Scalable Cloud Security via Asynchronous Virtual Machine Introspection

Sundaresan (sunny) Rajasekaran, Zhen Ni, Harpreet Singh Chawla, Neel Shah, Timothy Wood and Emery Berger[†·]

The George Washington University
[†]University of Massachusetts, Amherst

Introduction

- Software will always be vulnerable to attacks.
- Existing techniques for prevention are slow to detect attacks.
- Need a way for cloud platforms to provide security functionality as a service.

Introduction

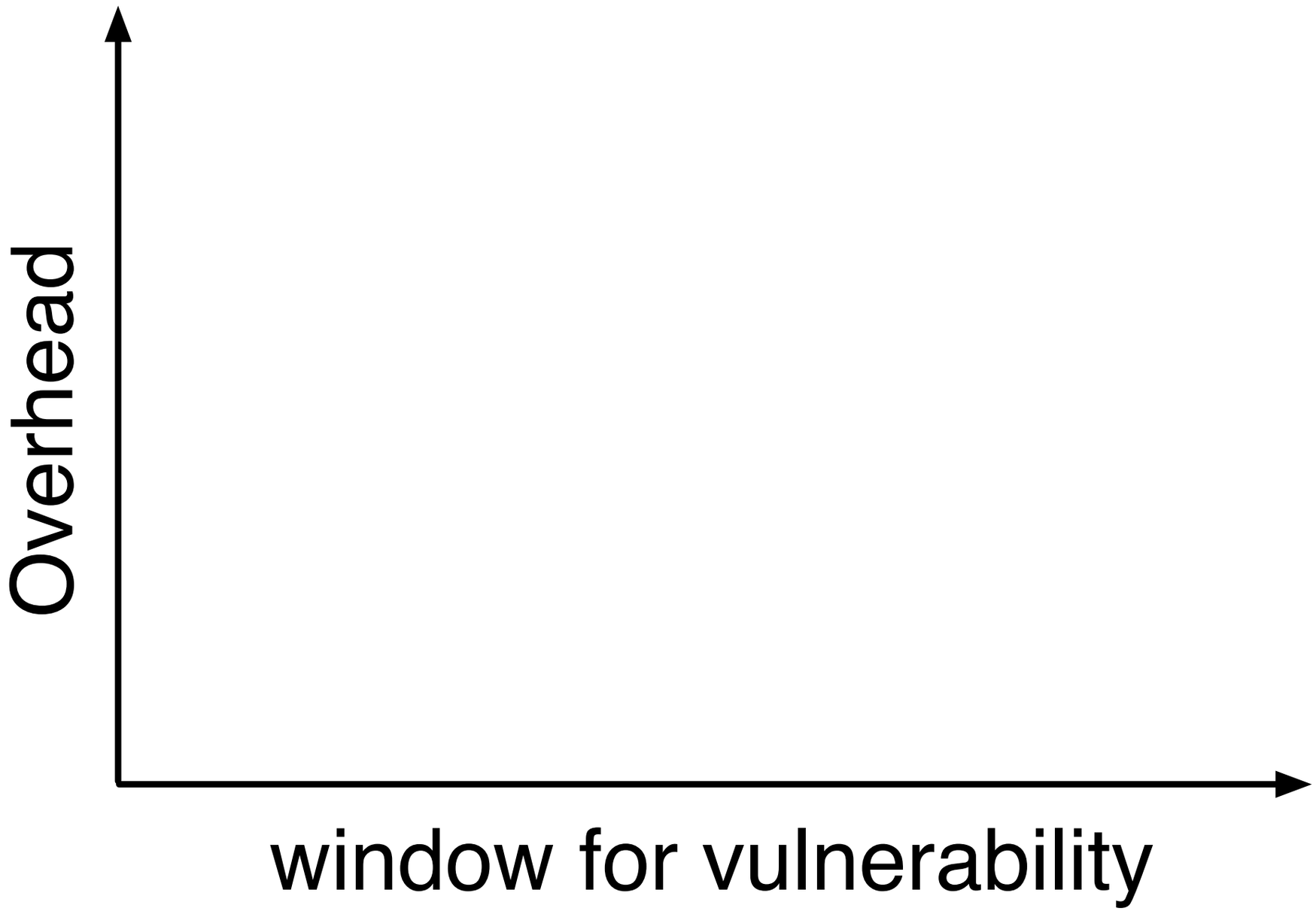
- Software will always be vulnerable to attacks.
- Existing techniques for prevention are slow to detect attacks.
- Need a way for cloud platforms to provide security functionality as a service.

How can the cloud detect attacks inside a VM?
How to provide strong security guarantees at low cost?

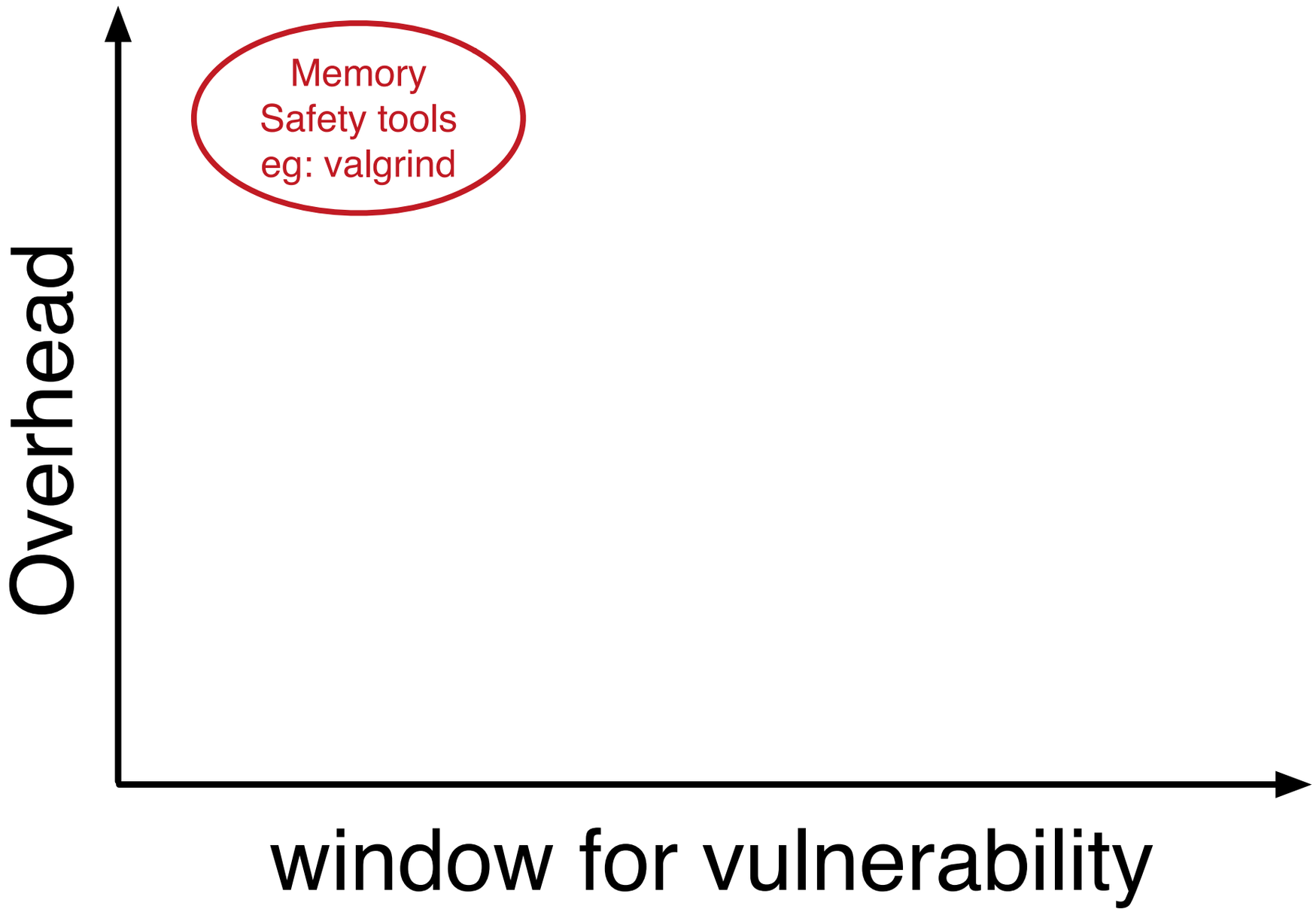
ScaaS

- **Scanning as a Service** framework for security in cloud data centers.
    - Scans for a wide range of attacks within both application and the operating system.
- Uses an **asynchronous checkpointing** mechanism to replicate a VM's memory onto a Scanner host for analysis.
- Uses **VM introspection** techniques to study the memory of the virtual machine.

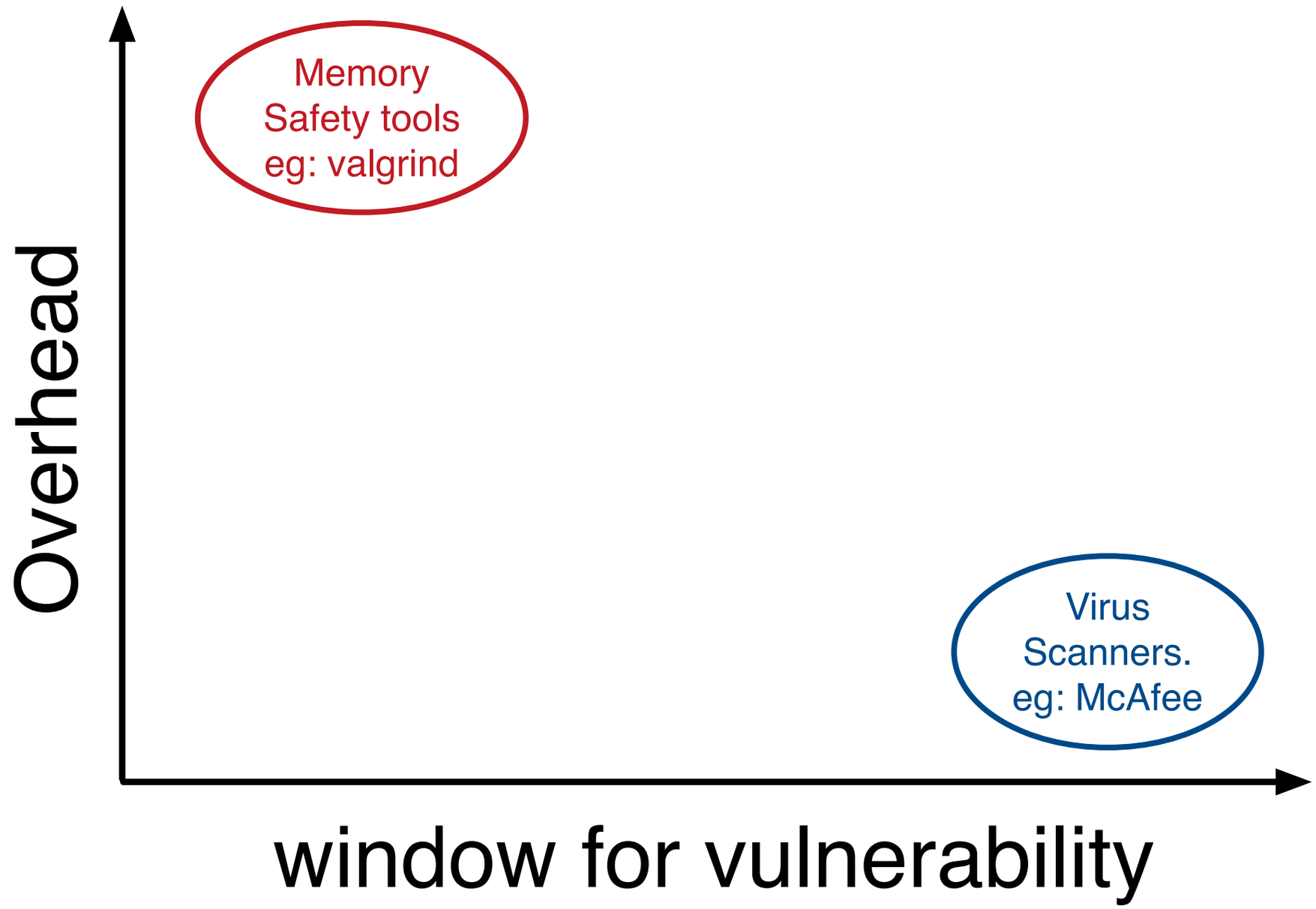
Where do we stand?



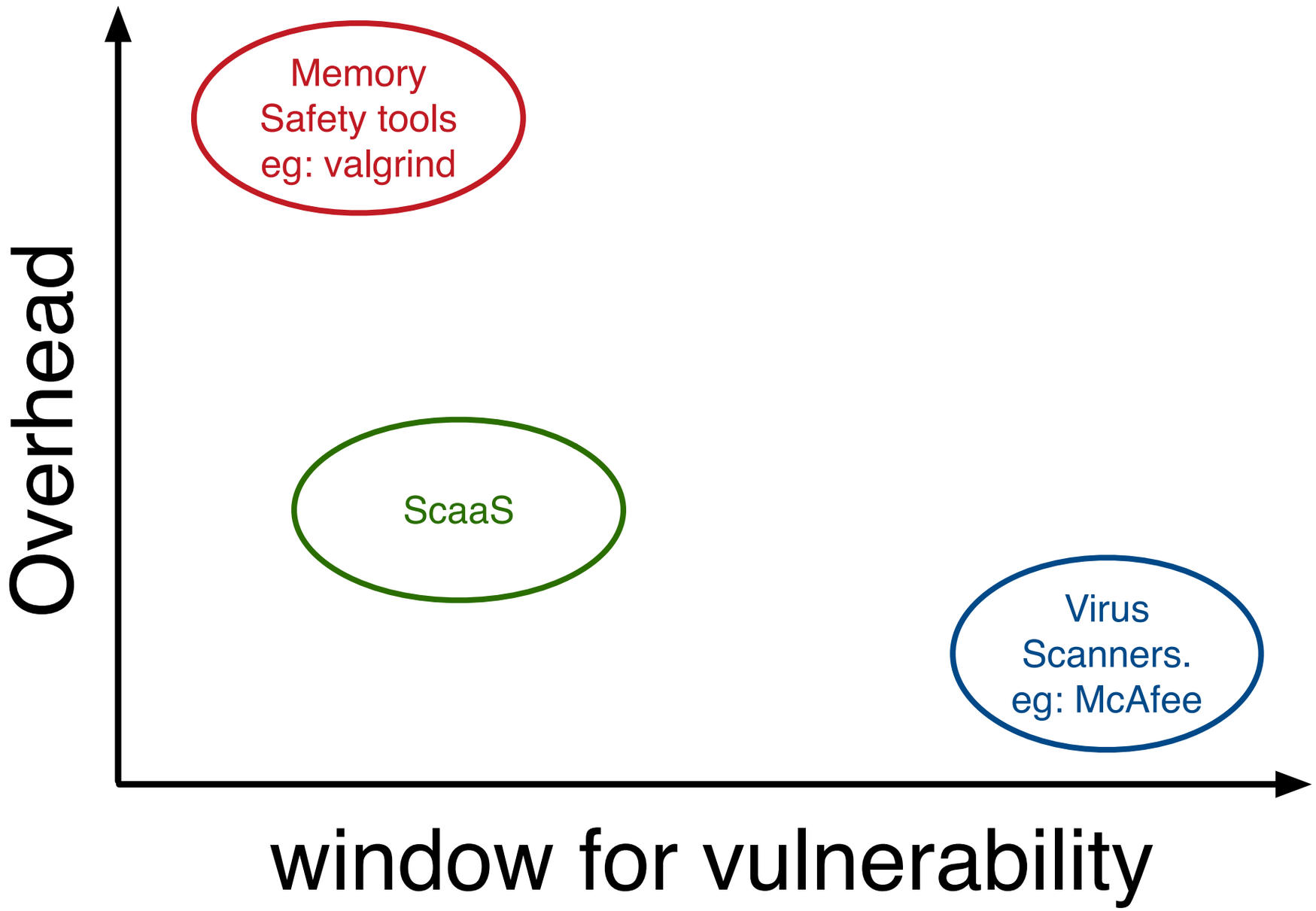
Where do we stand?



