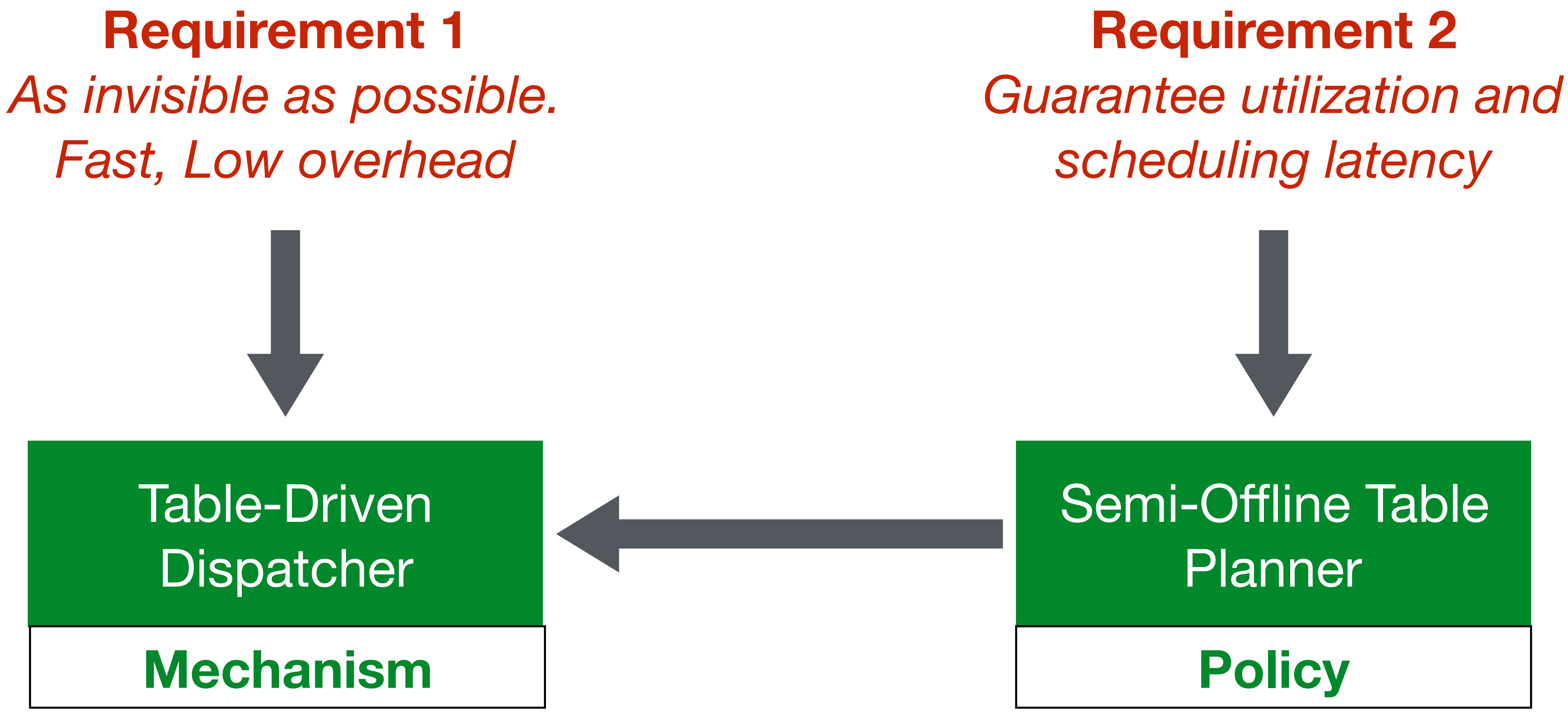
The Tableau Approach

Requirement 1
As invisible as possible. Fast, Low overhead

Requirement 2
Guarantee utilization and scheduling latency

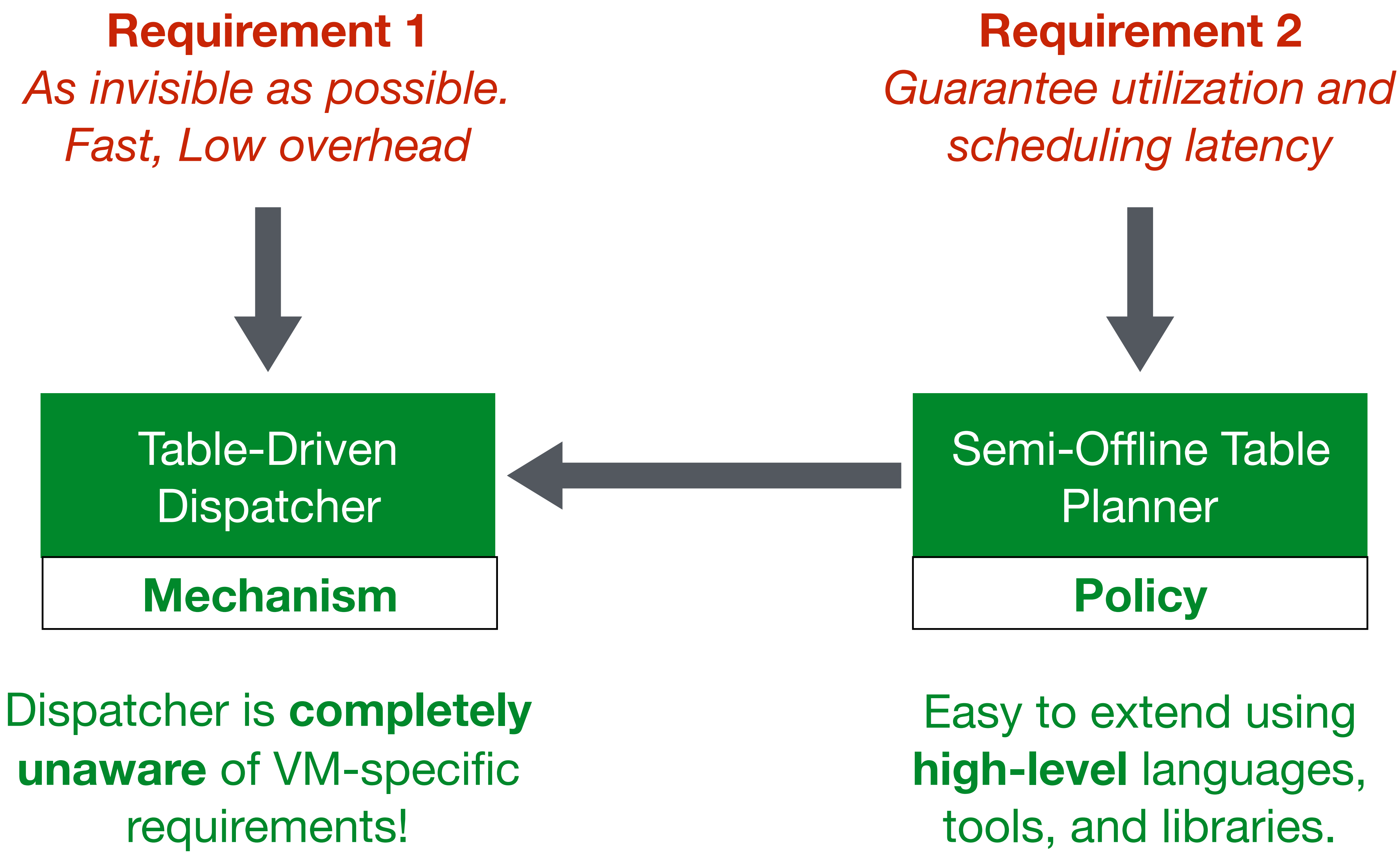


The Tableau Approach



Dispatcher is **completely unaware** of VM-specific requirements!

The Tableau Approach



The Tableau Approach

Requirement 1
As invisible as possible.
Fast, Low overhead

Requirement 2
Guarantee utilization and scheduling latency

Table-Driven Dispatcher
Mechanism

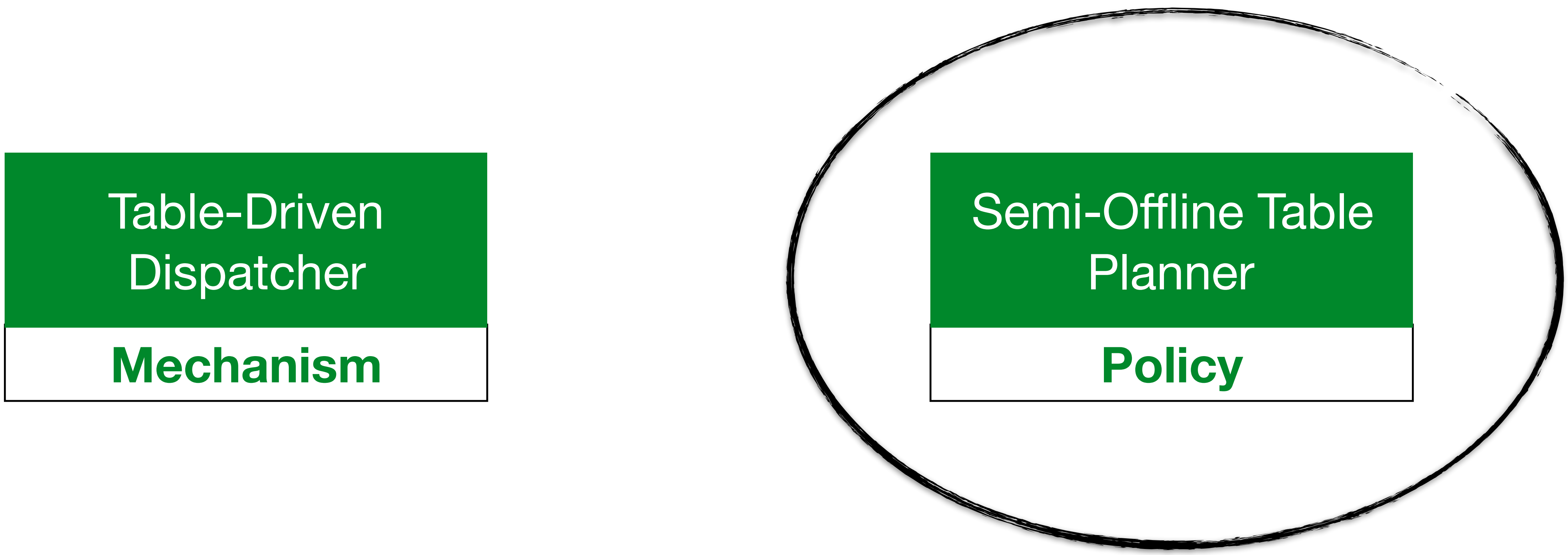
Semi-Offline Table Planner
Policy

Dispatcher is **completely unaware** of VM-specific requirements!

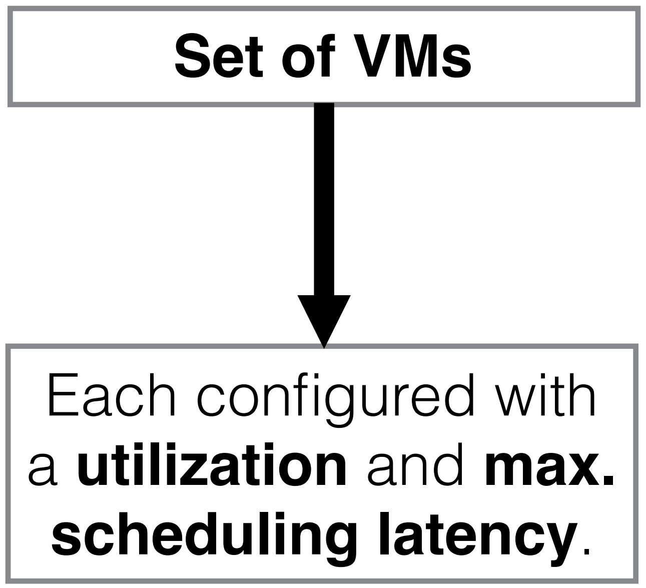
Easy to extend using **high-level** languages, tools, and libraries.

Can be **pre-generated** or generated on a **separate machine**.