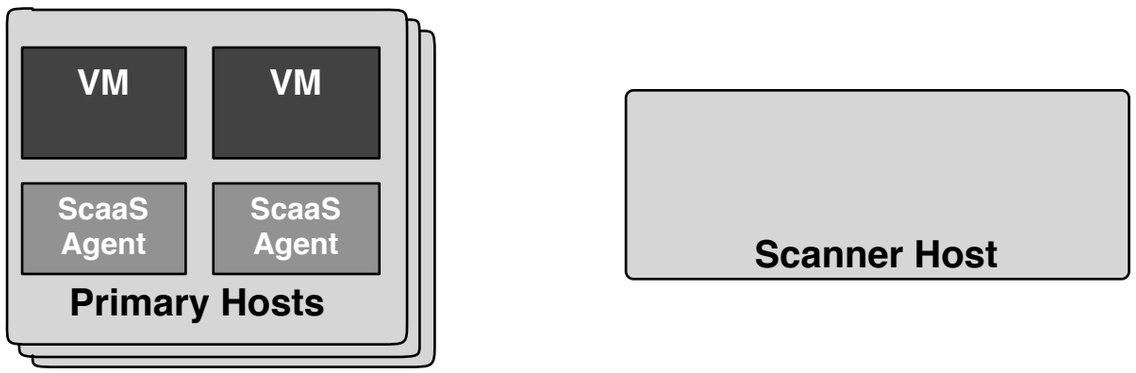
Where do we stand?



Where do we stand?

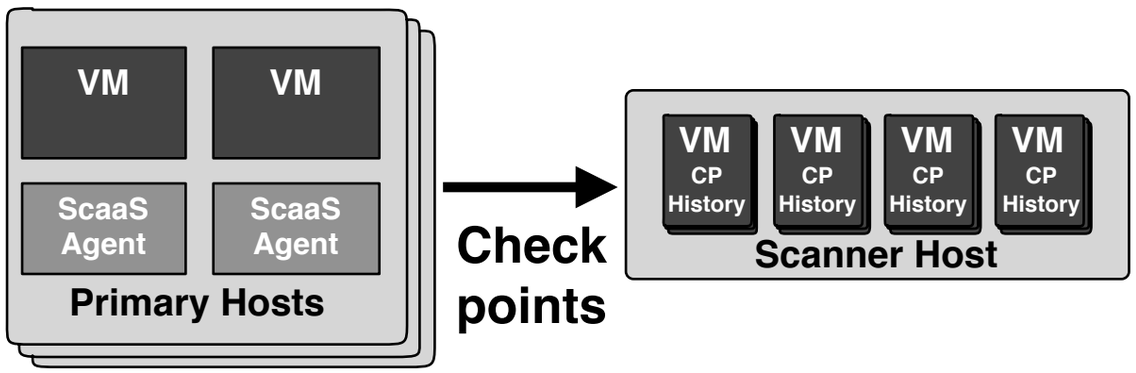


ScaaS Architecture



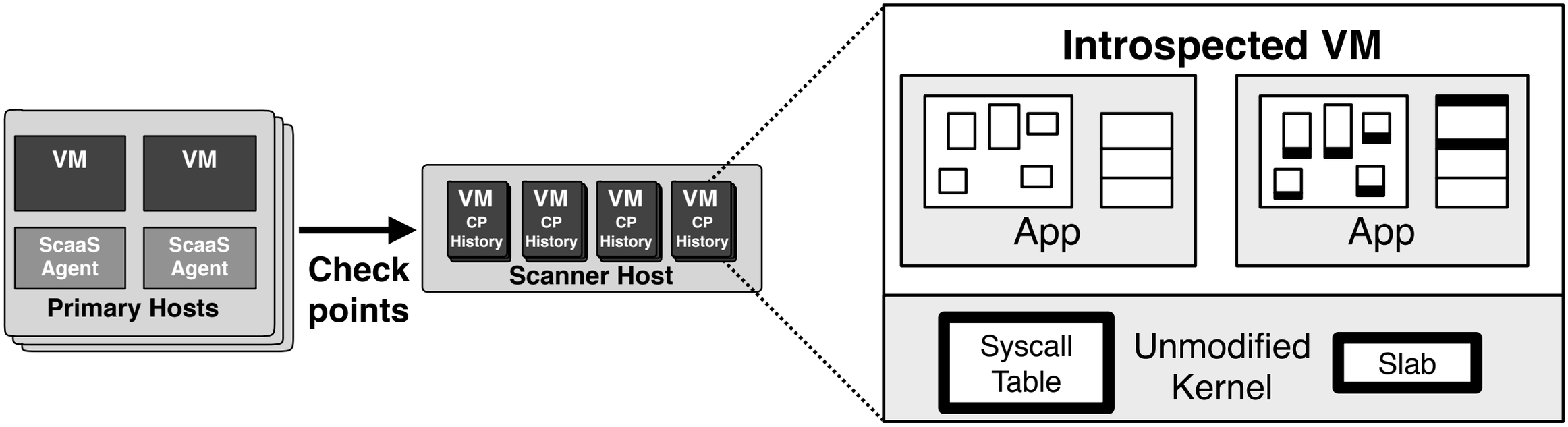
- VMs periodically send checkpoints to the Scanners for analysis.
- A Scanner host uses VM introspection techniques to search for evidence of vulnerabilities.
    - Ensures integrity of Key Kernel data structures.

ScaaS Architecture



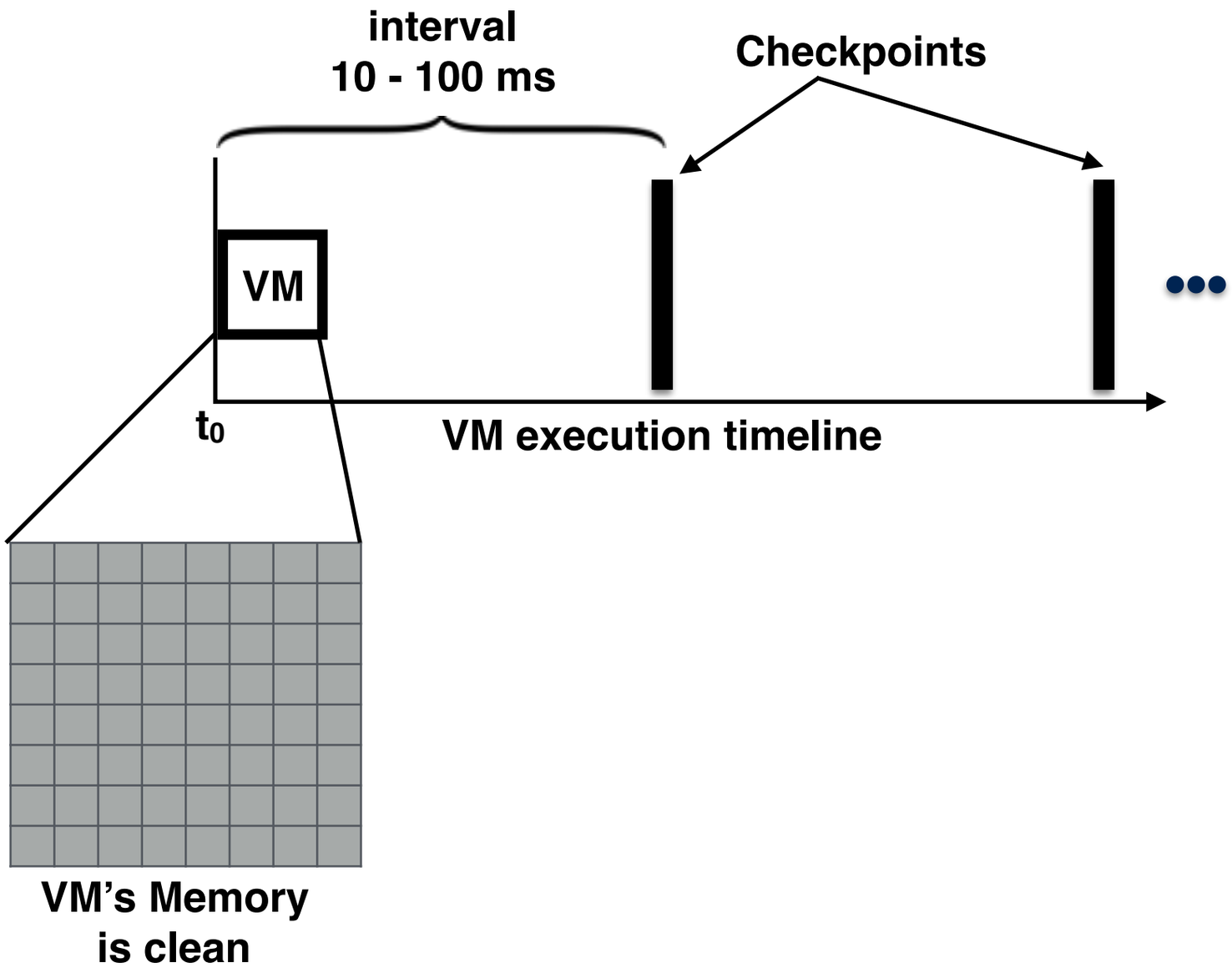
- VMs periodically send checkpoints to the Scanners for analysis.
- A Scanner host uses VM introspection techniques to search for evidence of vulnerabilities.
    - Ensures integrity of Key Kernel data structures.

ScaaS Architecture

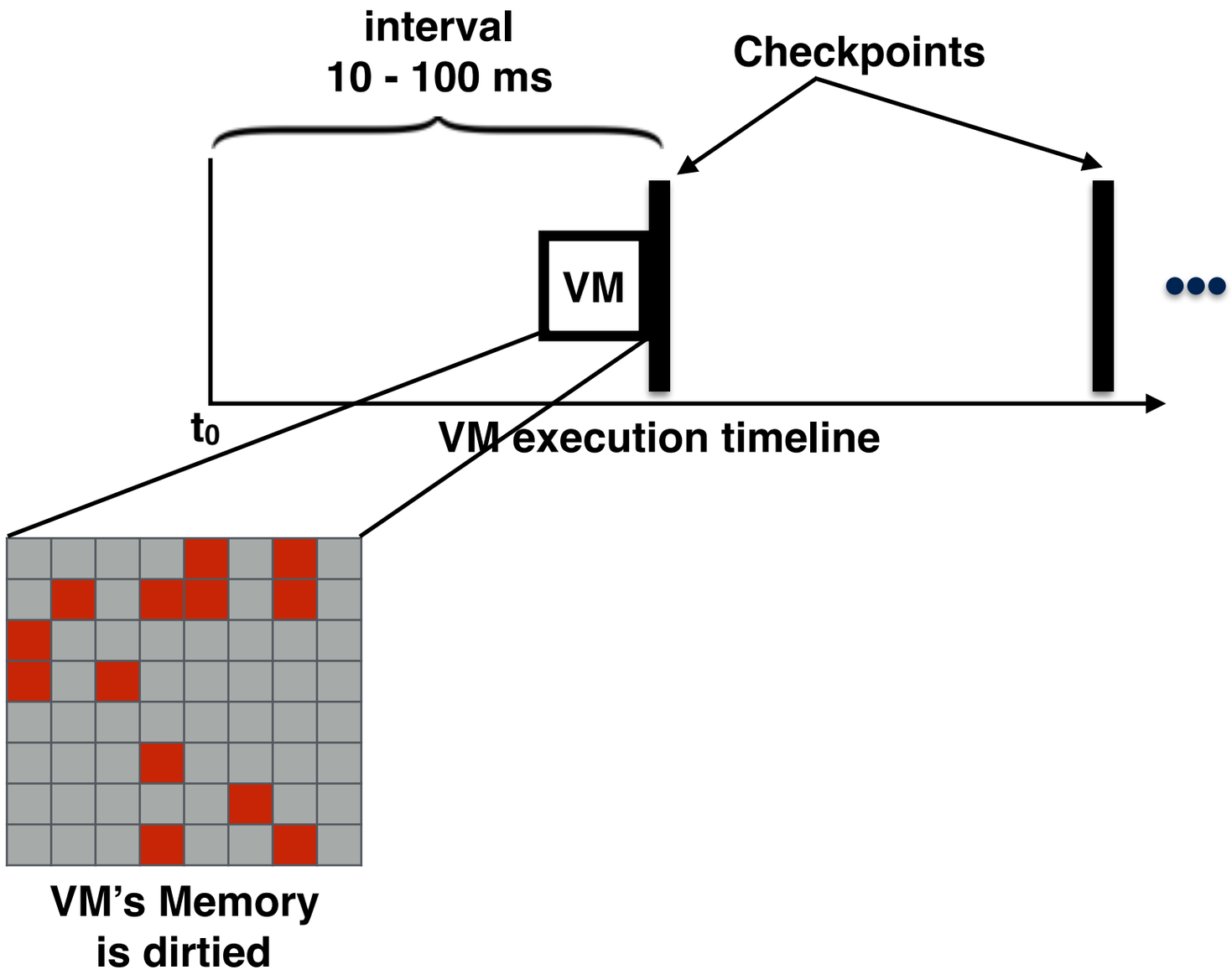


- VMs periodically send checkpoints to the Scanners for analysis.
- A Scanner host uses VM introspection techniques to search for evidence of vulnerabilities.
    - Ensures integrity of Key Kernel data structures.

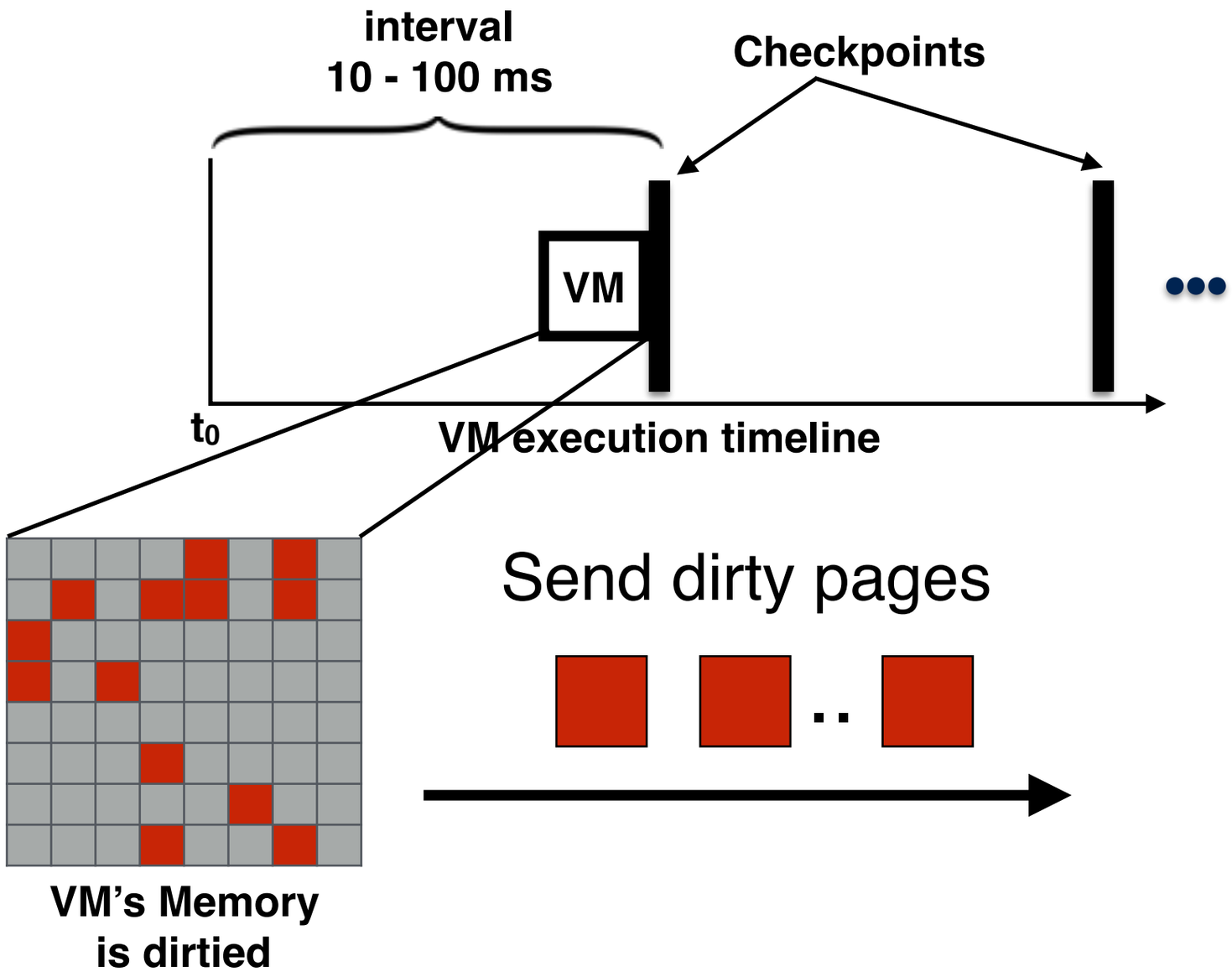
VM Checkpointing



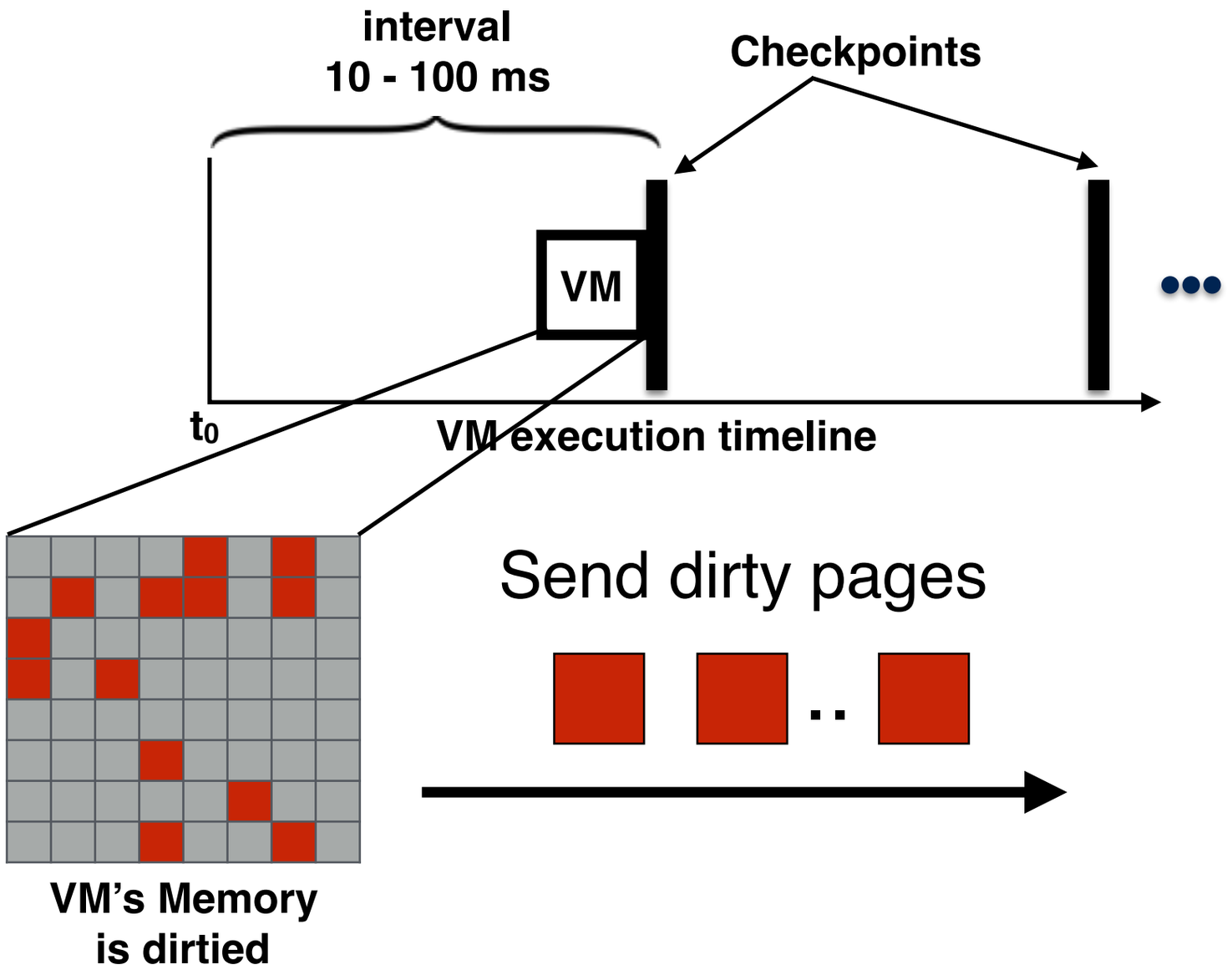
VM Checkpointing

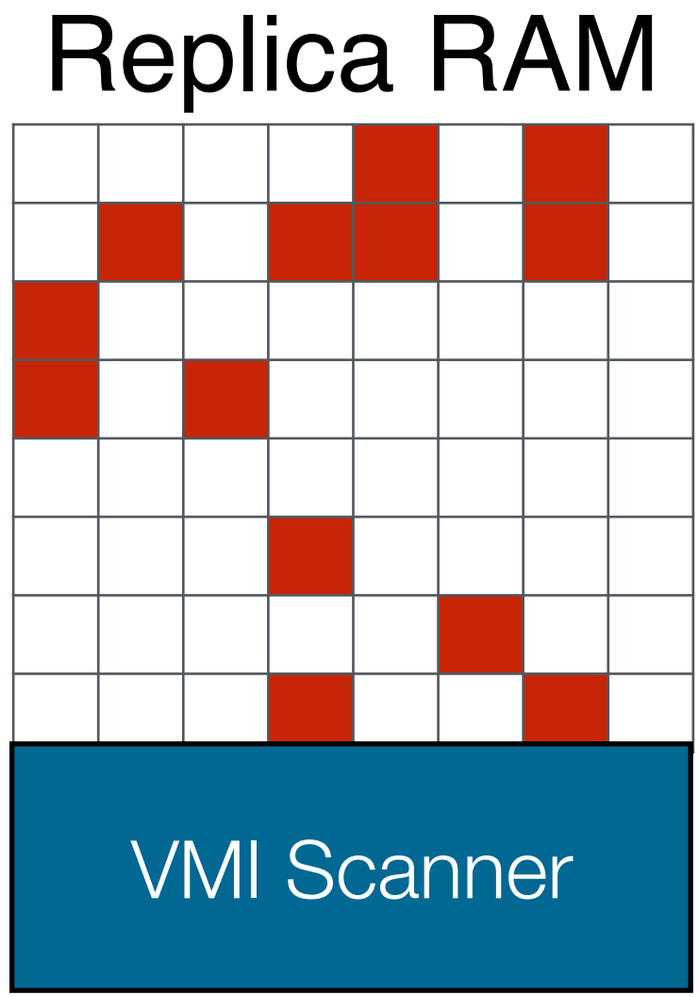


VM Checkpointing

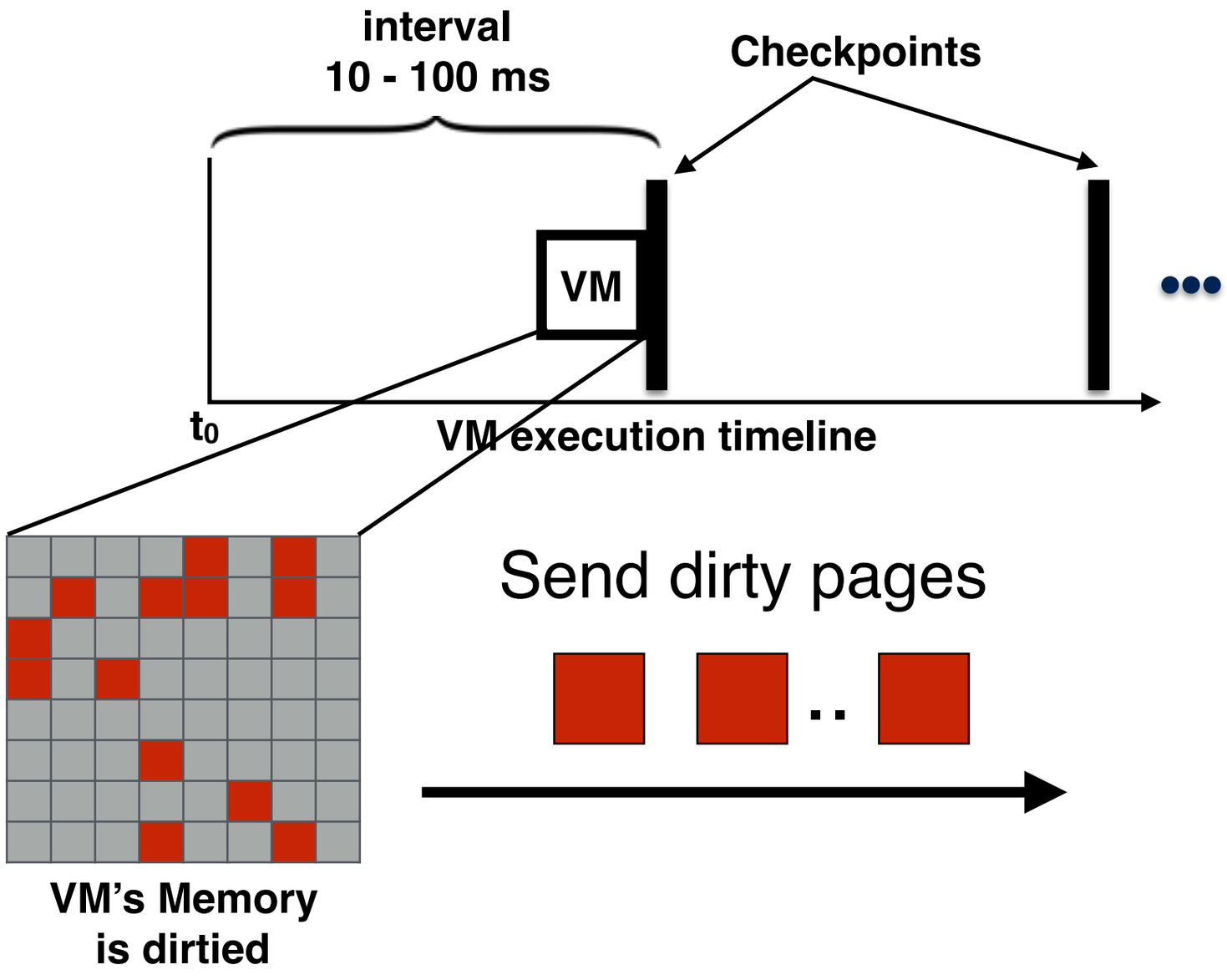


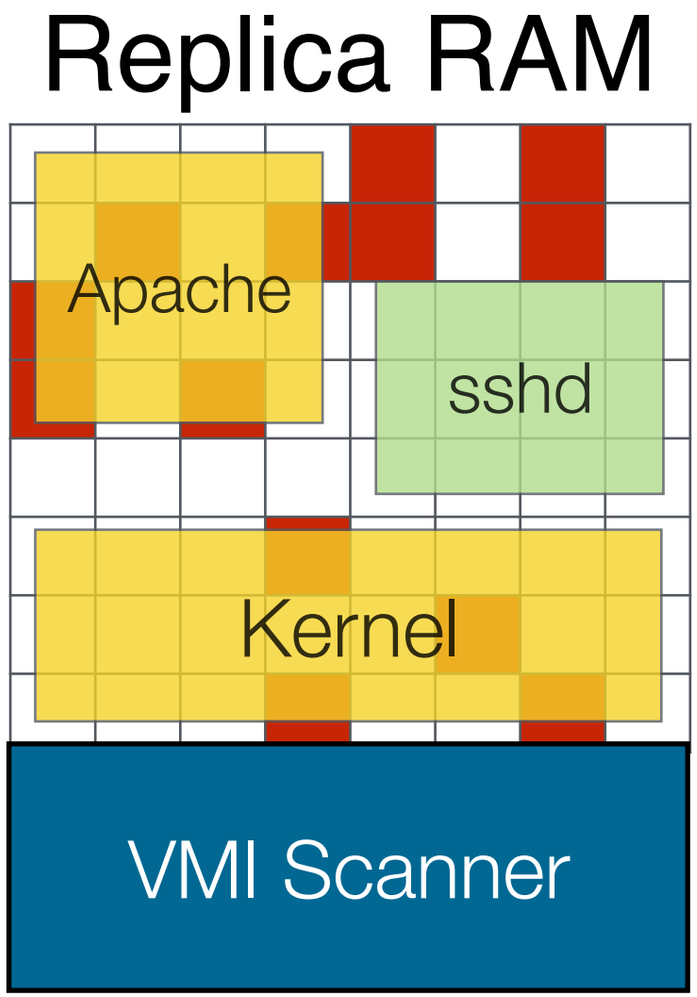
VM Checkpointing



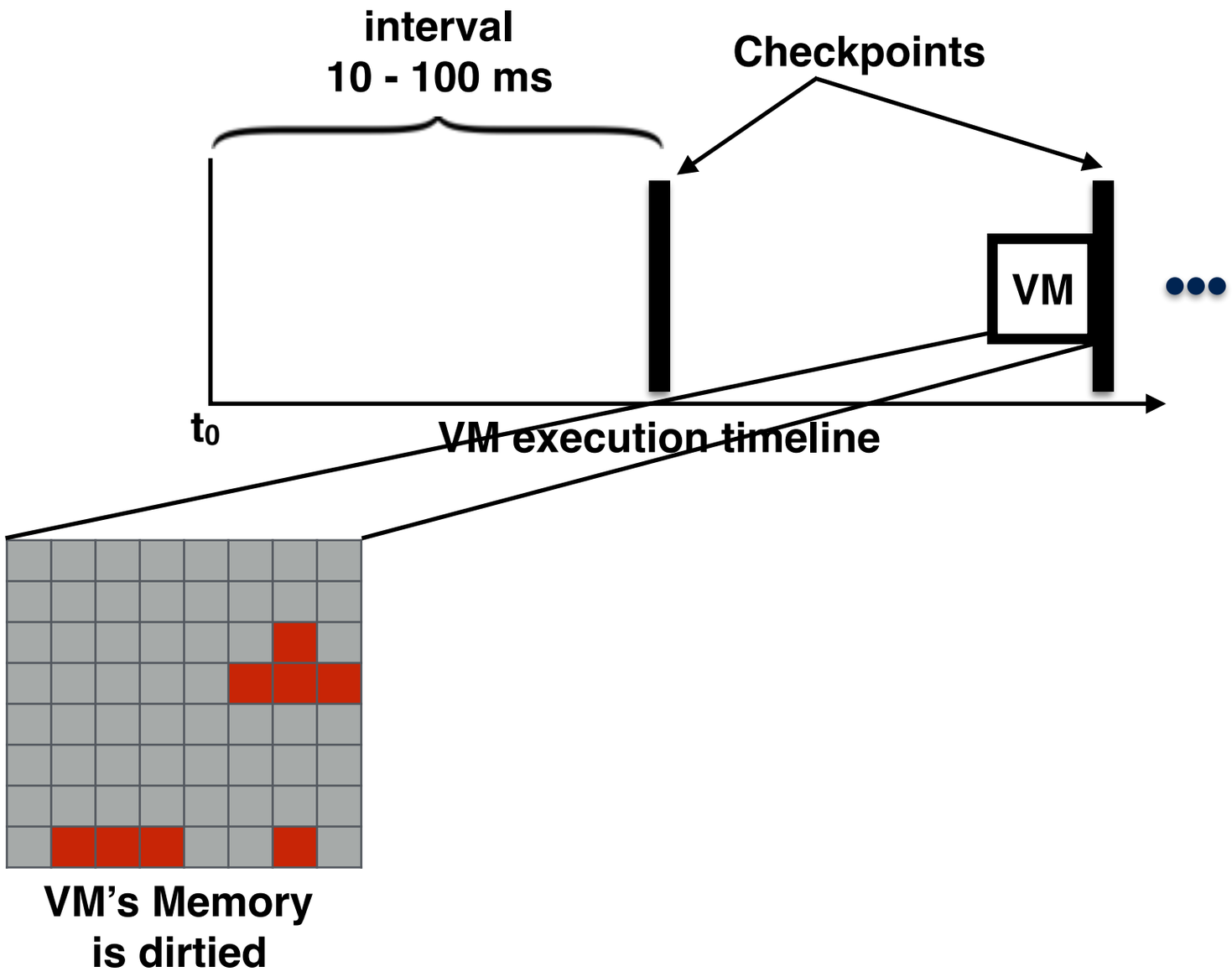


VM Checkpointing