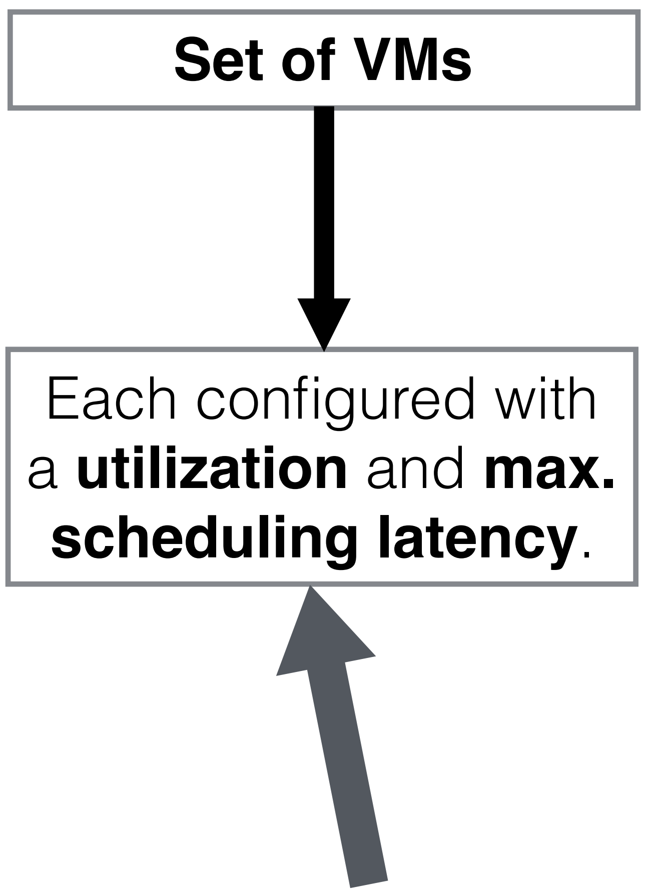
The Tableau Approach



Generating Tables Quickly

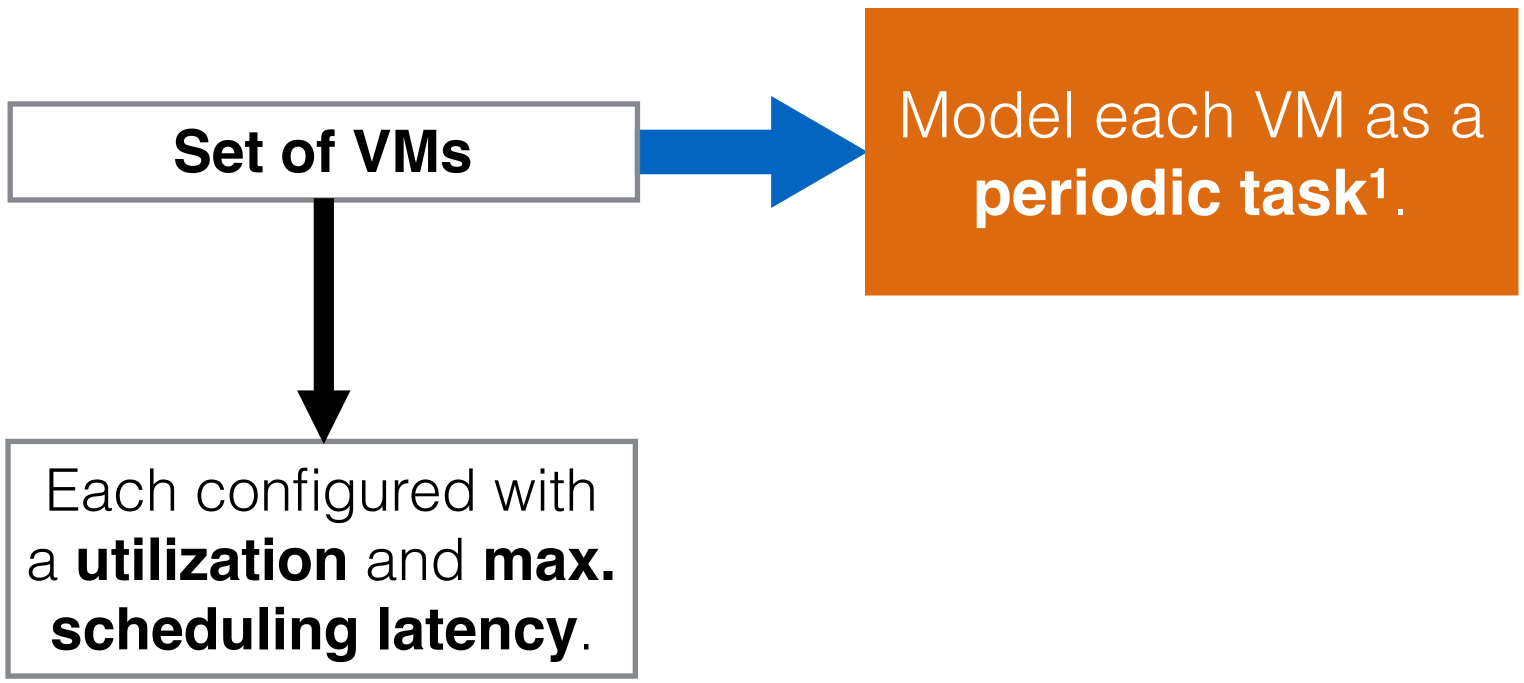


Generating Tables Quickly



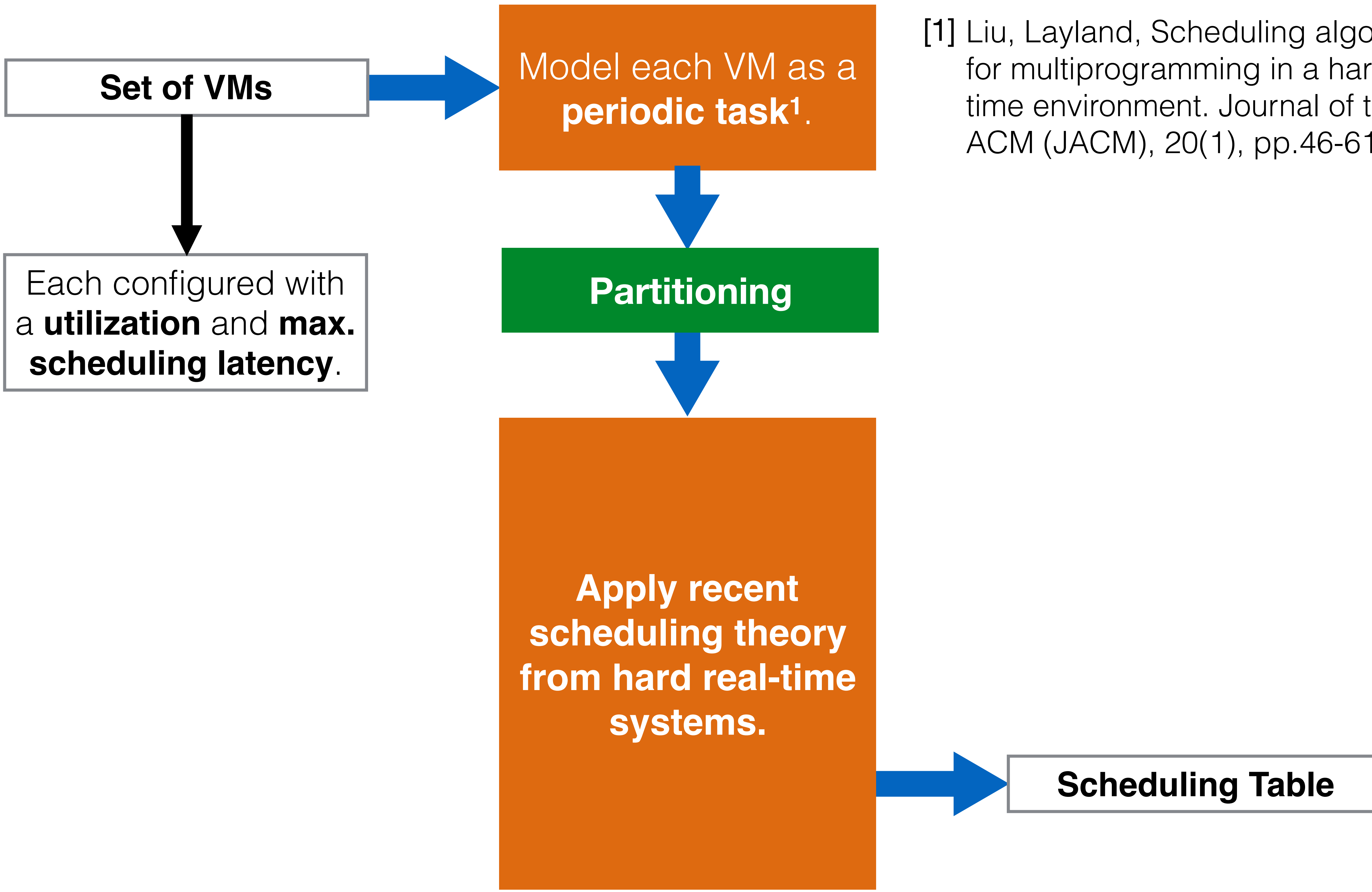
No more information than existing schedulers *(e.g., Credit requires a relative weight and timeslice)*

Generating Tables Quickly



[1] Liu, Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM), 20(1), pp.46-61, 1973

Generating Tables Quickly



[1] Liu, Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM), 20(1), pp.46-61, 1973

Generating Tables Quickly

Set of VMs

Model each VM as a **periodic task**[1].

Each configu
a **utilization** a
scheduling l

Performed **entirely in userspace** of supervisory VM.

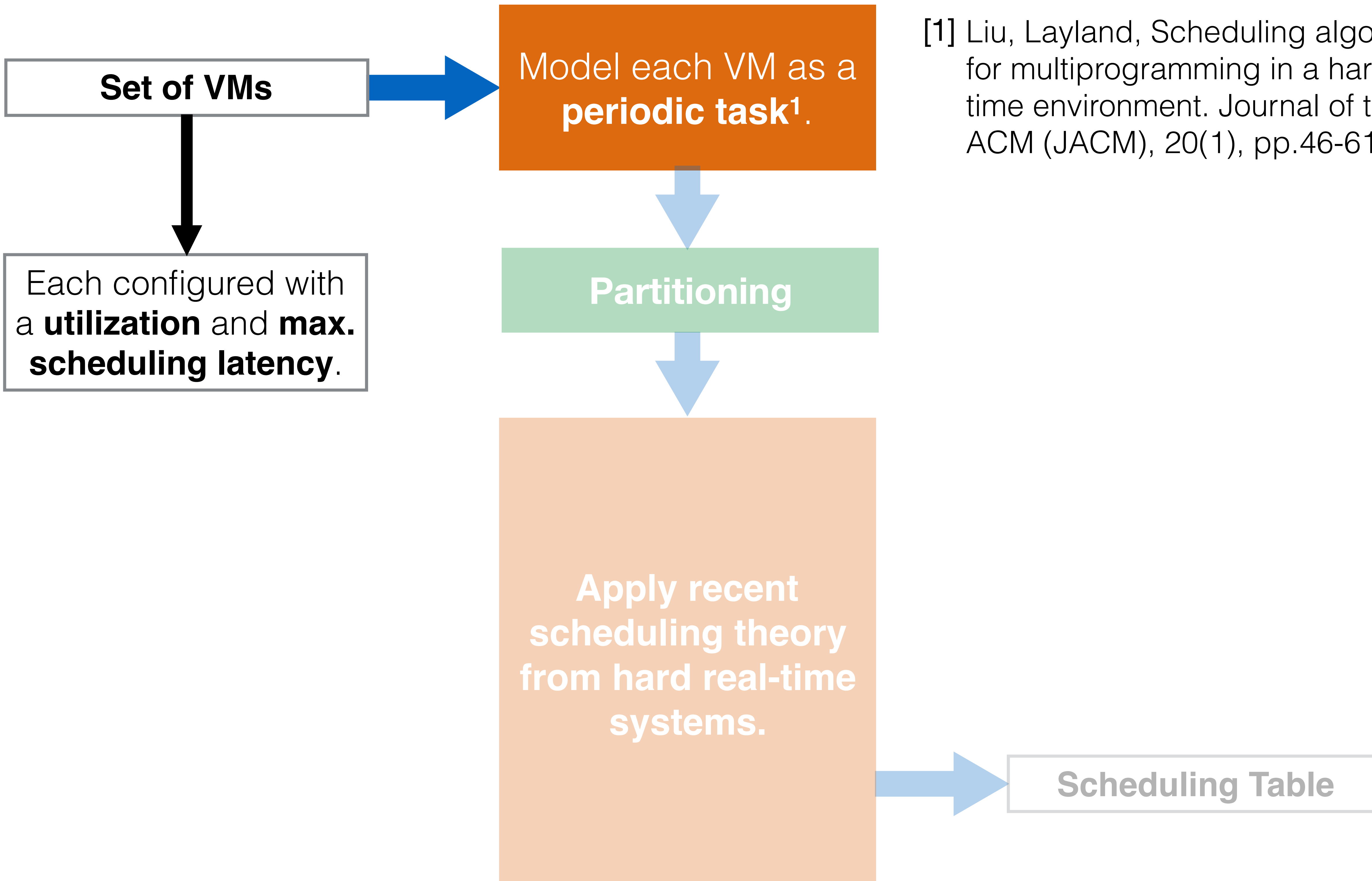
Implemented **in Python using a mature library (SchedCAT)**.

Apply recent scheduling theory from hard real-time systems.

Scheduling Table

[1] Liu, Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM), 20(1), pp.46-61.

Generating Tables Quickly



[1] Liu, Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM), 20(1), pp.46-61.

Modeling VMs as Periodic Tasks

VM (vCPU)

Utilization (U)
A percentage of CPU time reserved for VM.

Max Sched. Delay (L)
An upper bound on scheduling delay.

Periodic Task

Budget (C)

Period (T)

Modeling VMs as Periodic Tasks

VM (vCPU)

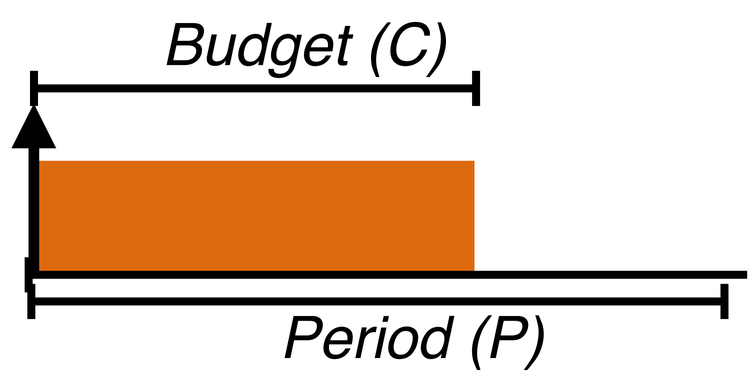
Utilization (U)
A percentage of CPU time reserved for VM.

Max Sched. Delay (L)
An upper bound on scheduling delay.

Periodic Task

Budget (C)

Period (P)



Modeling VMs as Periodic Tasks

VM (vCPU)

Utilization (U)
A percentage of CPU time reserved for VM.

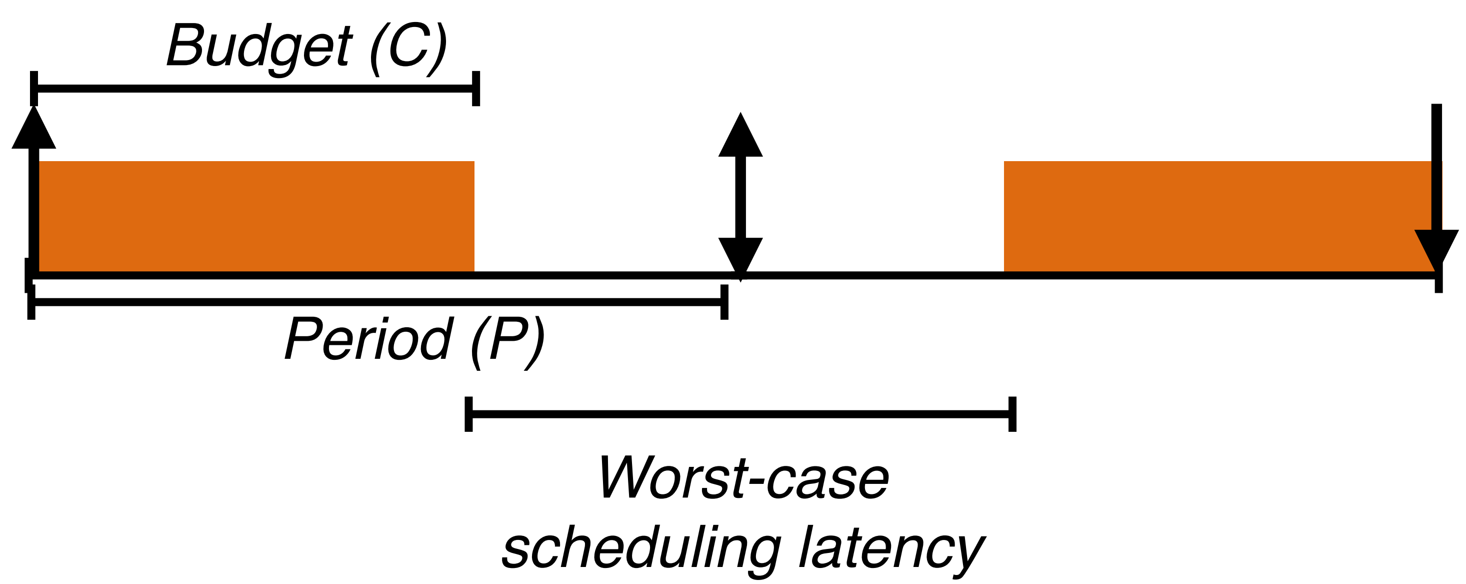
Max Sched. Delay (L)
An upper bound on scheduling delay.