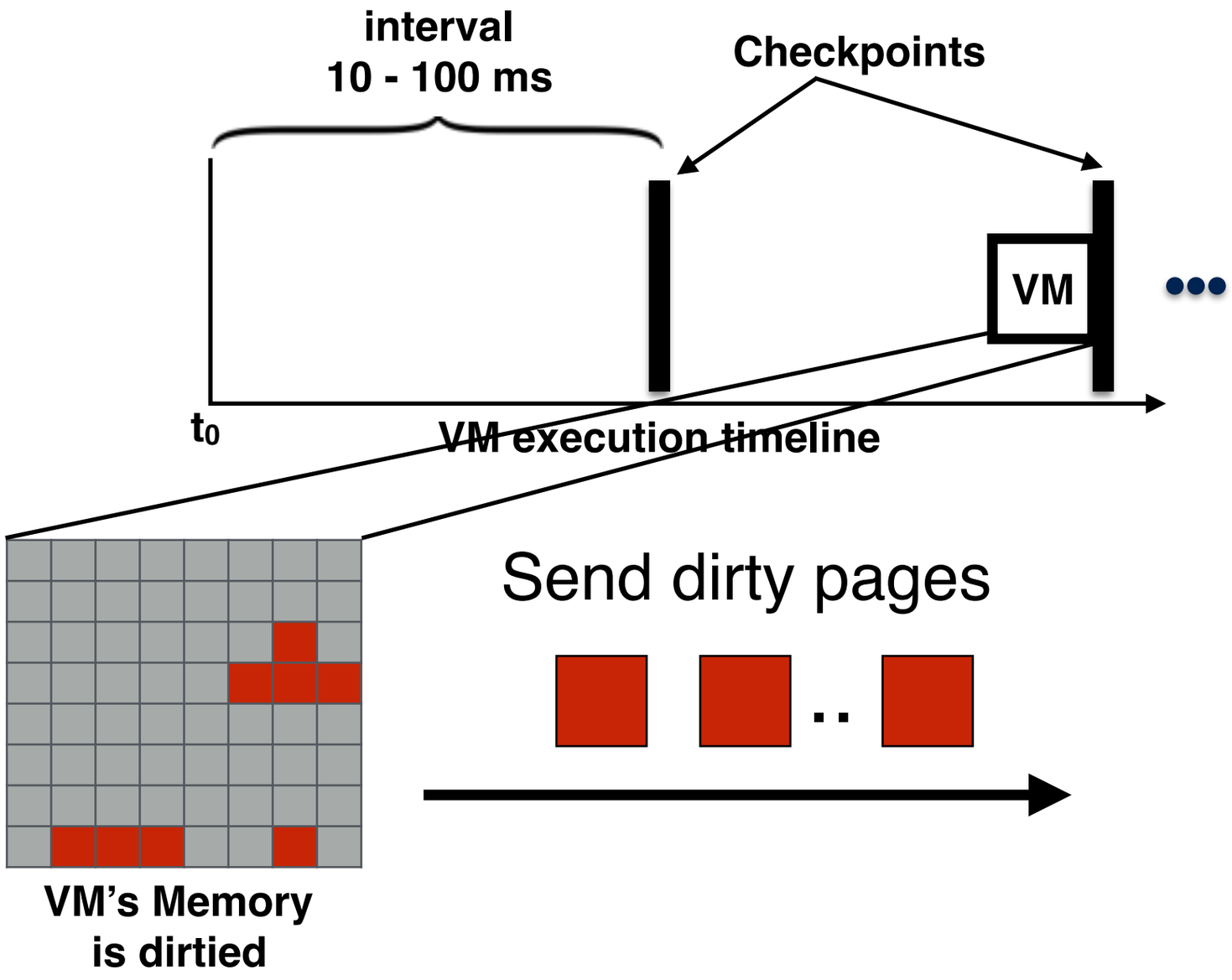




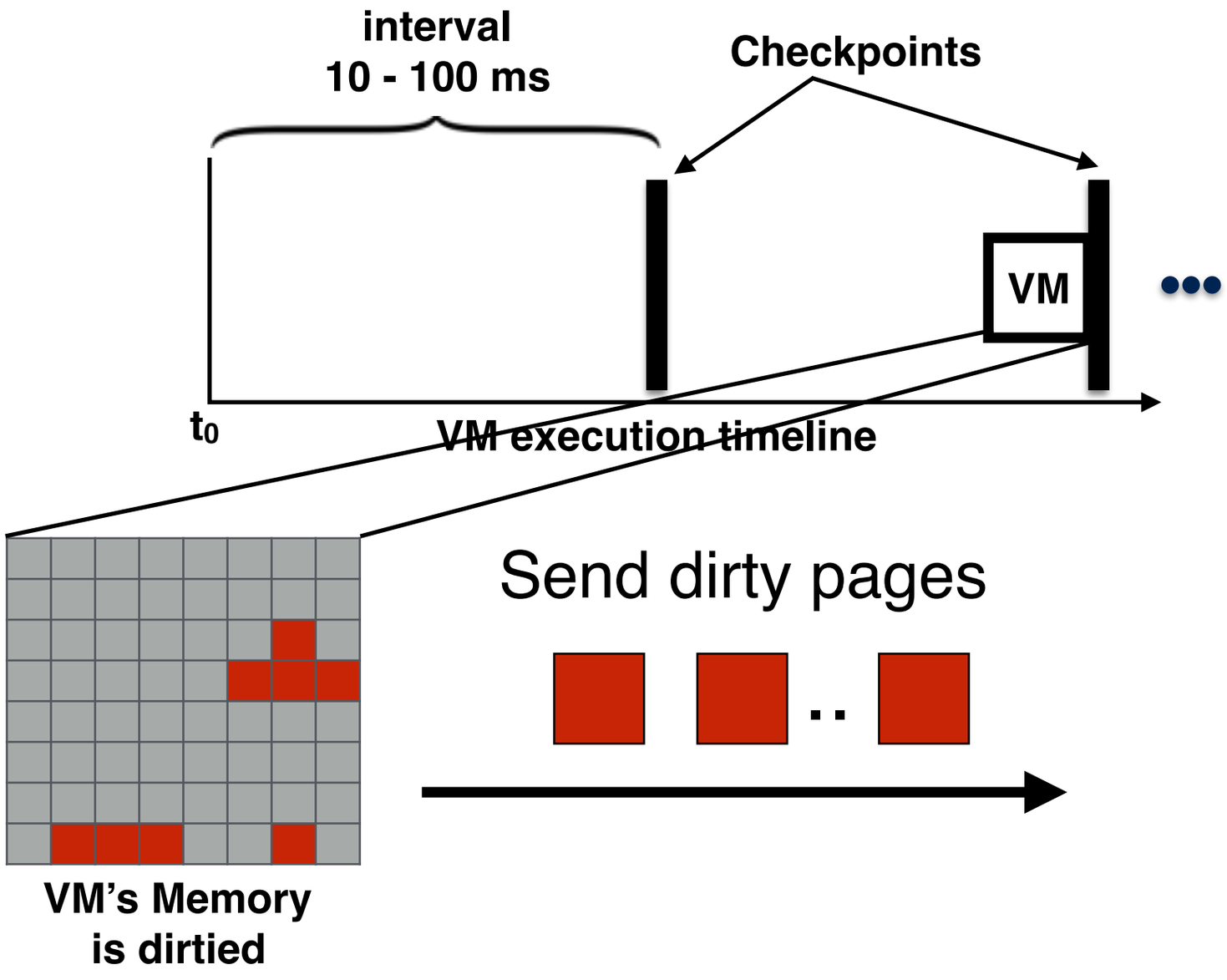
VM Checkpointing

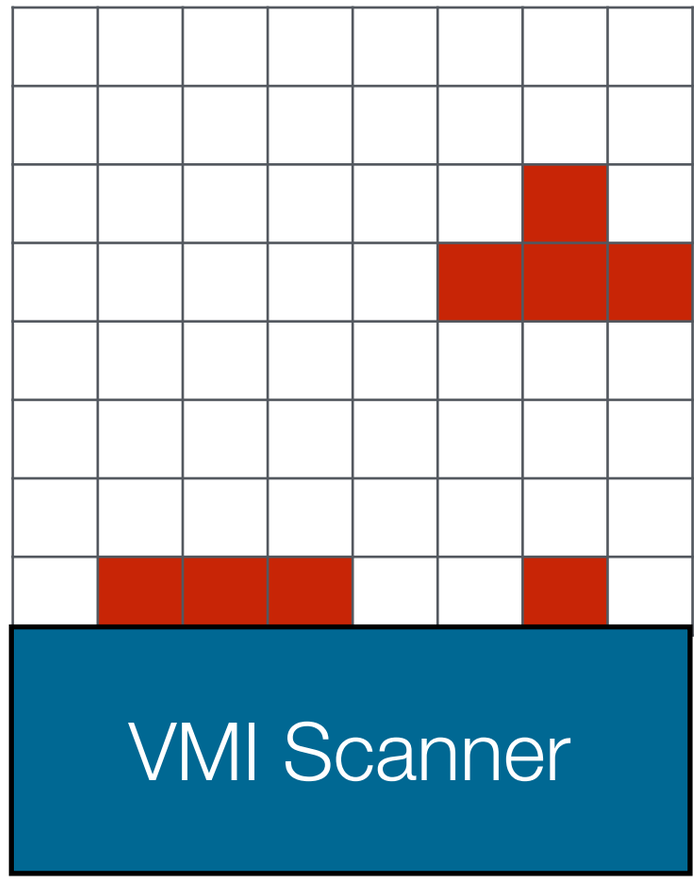


VM Checkpointing

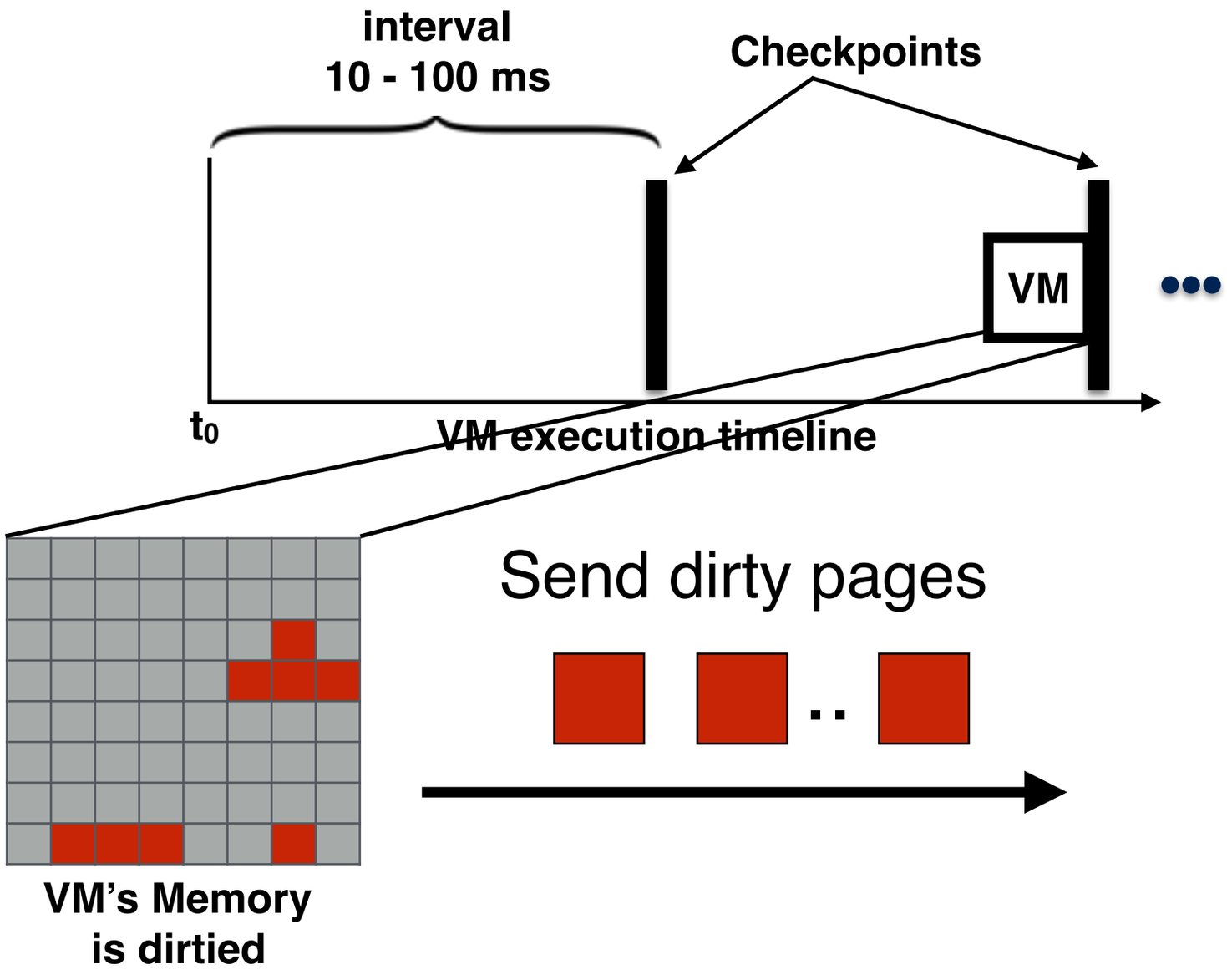


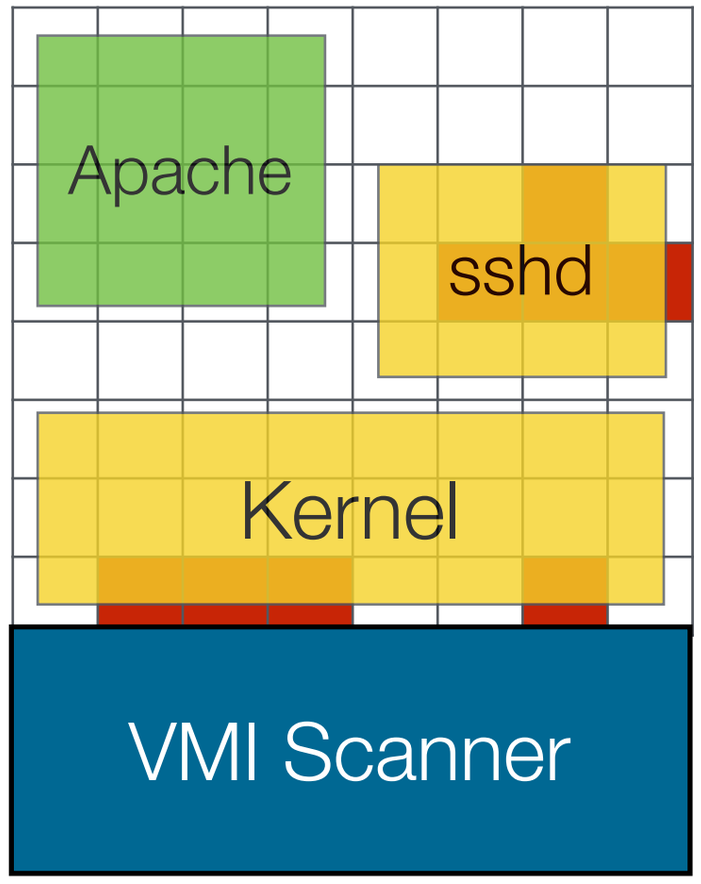
VM Checkpointing



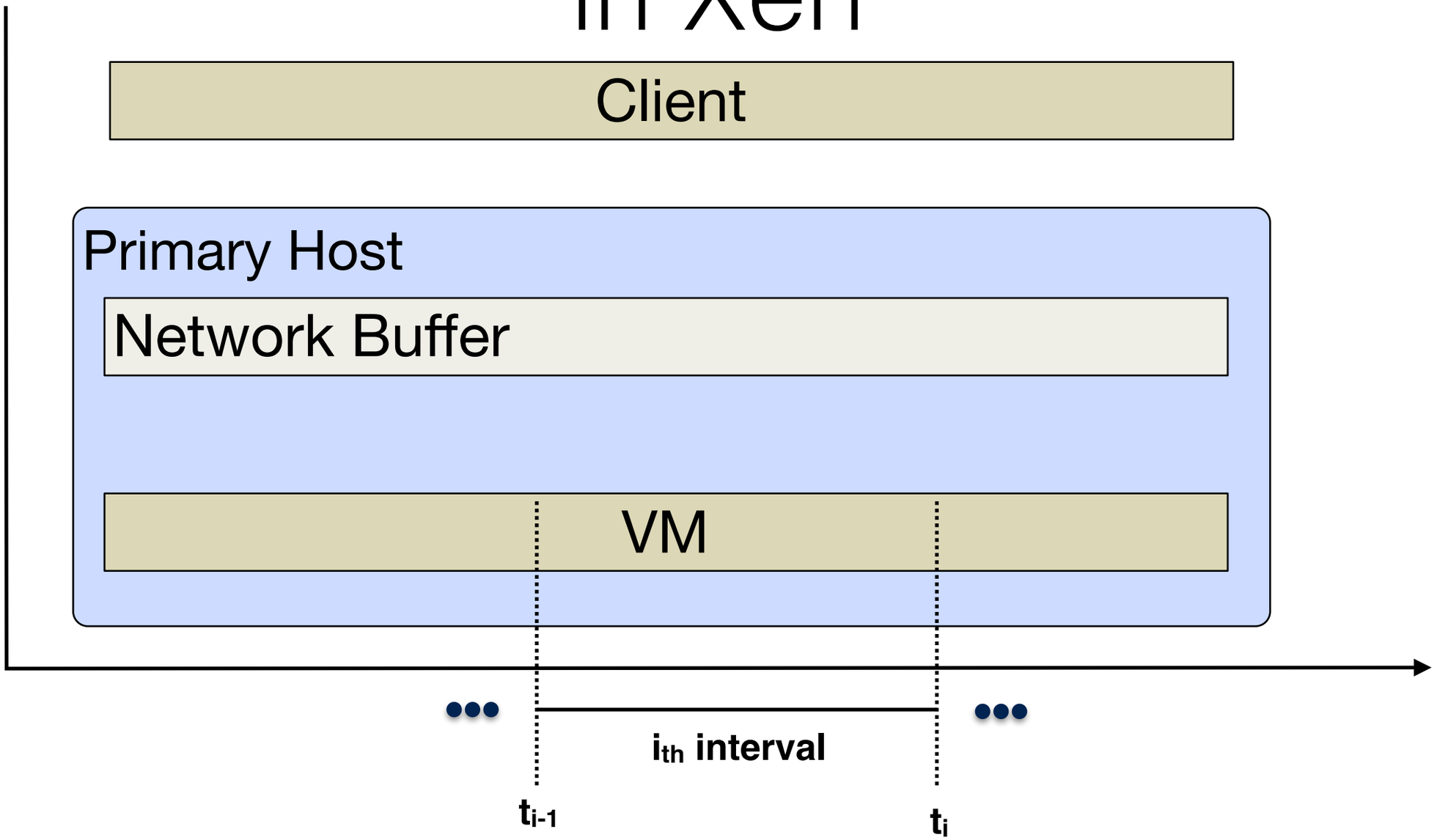


VM Checkpointing



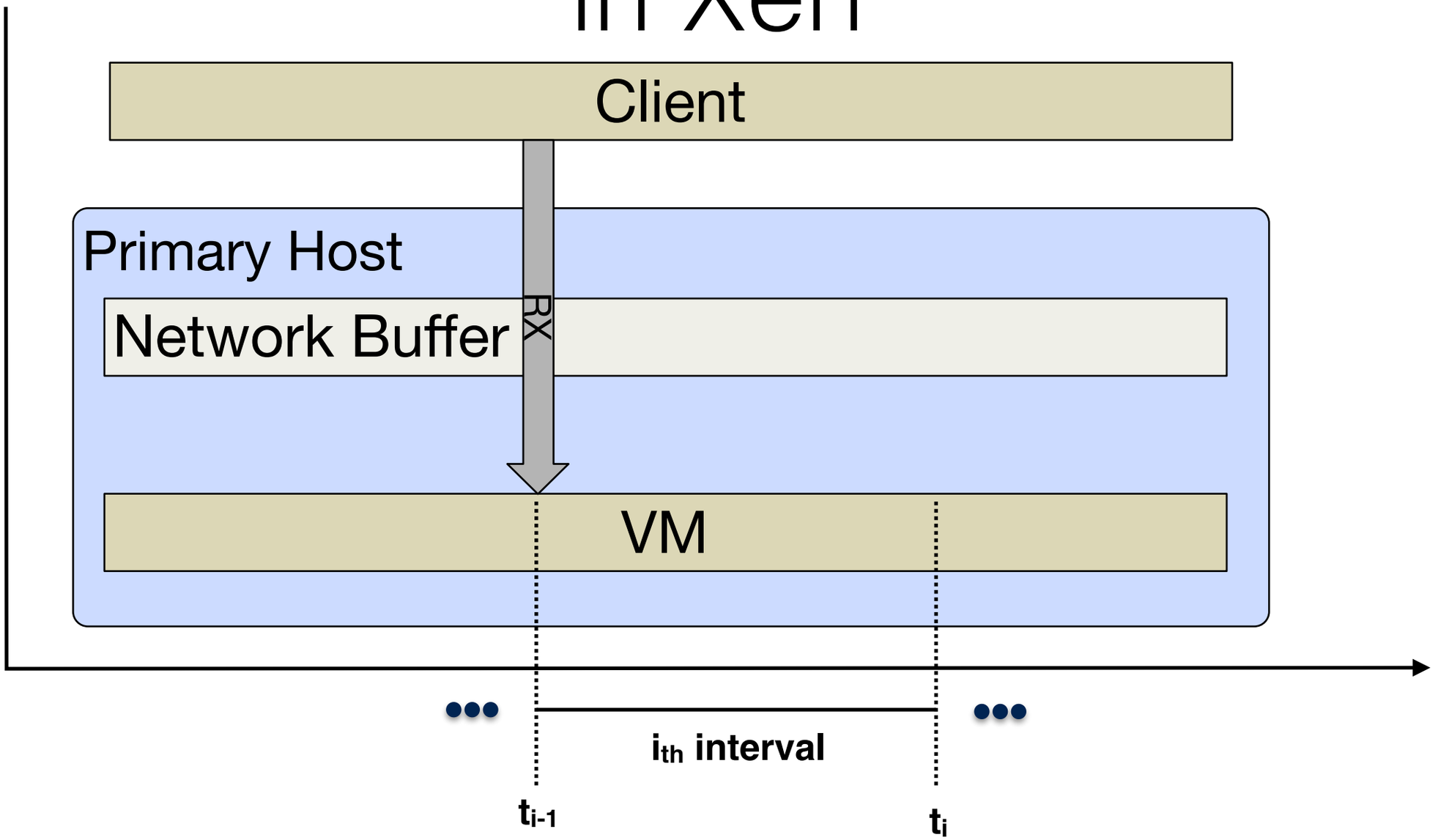


Network buffering using Remus in Xen



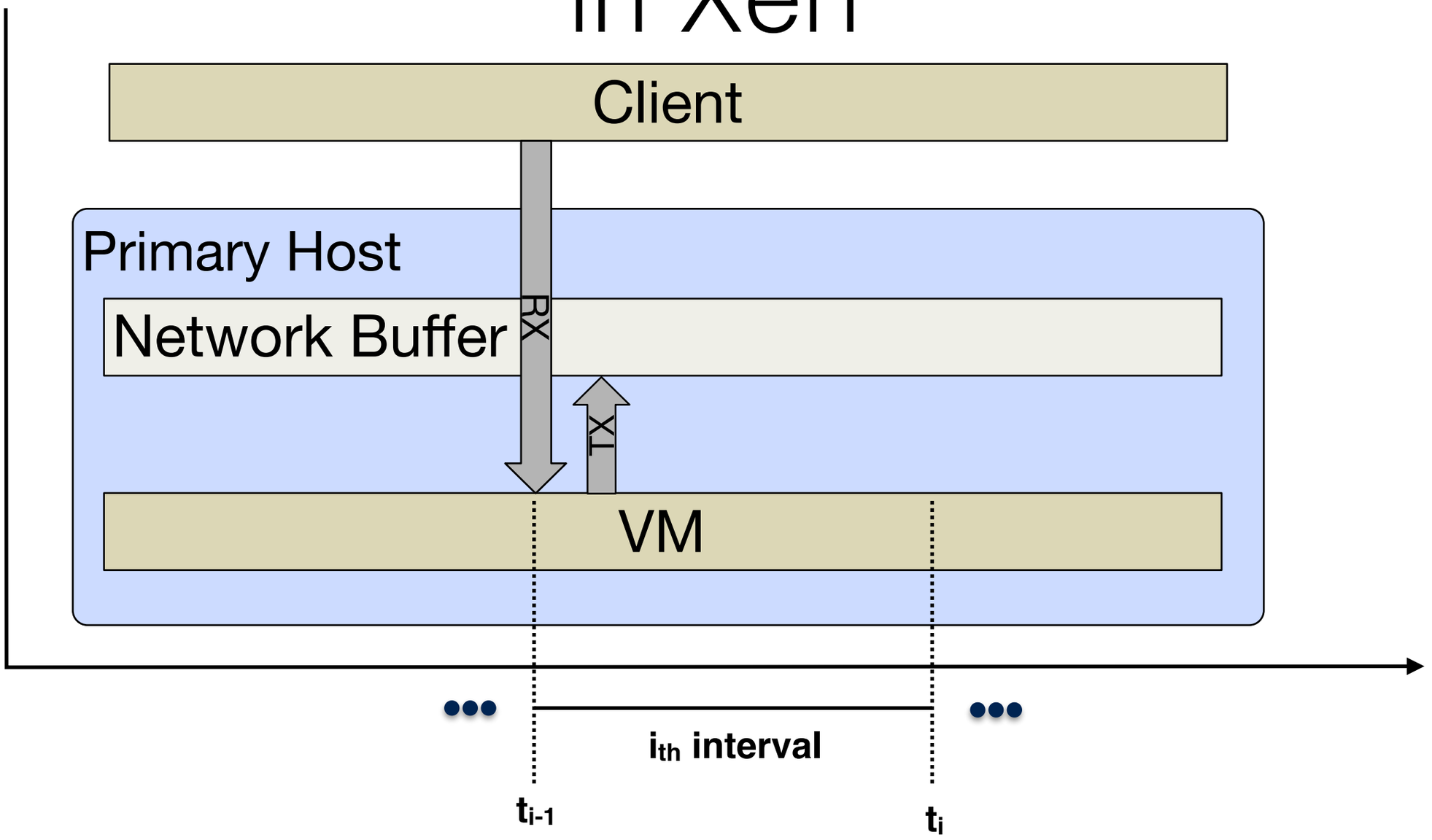
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



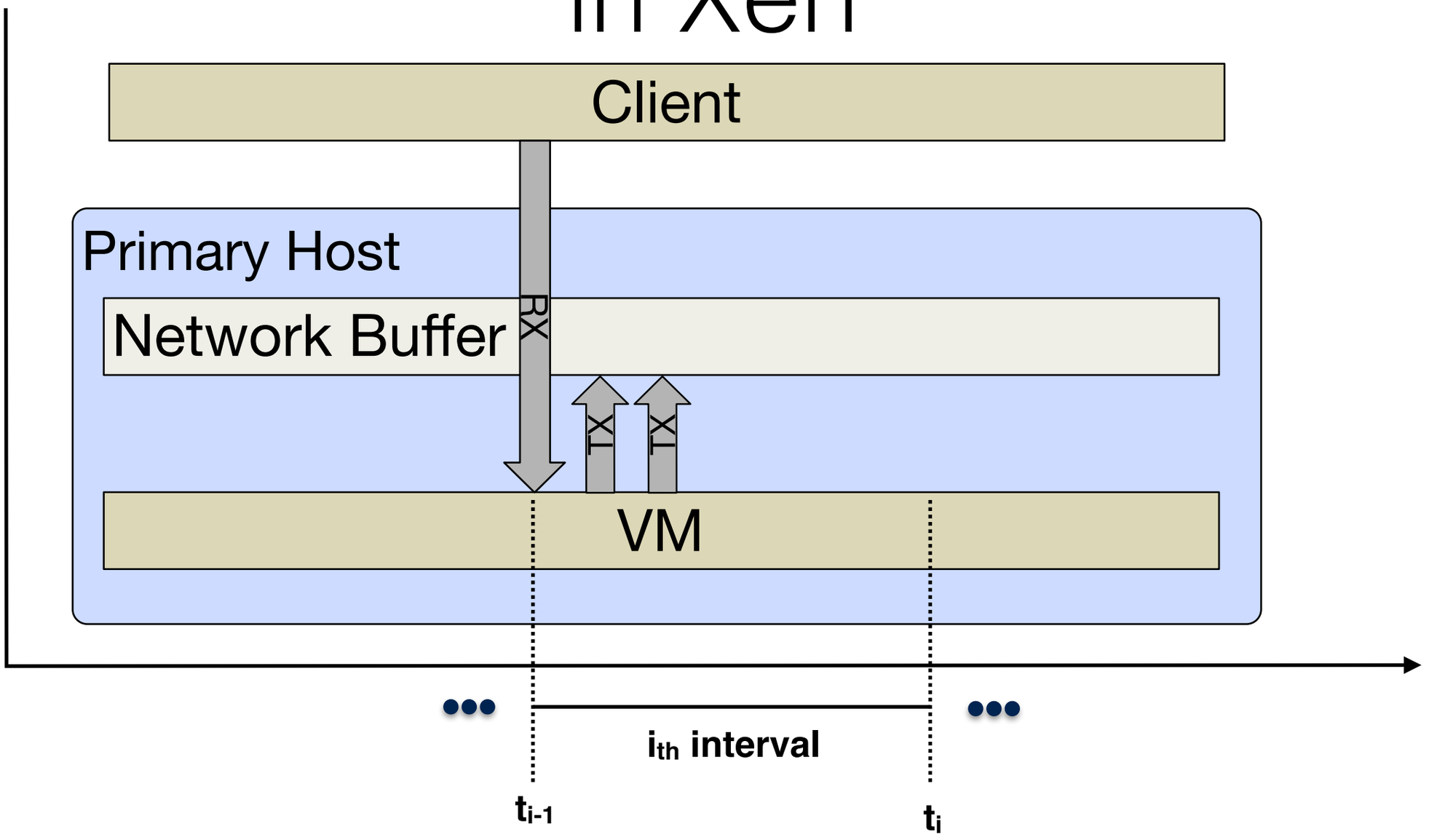
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



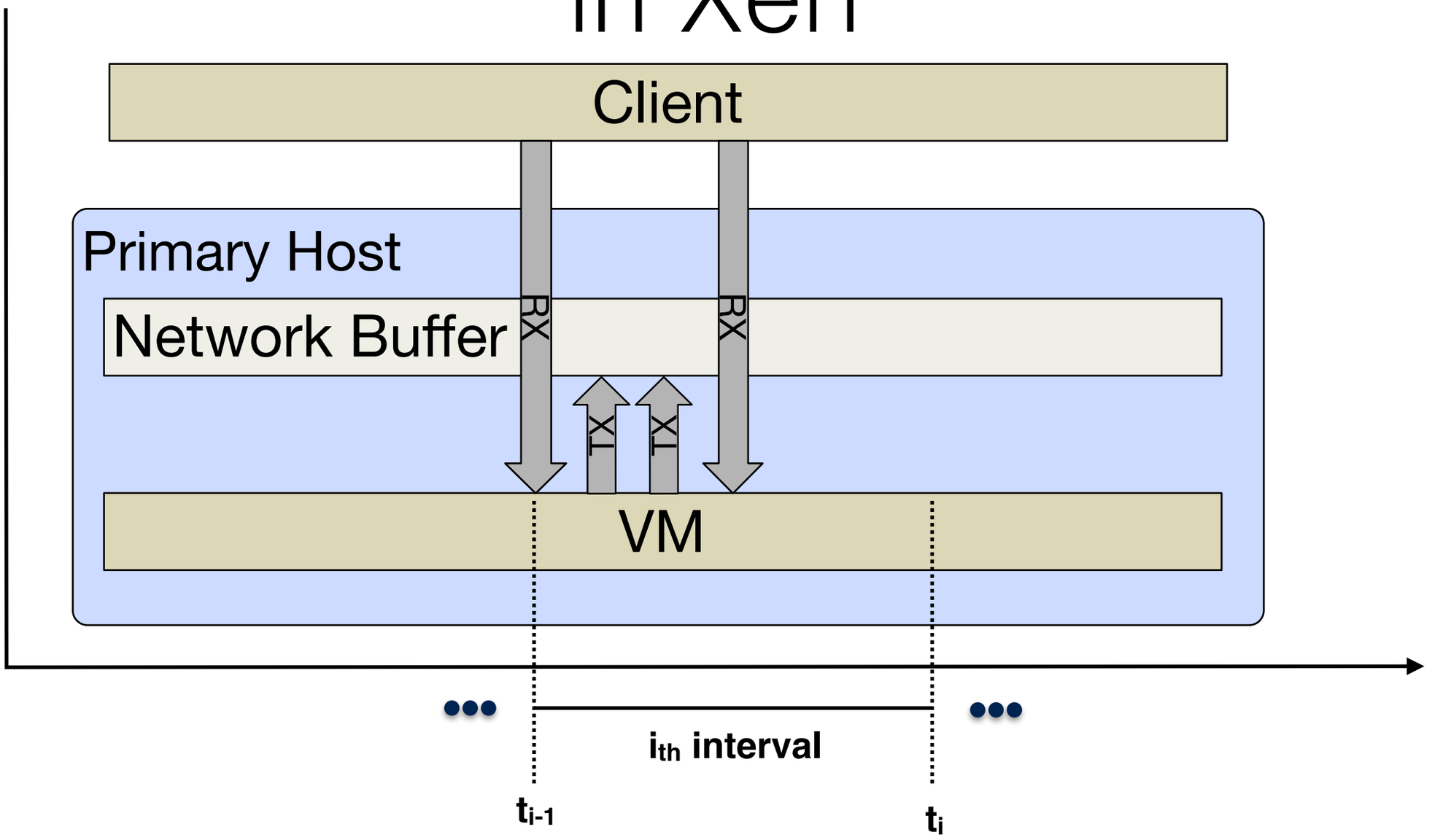
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