↓

Periodic Task

Budget (C)

Period (P)



Modeling VMs as Periodic Tasks

VM (vCPU)

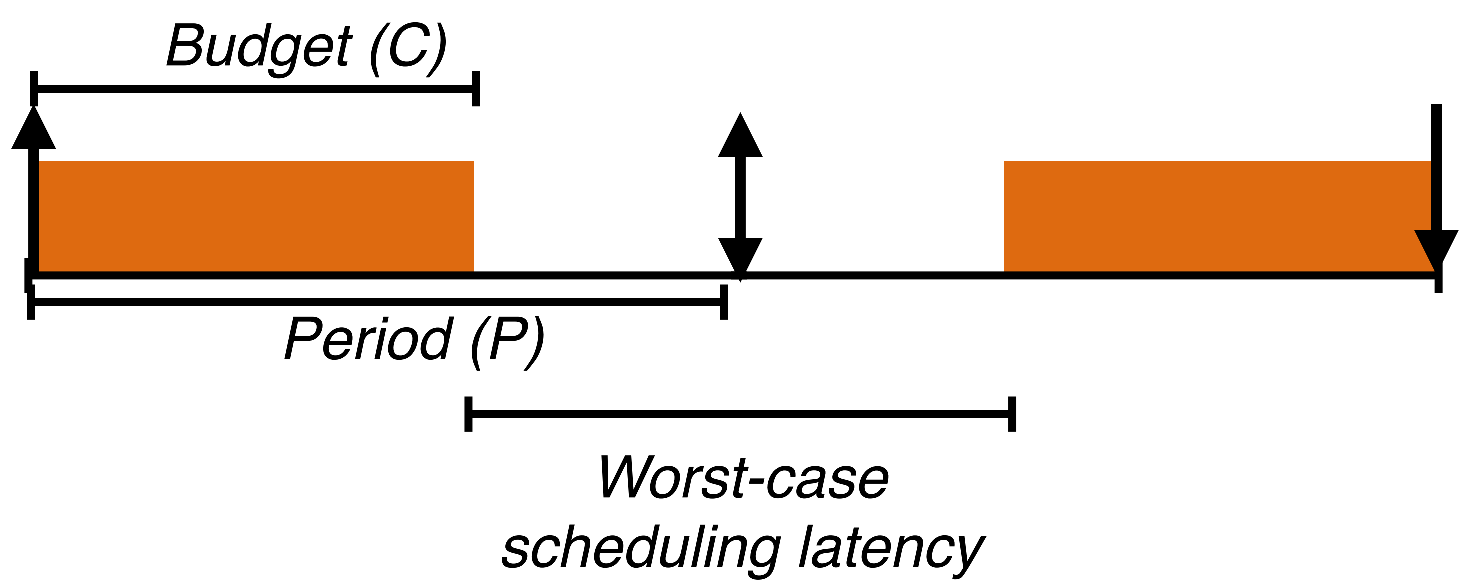
Utilization (U)
A percentage of CPU time reserved for VM.

Max Sched. Delay (L)
An upper bound on scheduling delay.

Periodic Task

Budget (C)

Period (P)



2 x (P - C)