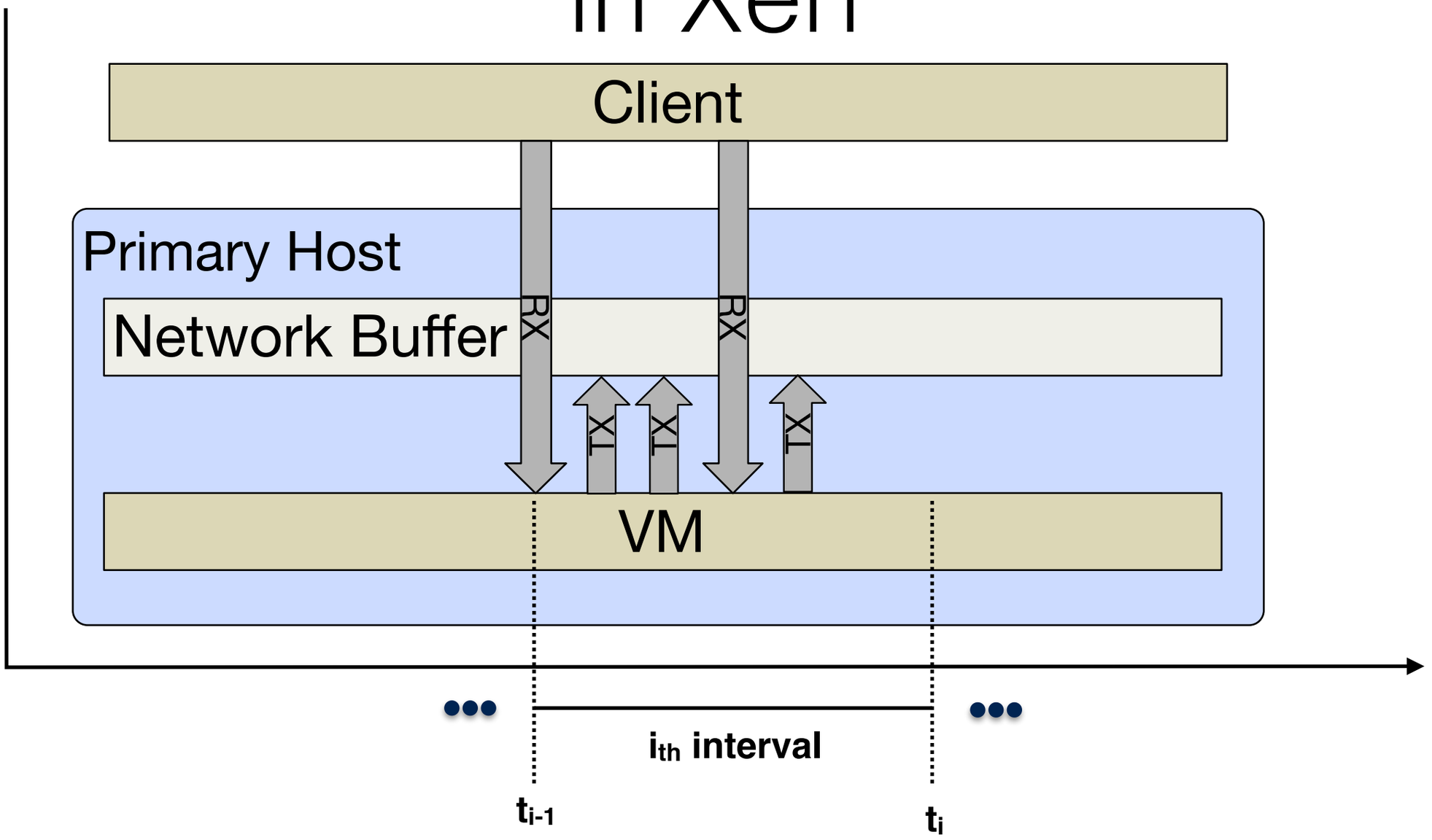
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



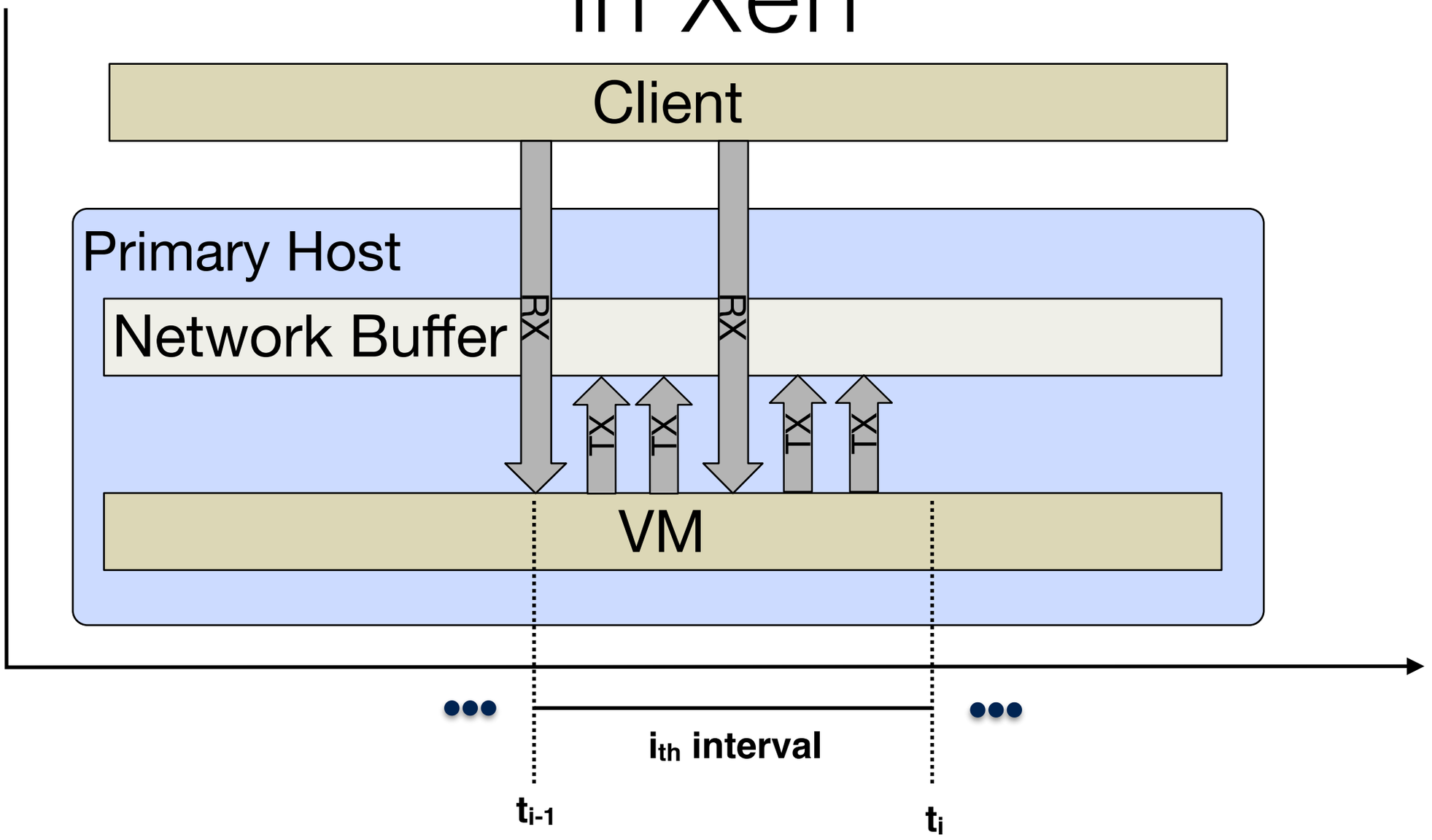
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



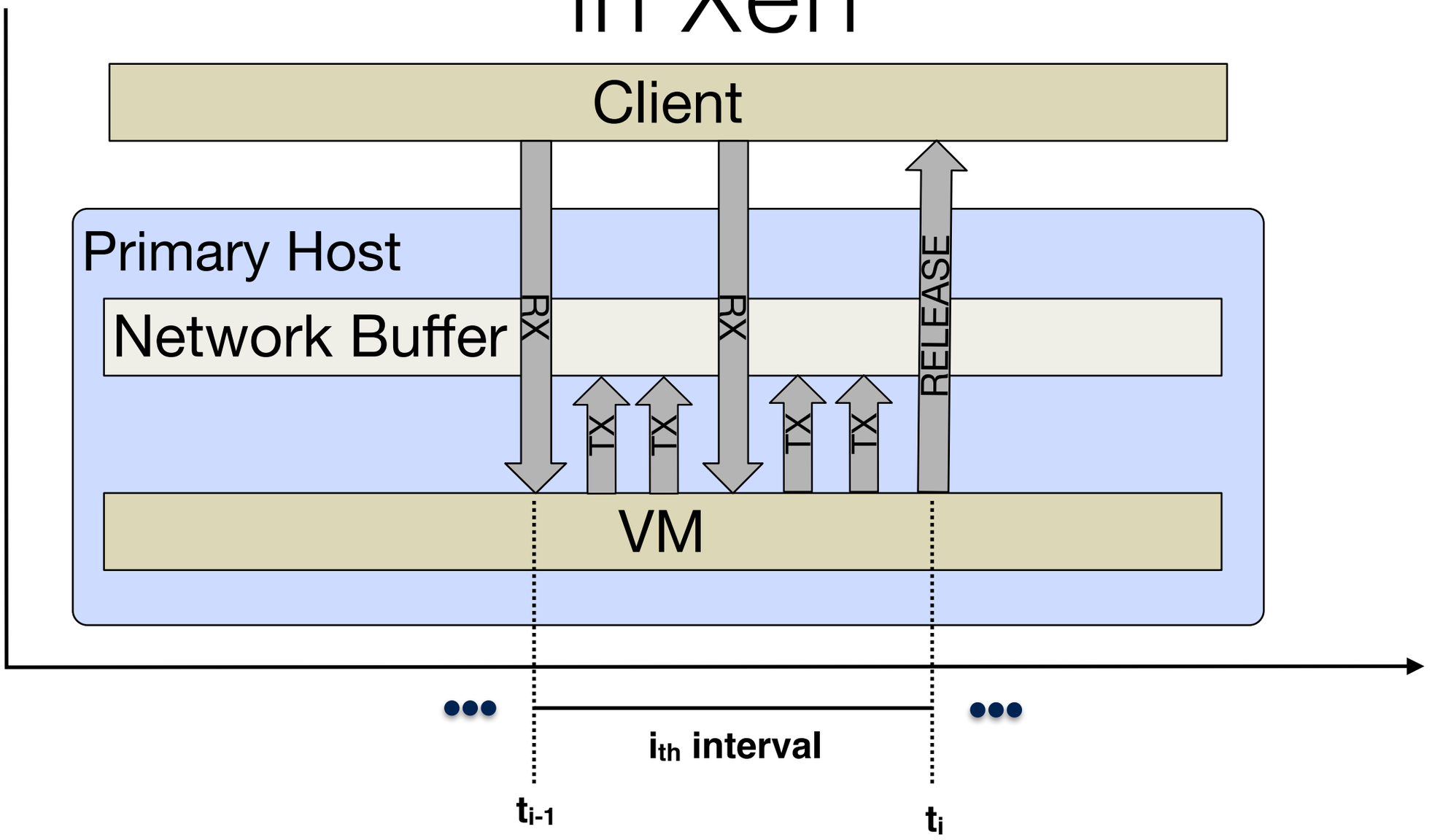
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



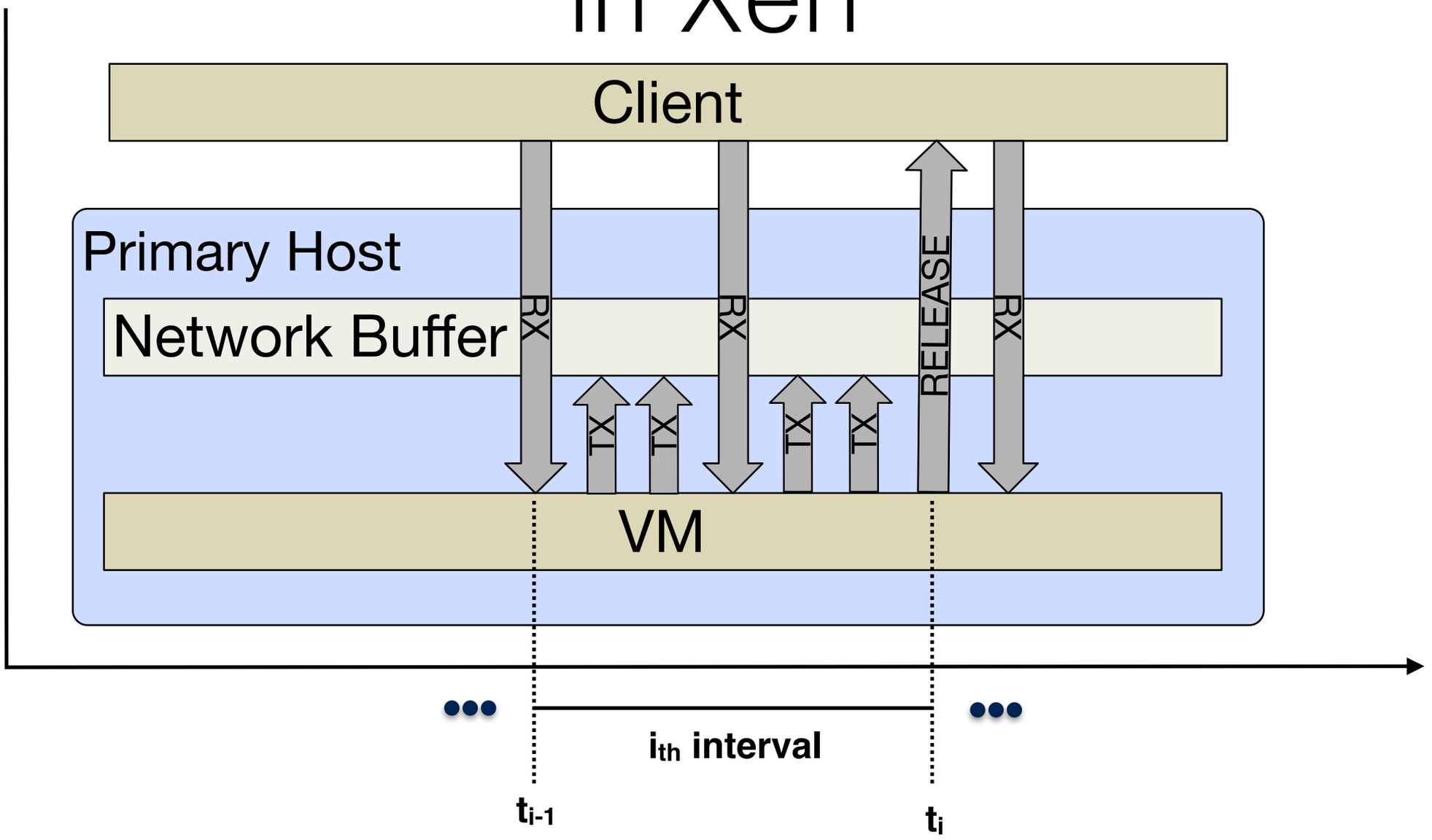
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



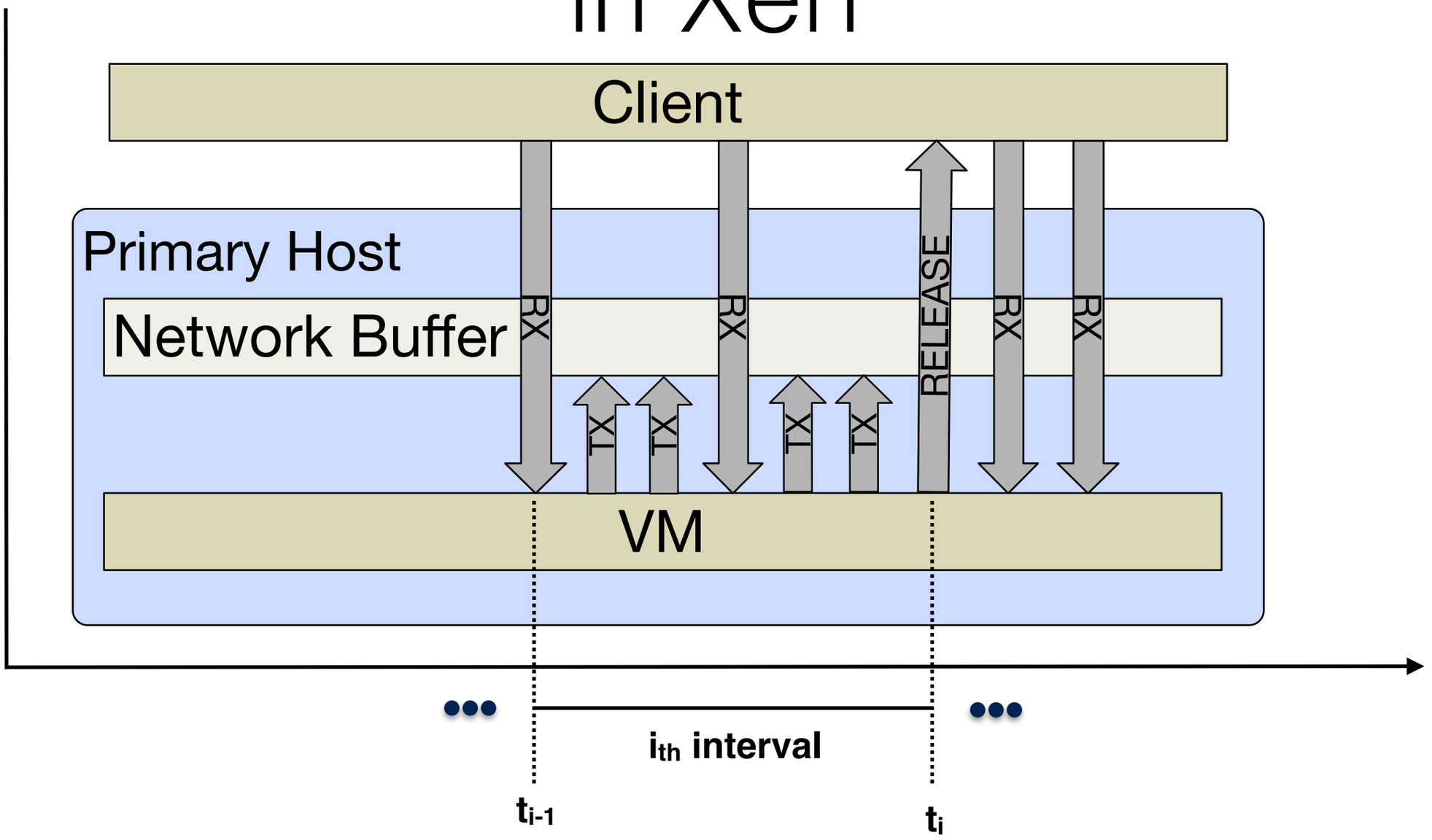
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