Generating Tables Quickly

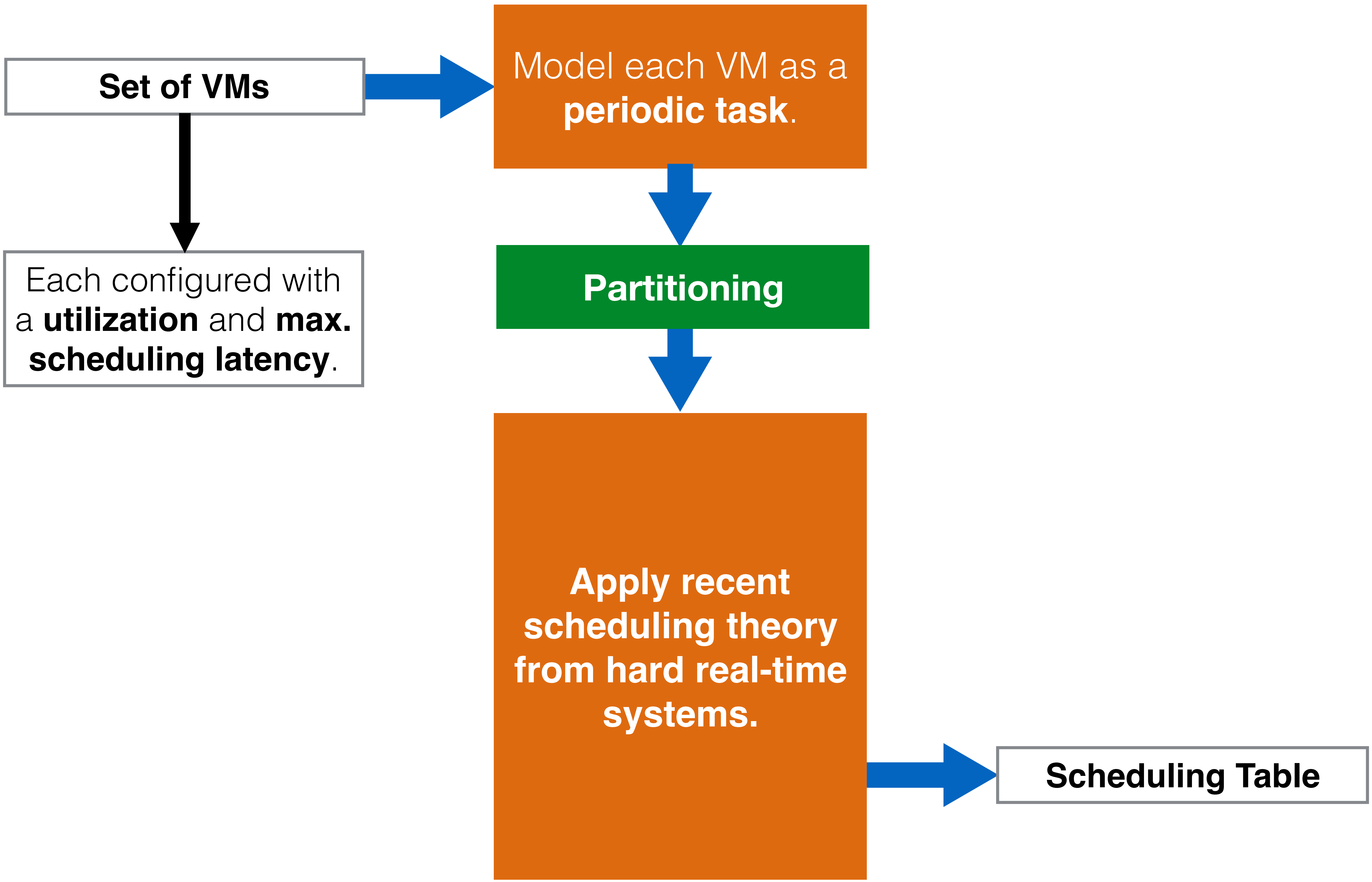


Table-Generation Times

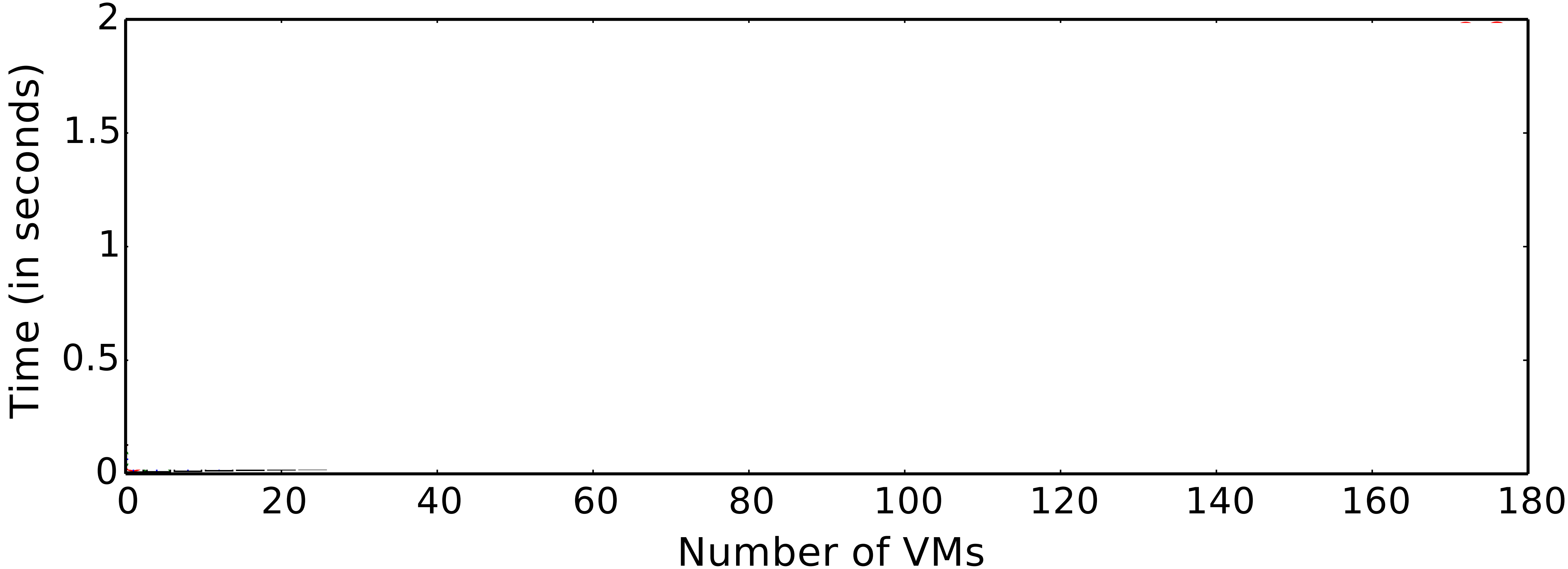


Table-Generation Times

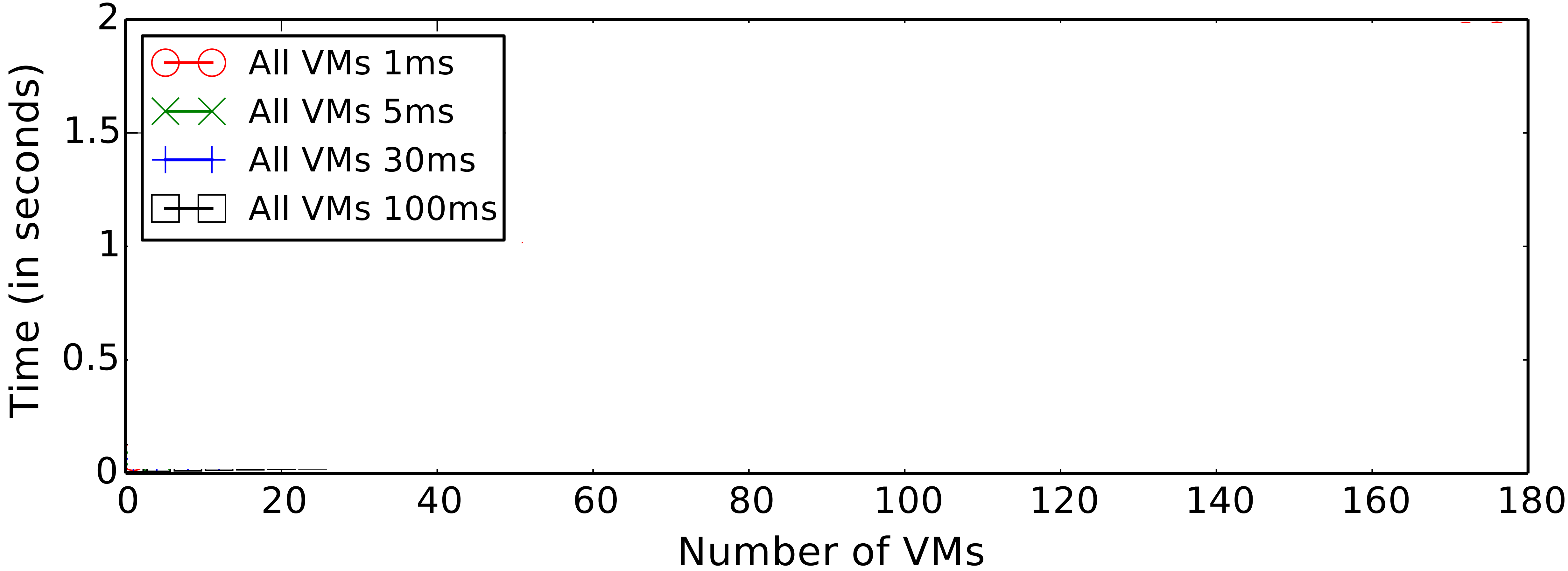


Table-Generation Times

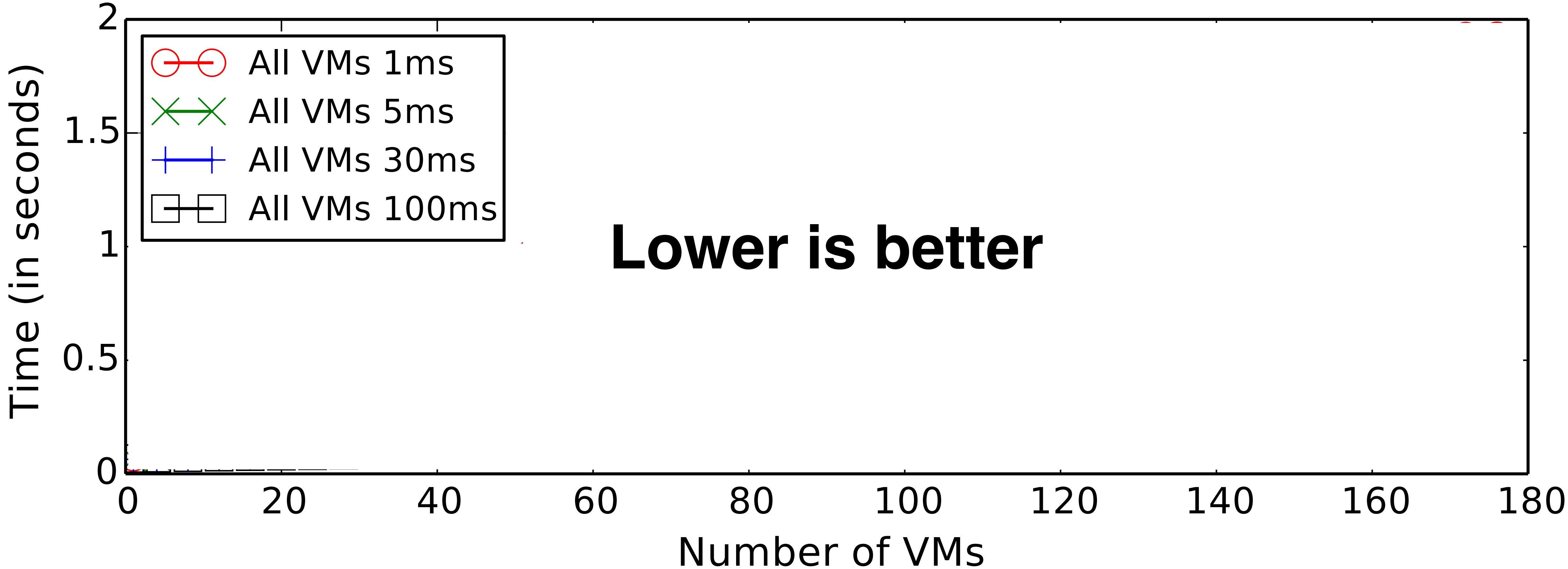


Table-Generation Times

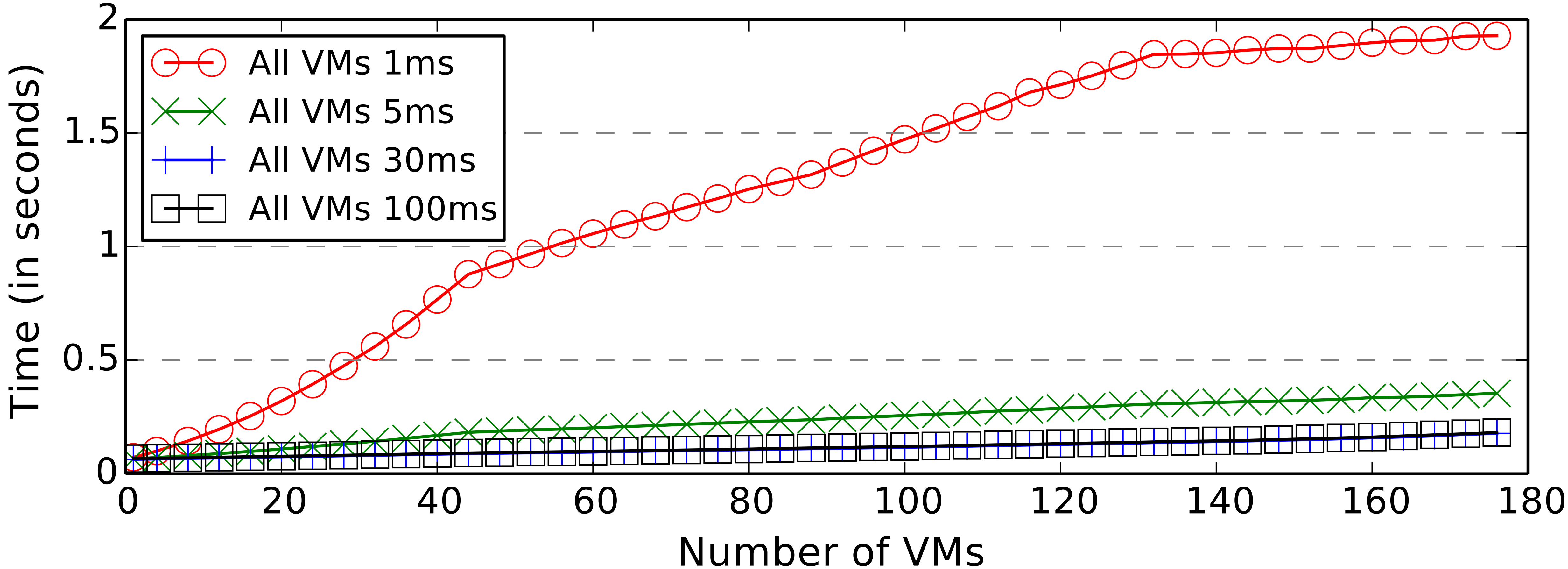


Table-Generation Times

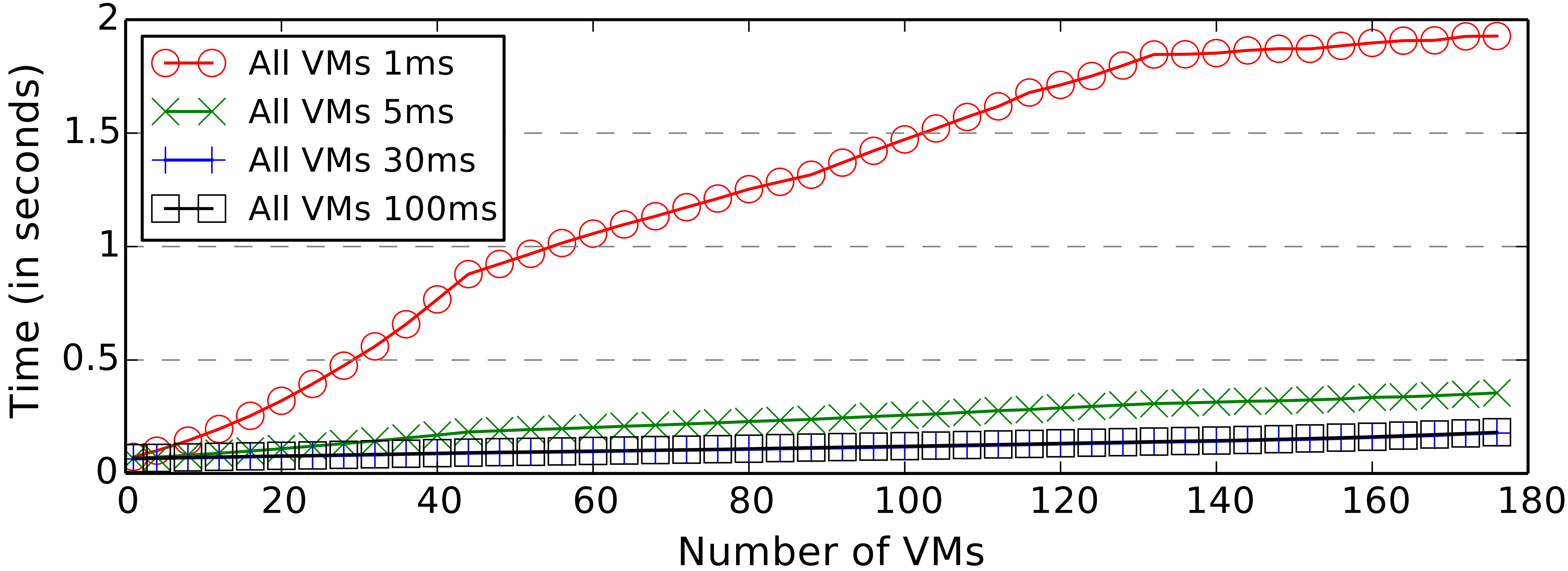
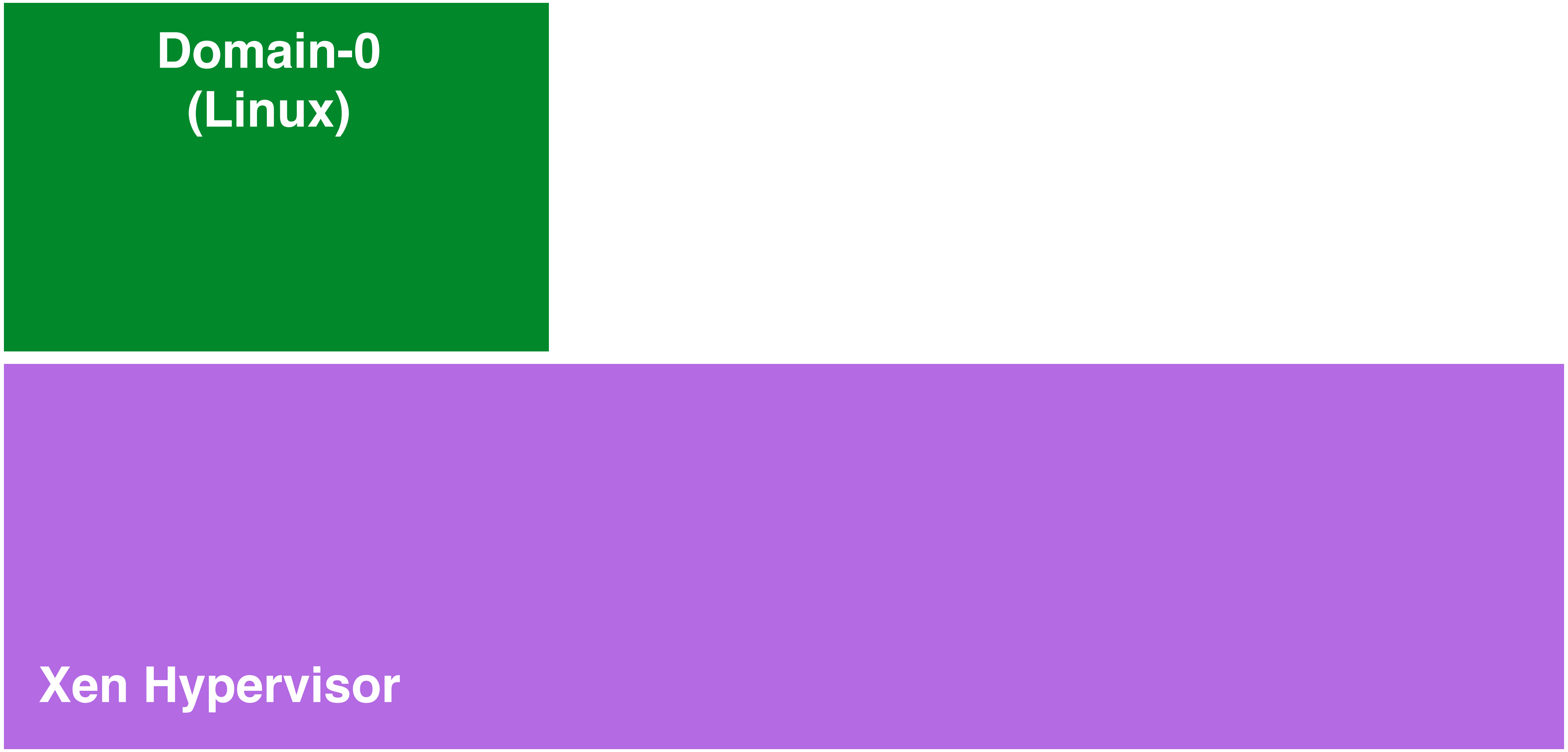


Table generation times are reasonable compared to VM creation and teardown times.