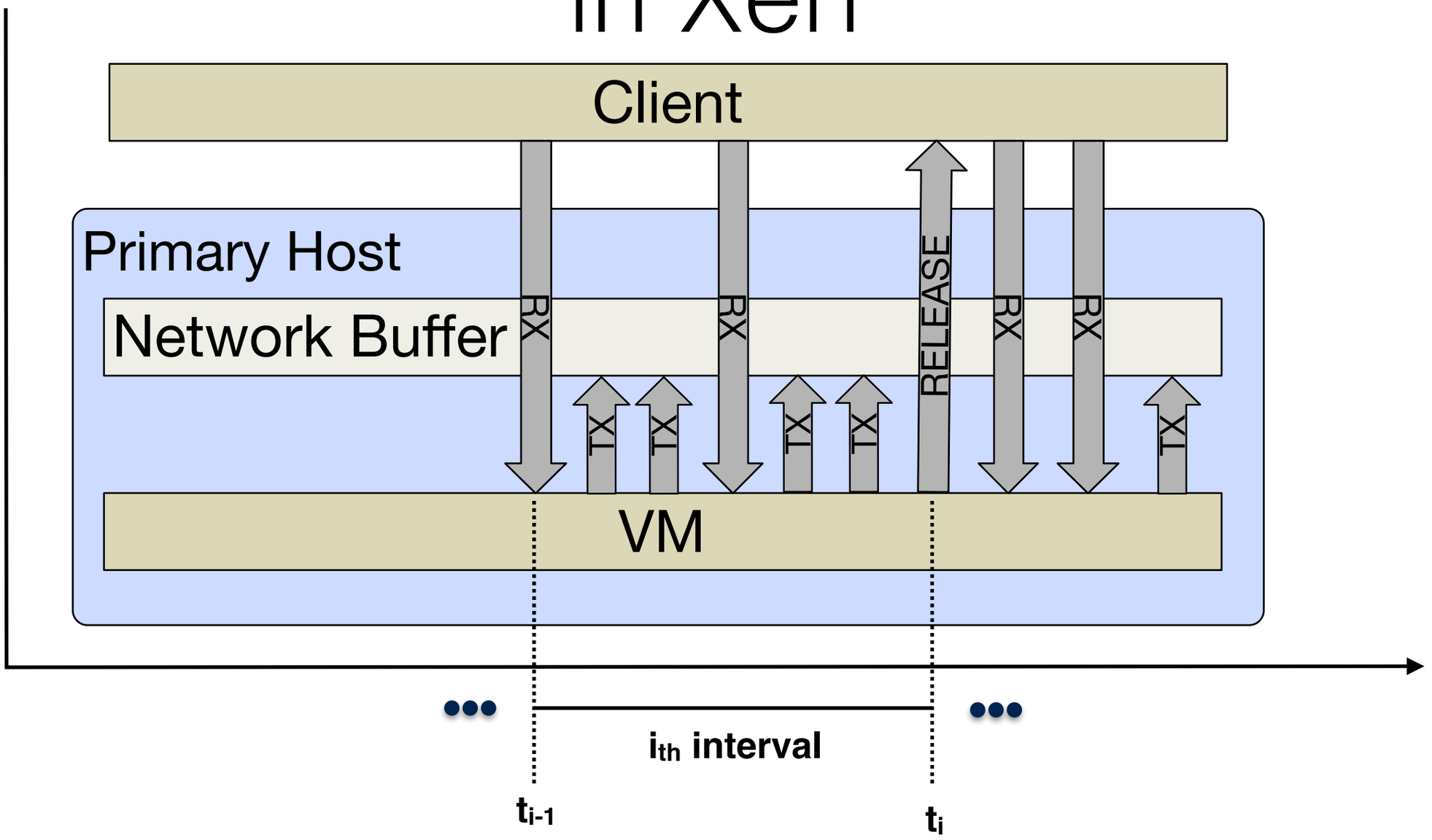
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

Network buffering using Remus in Xen



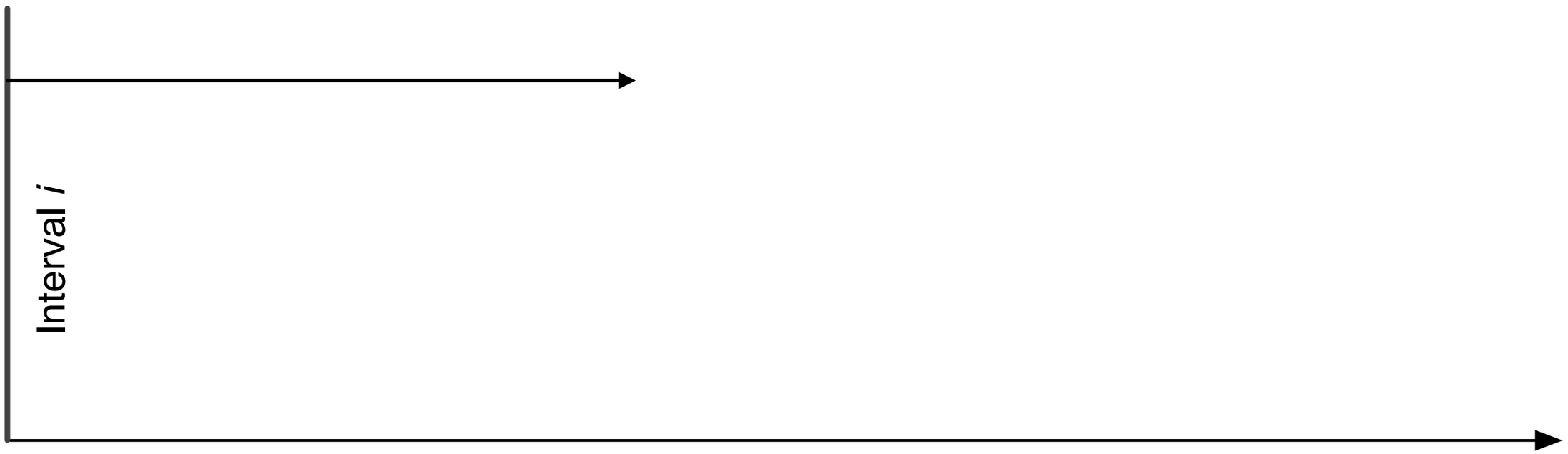
- All network packets are buffered for each interval.
- The buffer content is released only at the end of the interval.

ScaaS Execution

ScaaS Execution Timeline

- Pause briefly at each checkpoint to be scanned for security vulnerabilities.
- ScaaS says if it is safe to release the buffer.
- If an attack is found, the VM can be rolled back and analyzed.

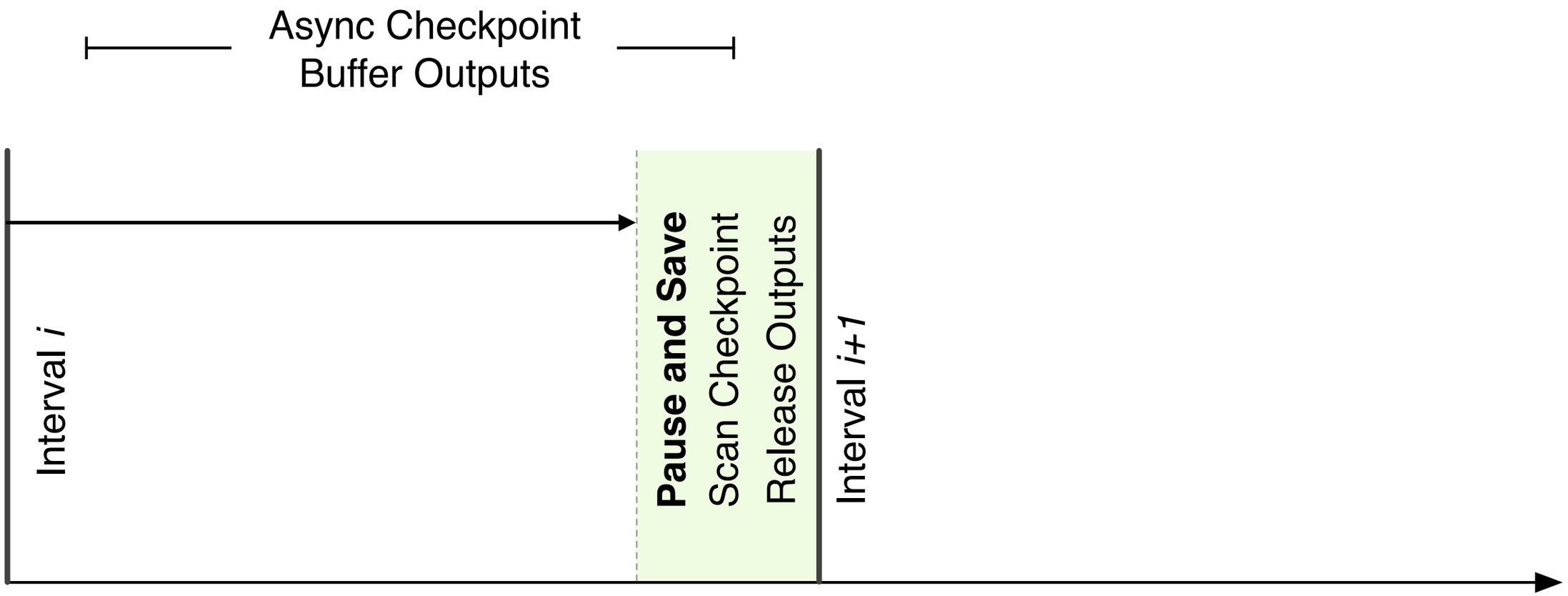
ScaaS Execution



ScaaS Execution Timeline

- Pause briefly at each checkpoint to be scanned for security vulnerabilities.
- ScaaS says if it is safe to release the buffer.
- If an attack is found, the VM can be rolled back and analyzed.

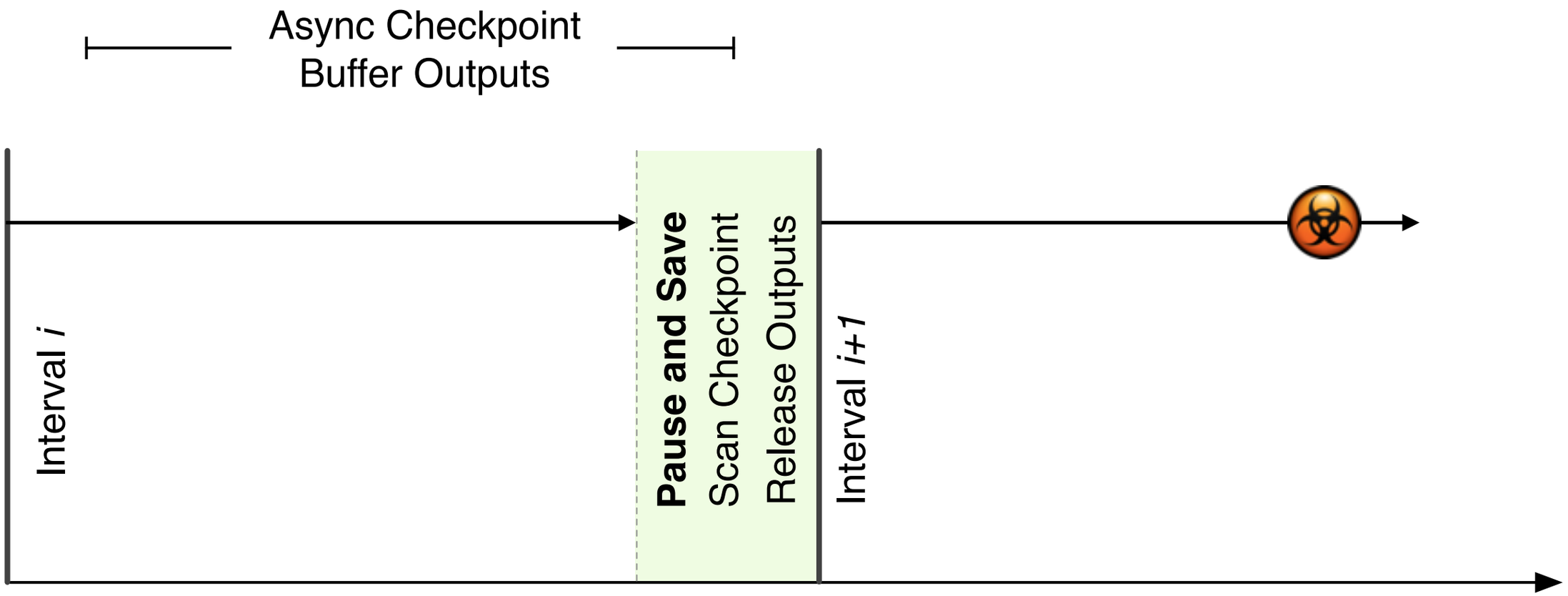
ScaaS Execution



ScaaS Execution Timeline

- Pause briefly at each checkpoint to be scanned for security vulnerabilities.
- ScaaS says if it is safe to release the buffer.
- If an attack is found, the VM can be rolled back and analyzed.

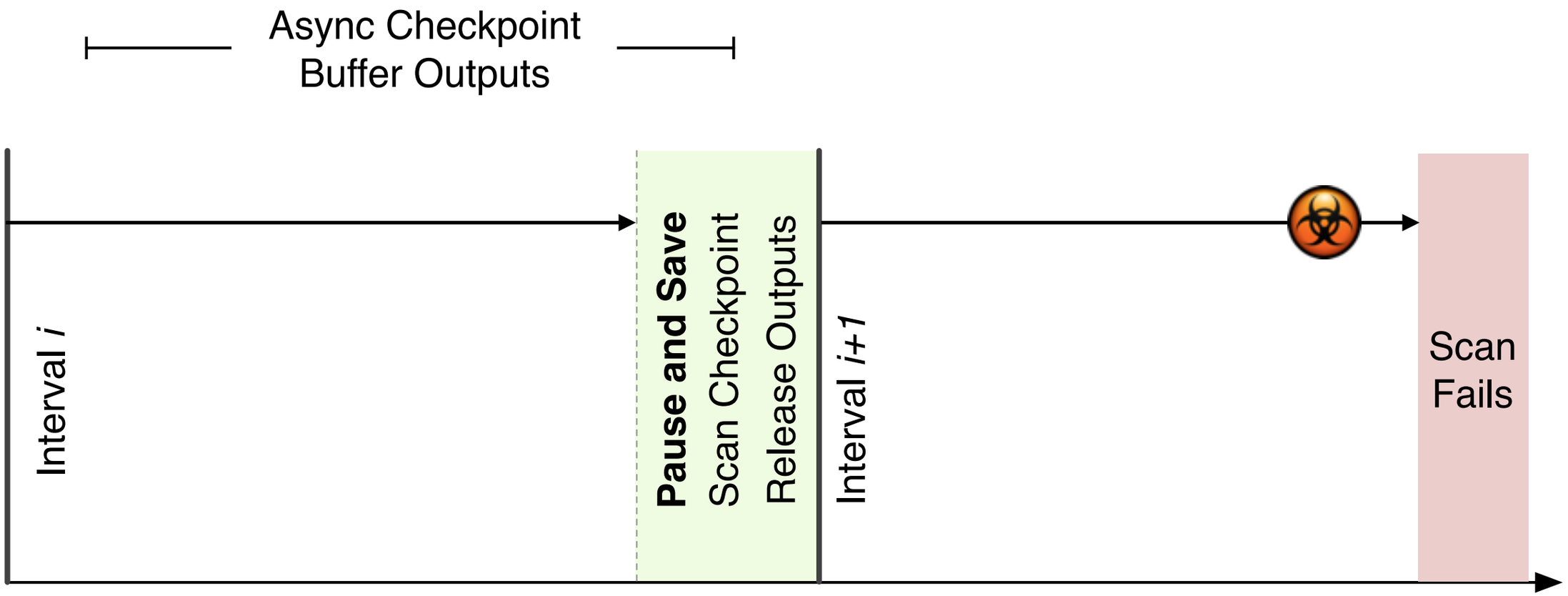
ScaaS Execution



ScaaS Execution Timeline

- Pause briefly at each checkpoint to be scanned for security vulnerabilities.
- ScaaS says if it is safe to release the buffer.
- If an attack is found, the VM can be rolled back and analyzed.