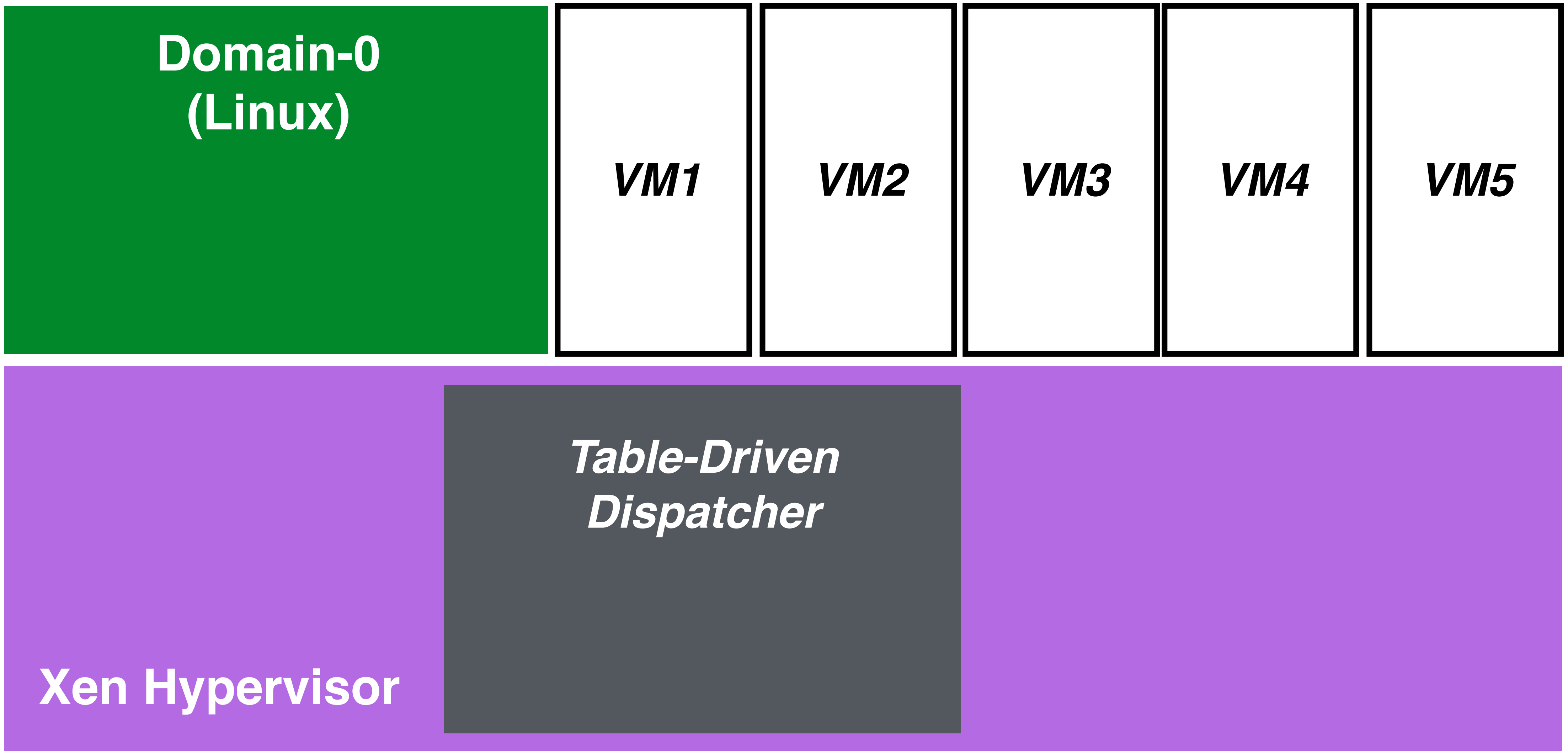
The Tableau Approach

Implementation in Xen



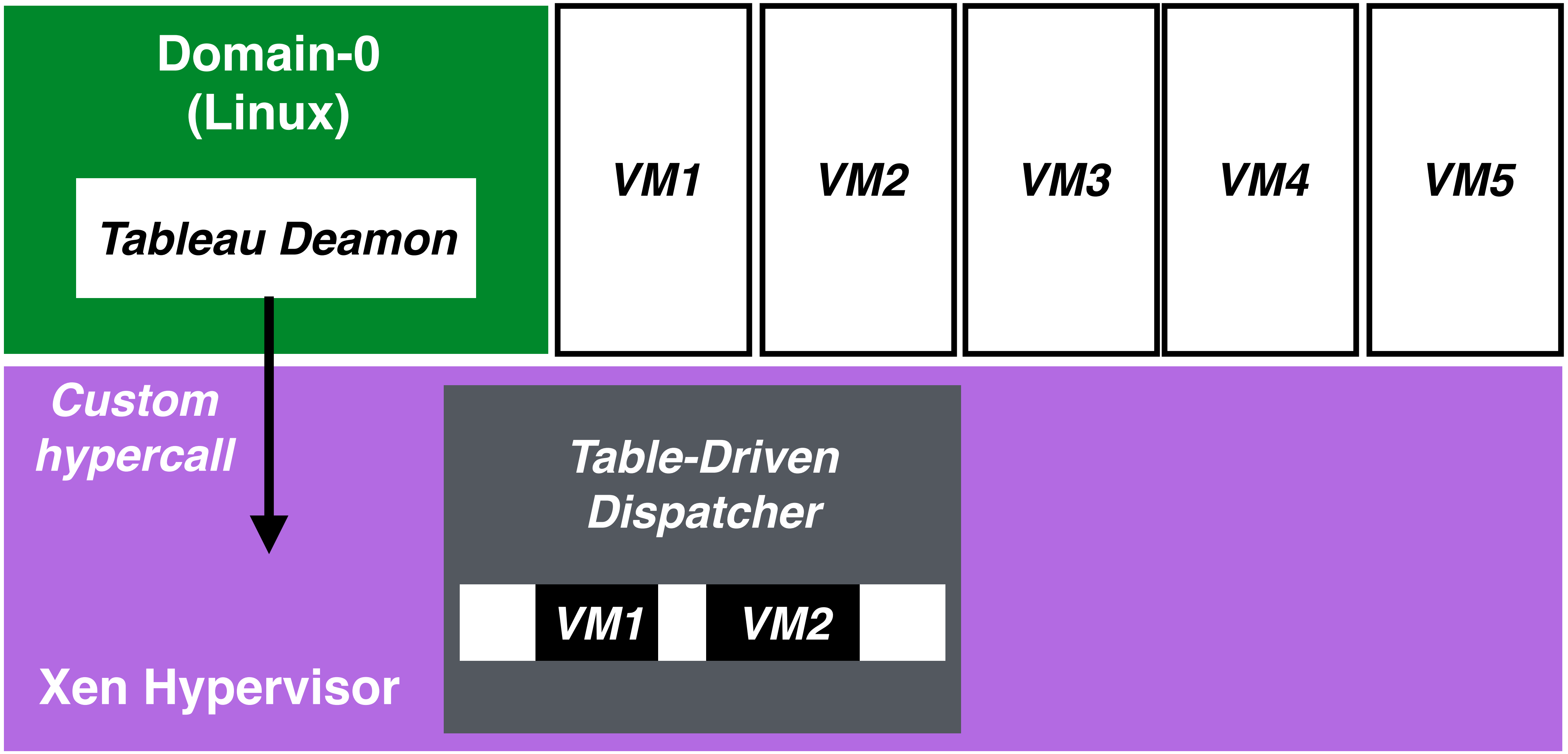
- Popular open-source hypervisor (Amazon AWS)
- Supervisory VM (domain-0) created at boot time.

Implementation in Xen



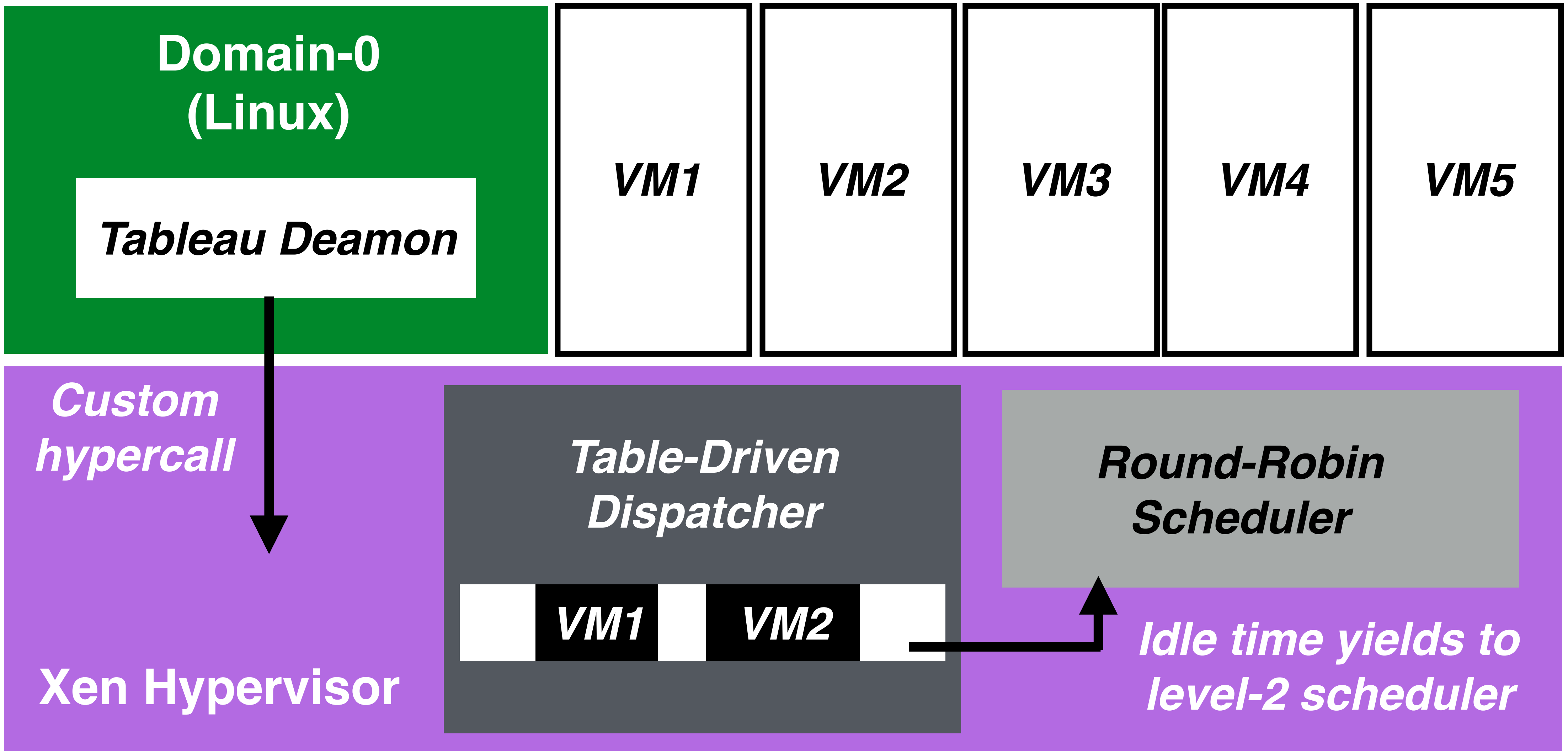
- Simple, table-driven dispatcher implemented within the hypervisor.

Implementation in Xen



- Userspace daemon responsible for re-generating tables whenever a VM is created.
- ~1,600 lines of Python code.

Implementation in Xen



- For work-conserving behavior, idle time in tables (white blocks) yields to round-robin scheduler. Picks runnable core-local VMs to schedule.

This Talk

- Tableau
- **Evaluation**
- Conclusion

Summary of Results

Tableau **incurs lower runtime overheads** compared to the other evaluated Xen schedulers

Summary of Results

Tableau **incurs lower runtime overheads** compared to the other evaluated Xen schedulers

Tableau enables **accurate control over scheduling latency**.

Summary of Results

Tableau **incurs lower runtime overheads** compared to the other evaluated Xen schedulers

Tableau enables **accurate control over scheduling latency**.

Tableau achieves **higher SLA-aware application throughput**.

Summary of Results

Tableau **incurs lower runtime overheads** compared to the other evaluated Xen schedulers

See our paper for details!

Tableau enables **accurate control over scheduling latency**.

Tableau achieves **higher SLA-aware application throughput.**

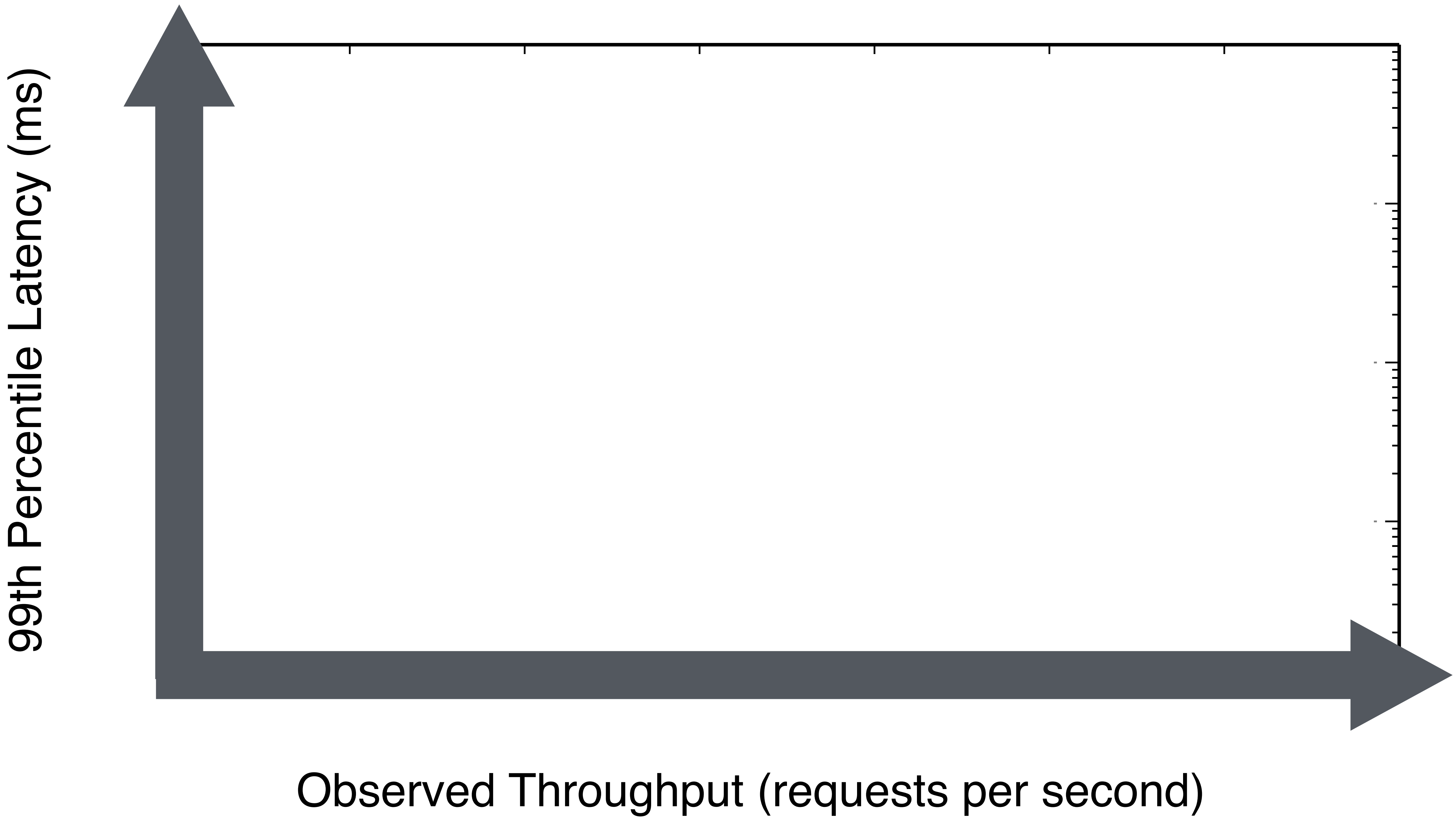
Platform

- Server machine:
  - 16 cores (2 sockets), 512 GiB RAM
  - Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz
  - Ubuntu 16.04.3
  - Xen 4.9
- Load generation machine:
  - Identical machine connected via 10G ethernet.

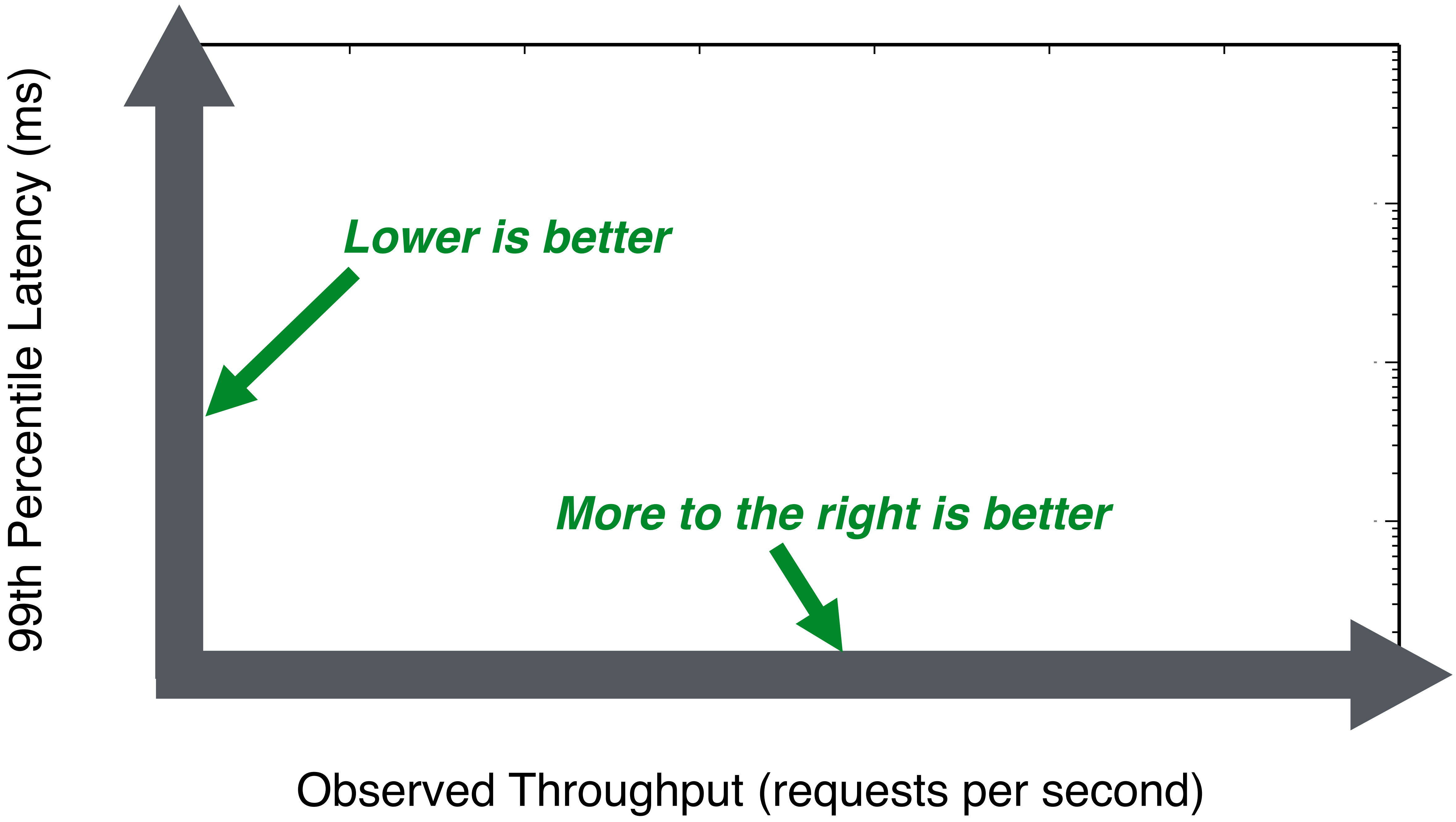
Experimental Setup

- We simulate a multi-tenant datacenter environment.
  - 4 VMs/core (25% utilization each).
  - 1 **vantage** VM, rest **background** VMs
  - Background VMs run different workloads based on **stress-ng** tool.
- Schedulers configured based on best practices:
  - 5ms timeslice in Credit.
  - Equivalent configuration in Tableau and RTDS (max 20ms scheduling latency)

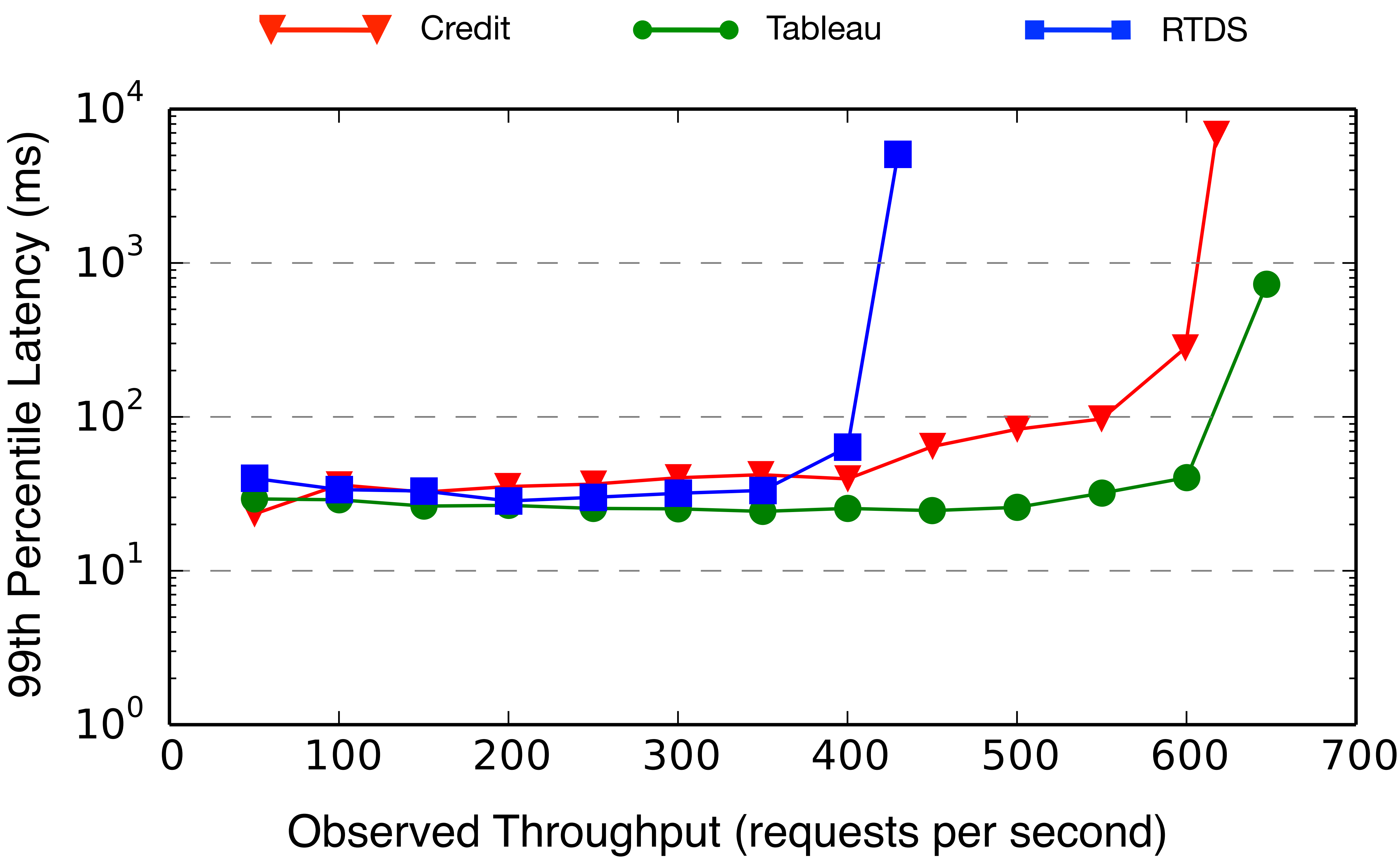
SLA-Aware Throughput



SLA-Aware Throughput

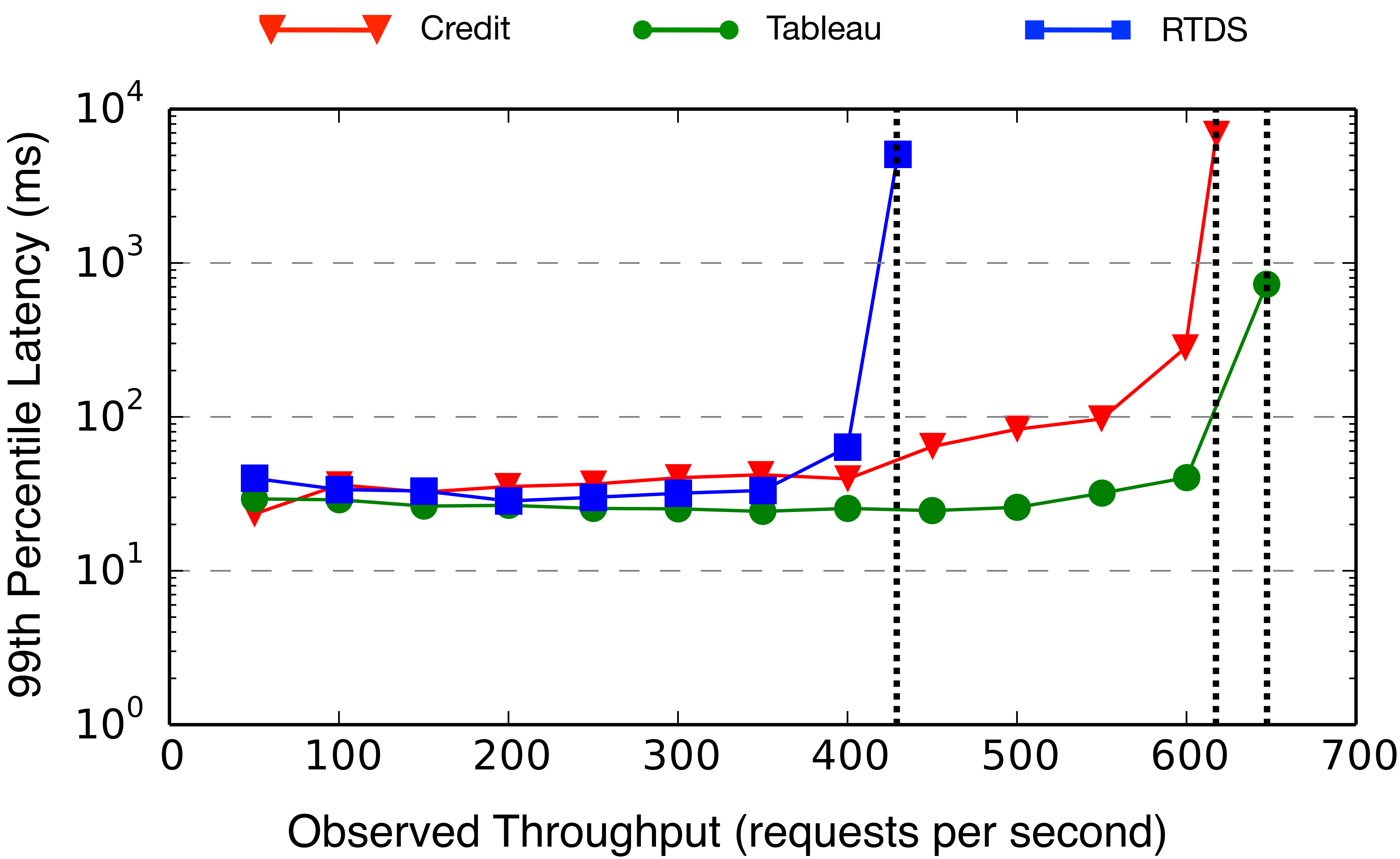


Peak Throughput



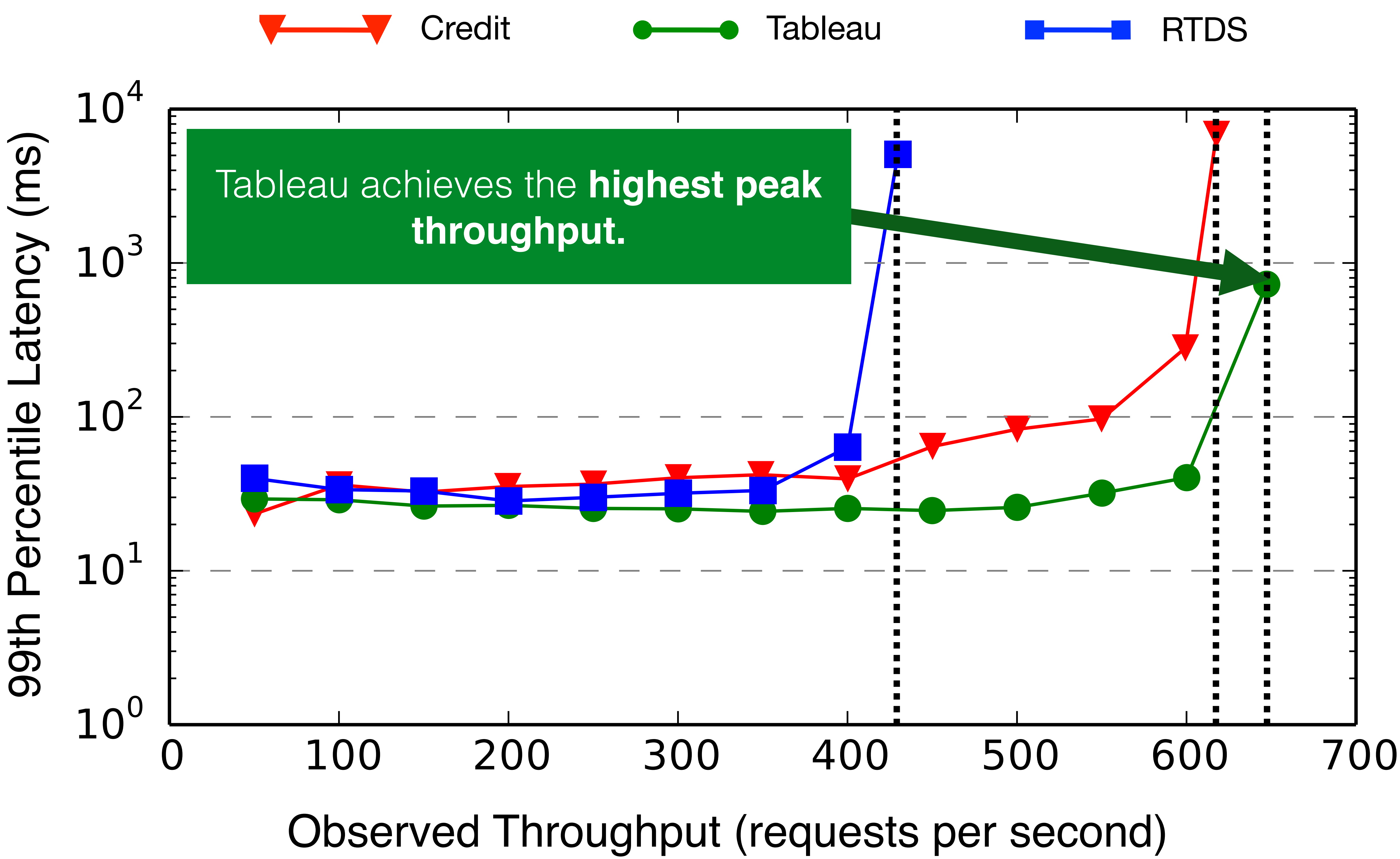
VMs Capped at 25%, 100K files, I/O background workload

Peak Throughput



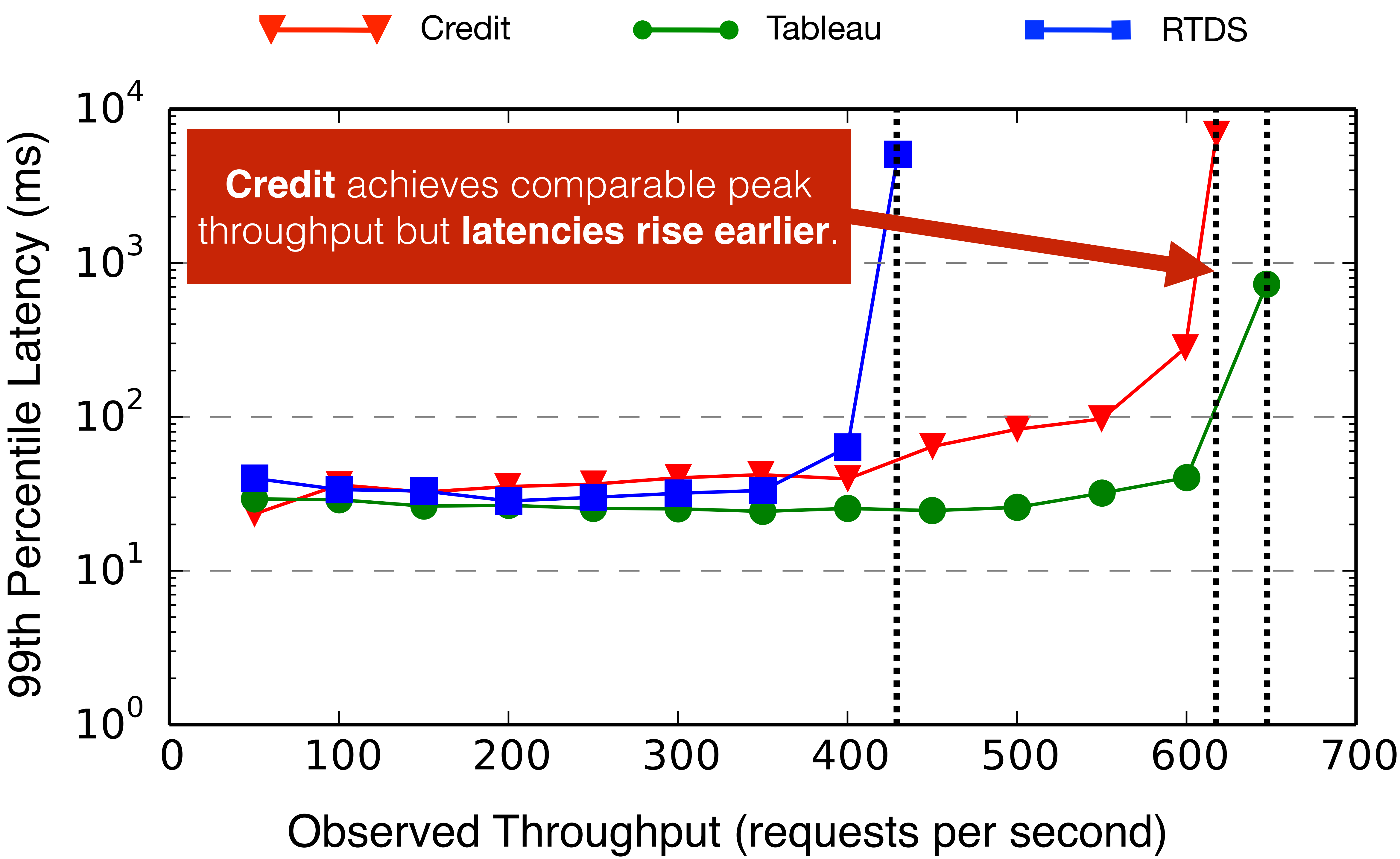
VMs Capped at 25%, 100K files, I/O background workload

Peak Throughput



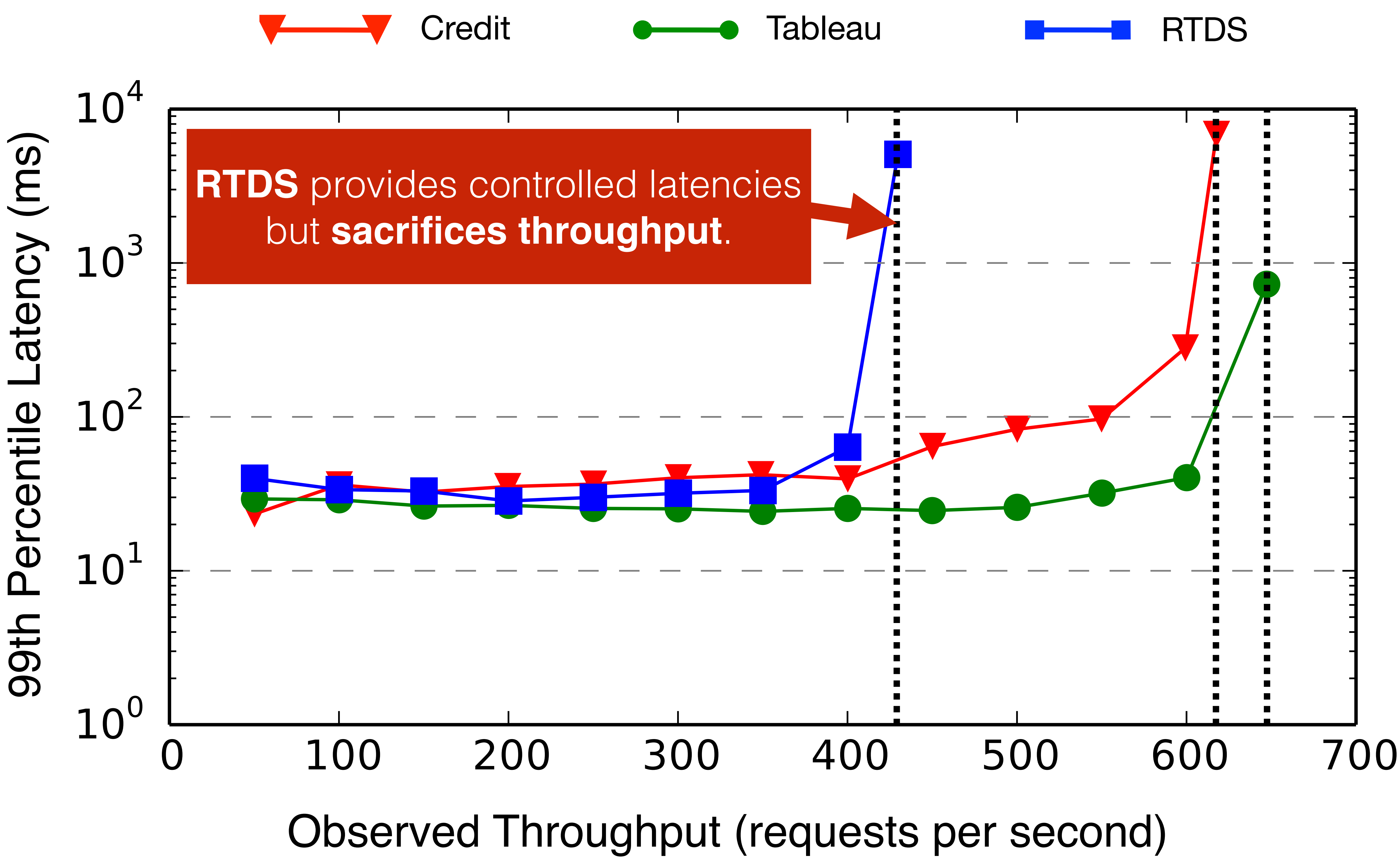
VMs Capped at 25%, 100K files, I/O background workload

Peak Throughput



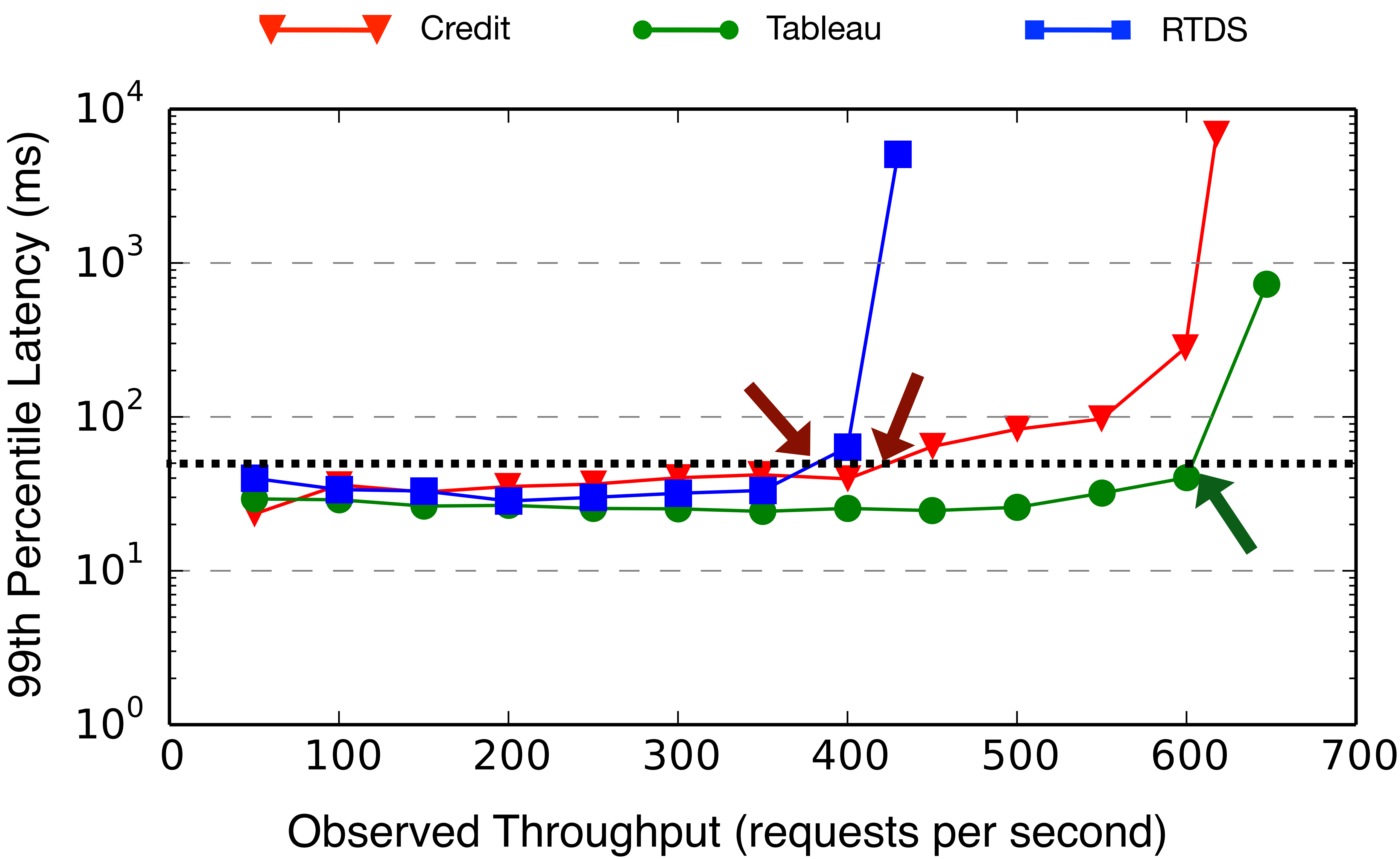
VMs Capped at 25%, 100K files, I/O background workload

Peak Throughput



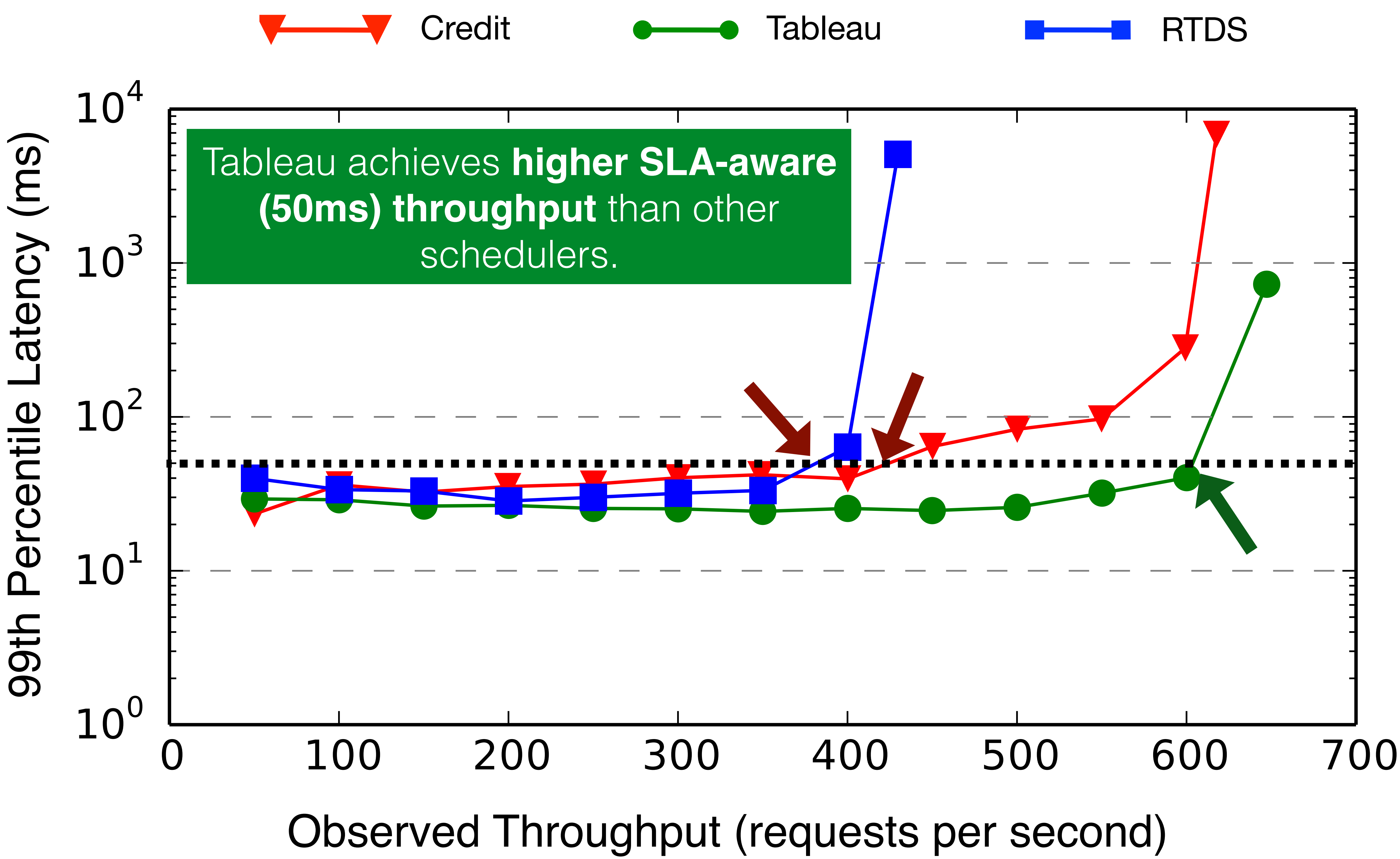
VMs Capped at 25%, 100K files, I/O background workload

SLA-Aware Throughput



VMs Capped at 25%, 100K files, I/O background workload

SLA-Aware Throughput (Capped Scenario)

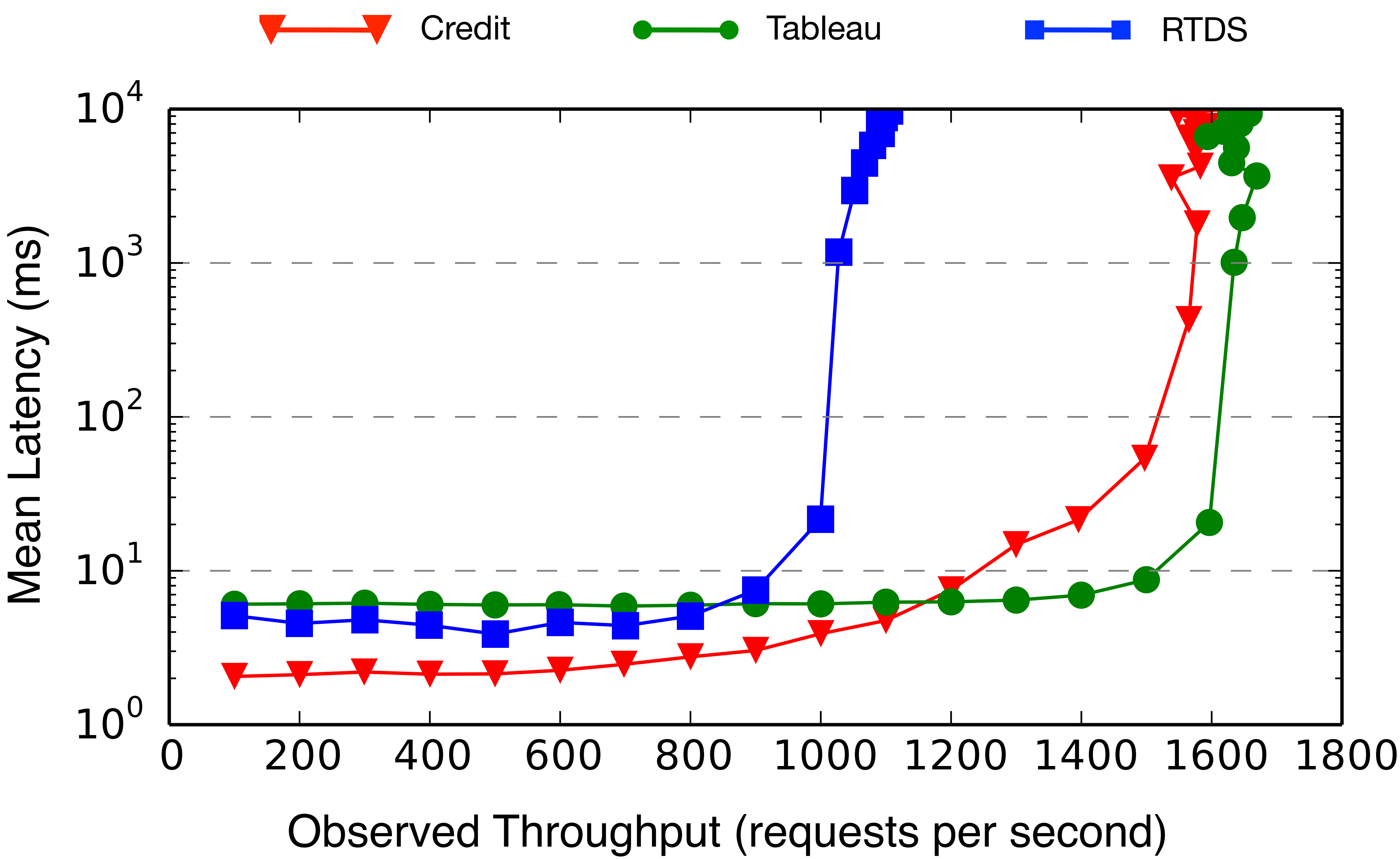


VMs Capped at 25%, 100K files, I/O background workload

Tableau Results in Higher Mean Latencies

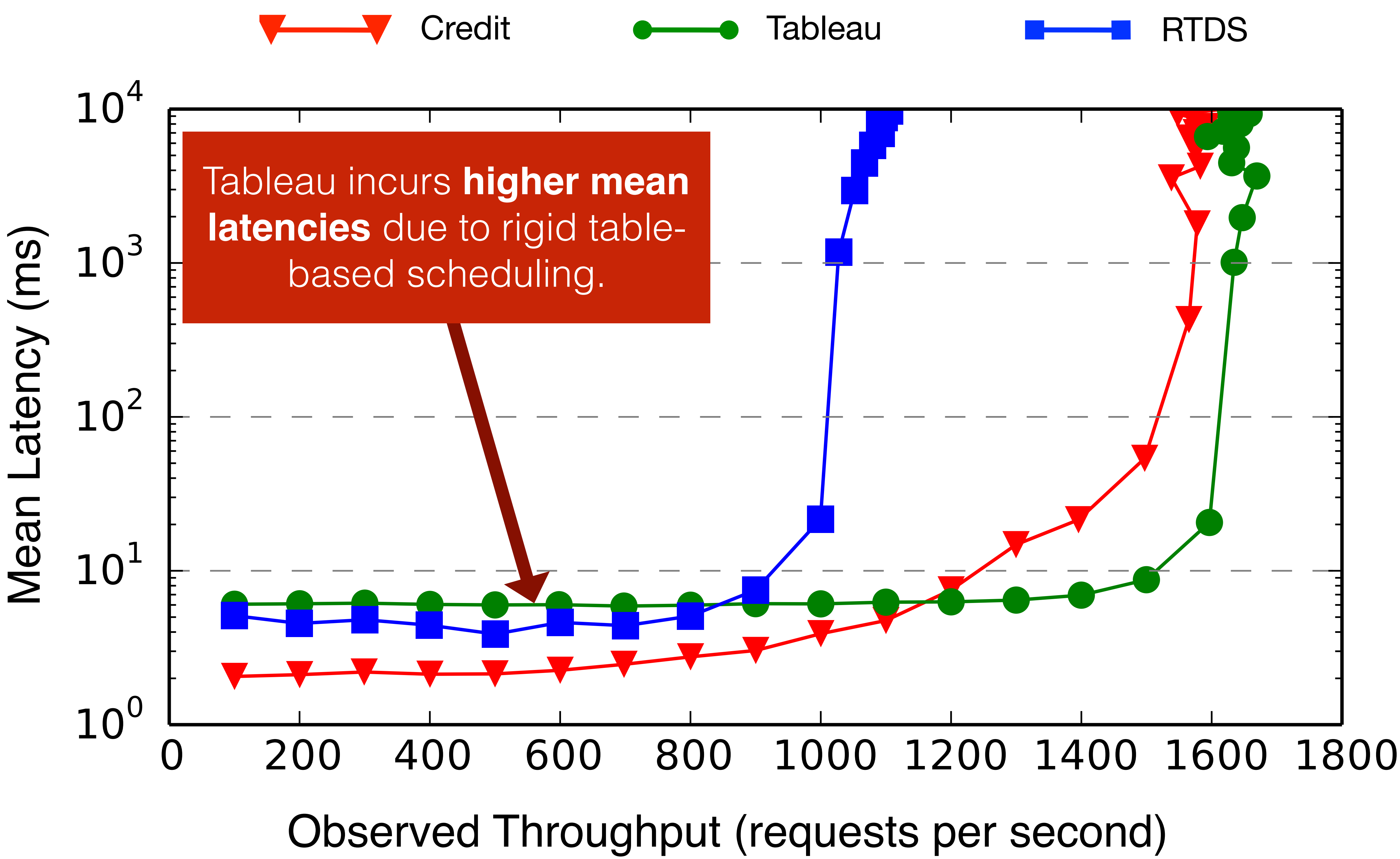
Hard-capped VMs under Tableau incur
higher mean latencies.

Tableau Results in Higher Mean Latencies



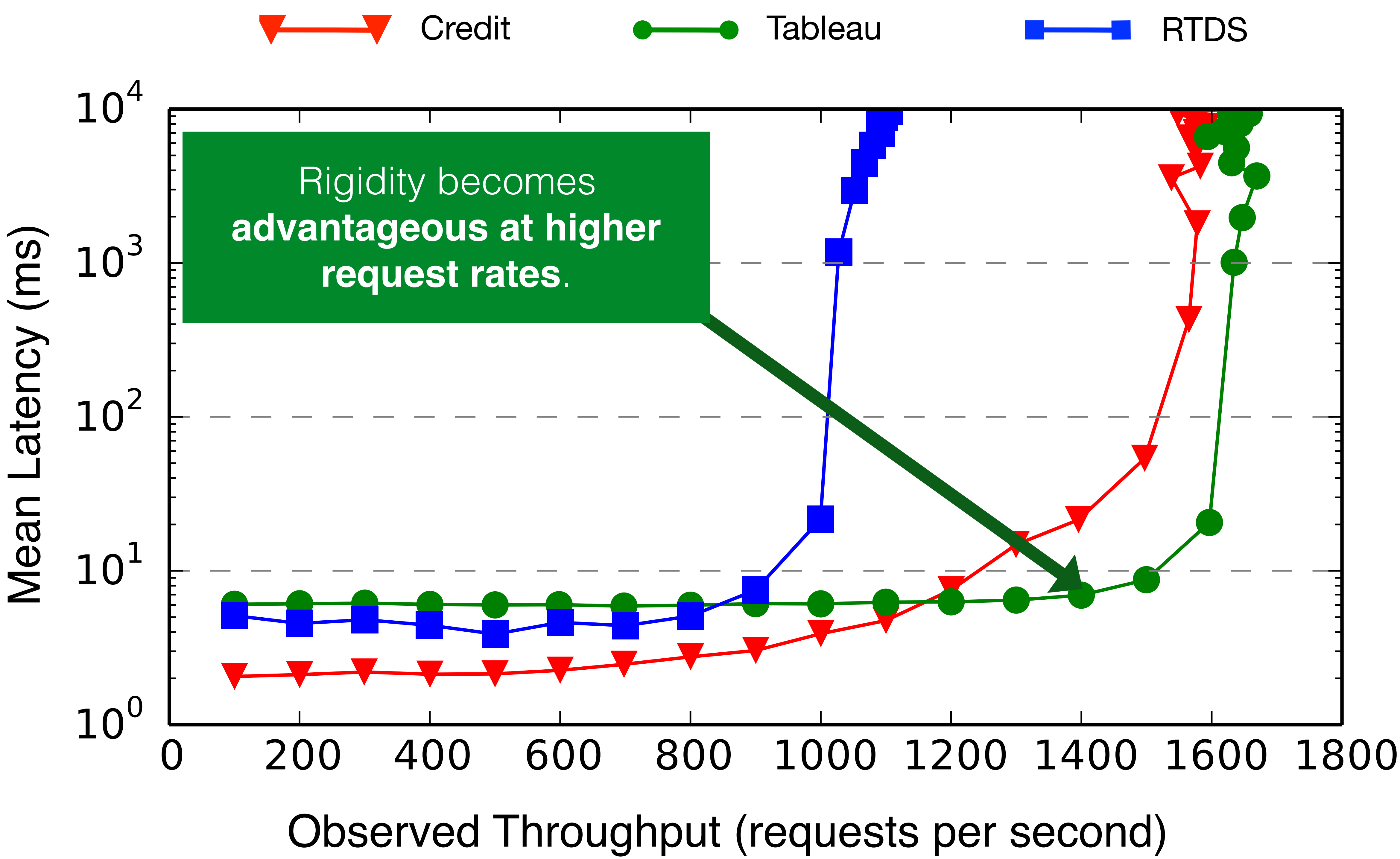
Capped VMs, 1K files, I/O background workload

Tableau Results in Higher Mean Latencies



Capped VMs, 1K files, I/O background workload

Tableau Results in Higher Mean Latencies



Capped VMs, 1K files, I/O background workload

Summary of Results

Tableau **incurs lower runtime overheads** compared to the other evaluated Xen schedulers

Tableau enables **accurate control over scheduling latency**.

Tableau achieves **higher SLA-aware application throughput.**

Hard capped VMs under Tableau incur **higher mean latency**, but entirely controllable.

This Talk

- Tableau
- Evaluation
- **Conclusion**

Contributions

Tableau

An **unorthodox scheduling approach** tailored for high-density public clouds.

Efficient

Incurs low overheads

Predictable

Accurate control over scheduling latency.

High-throughput

Provides high SLA-aware throughput.

Thanks!

Source-code available at:

http://tableau.mpi-sws.org/