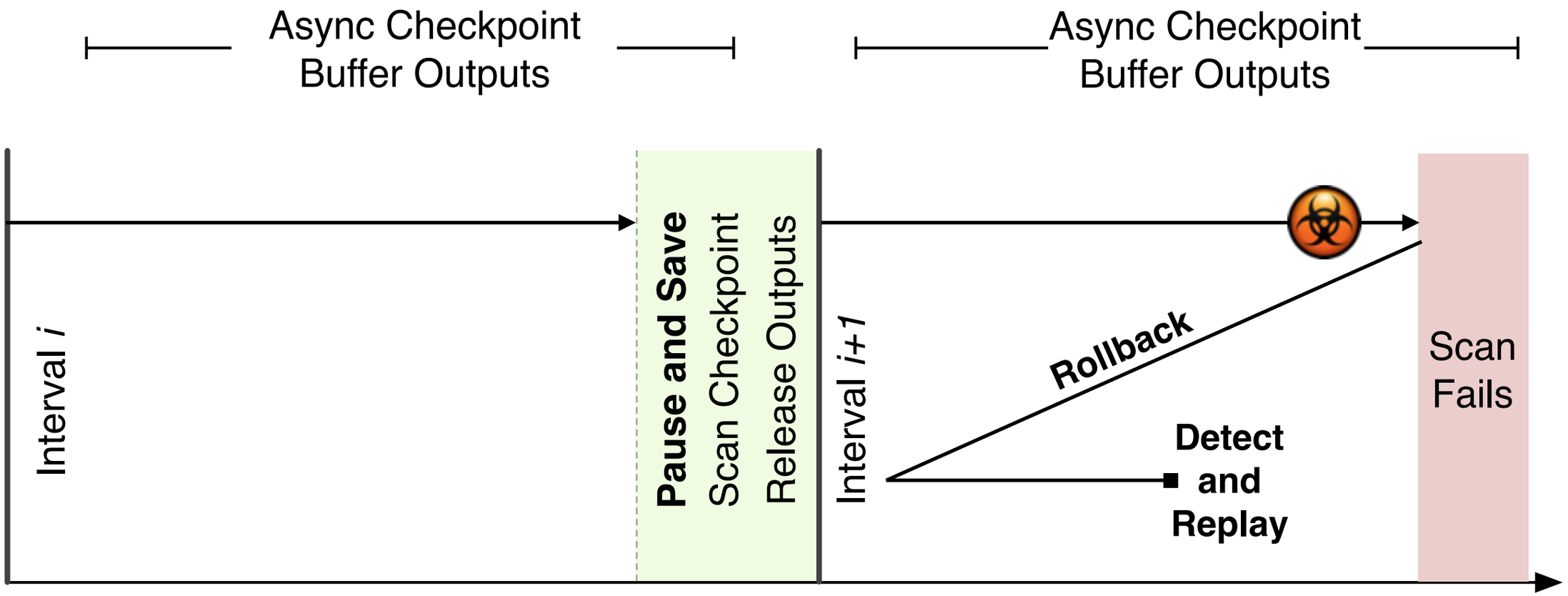
ScaaS Execution



ScaaS Execution Timeline

- Pause briefly at each checkpoint to be scanned for security vulnerabilities.
- ScaaS says if it is safe to release the buffer.
- If an attack is found, the VM can be rolled back and analyzed.

ScaaS Execution



ScaaS Execution Timeline

- Pause briefly at each checkpoint to be scanned for security vulnerabilities.
- ScaaS says if it is safe to release the buffer.
- If an attack is found, the VM can be rolled back and analyzed.

Attack Detection and Response

- **Forensic analysis:** Do analysis that cannot be done on runtime.

- **Rollback and Replay:** Useful when using breakpoints that trigger errors such as buffer overflow.

- **Honeypot mode:** Resume and run in a sandbox.

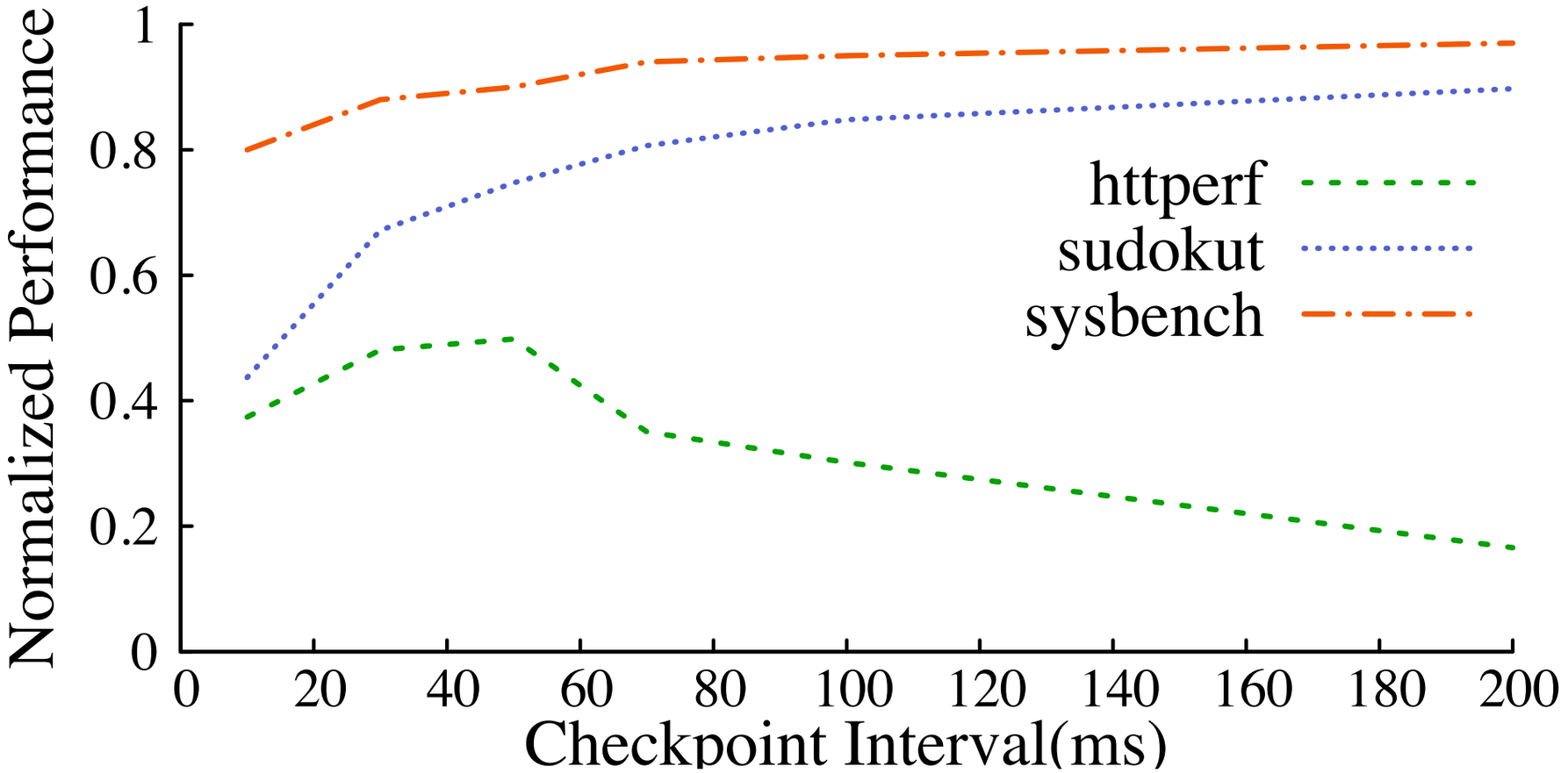
Prototype Evaluation

- Prototype of ScaaS using Xen 4.5.2
- 1Gbps link between Primary and Scanner host.
- Checkpointing using Remus.
- VM introspection using libVMI.

Types of Scans

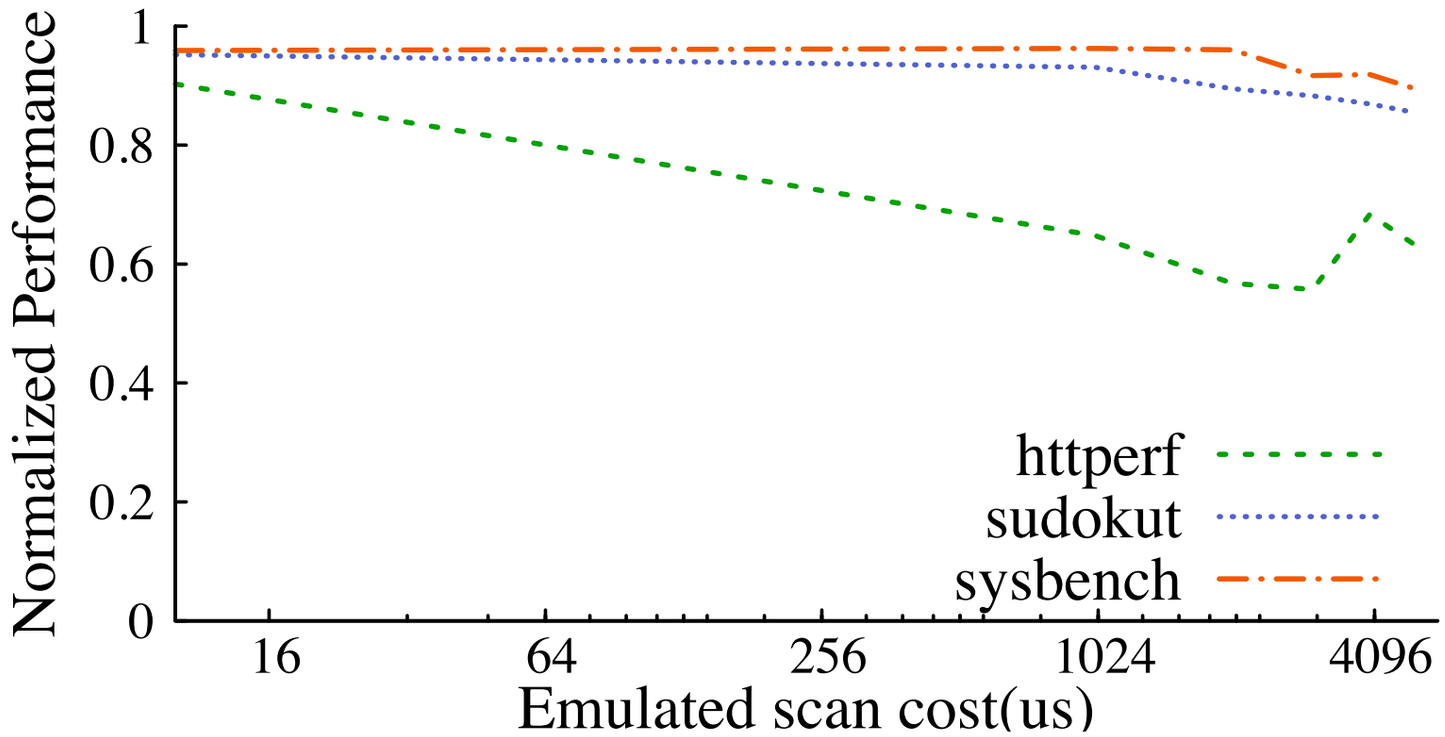
- **Process Black/White List Enforcer**:
    - Determines current running processes in a VM.Triggers errors depending on whether a target process is running or not.
- **Memory Fingerprinter:**
    - Hashes the memory pages to compare against known good states. eg: sys call table, that doesn’t change that often.

Checkpoint overhead



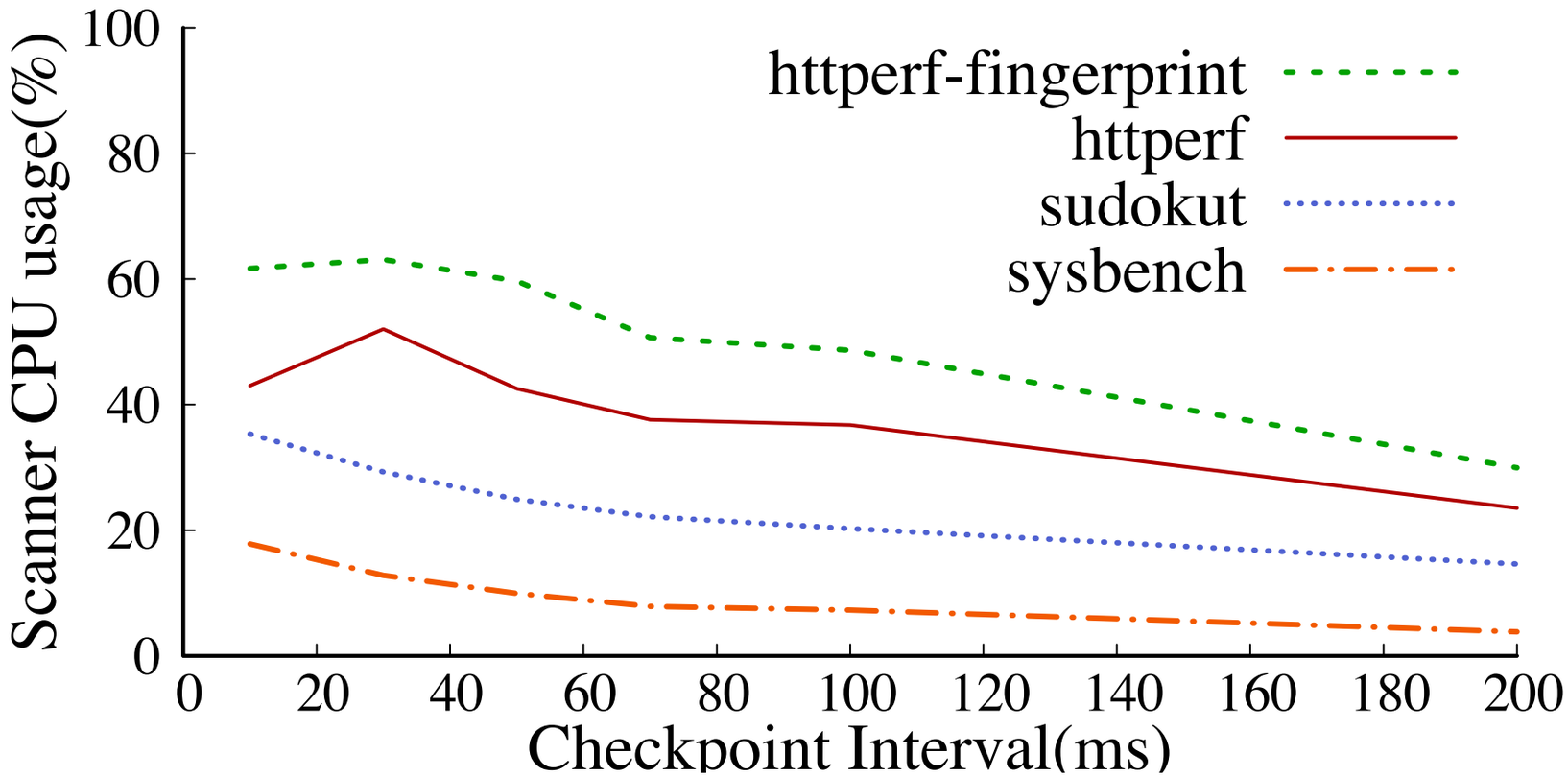
- Benchmarks vs. different checkpoint intervals
- CPU intensive benchmarks perform well with longer intervals
- httperf is a latency sensitive benchmark
  - Longer the interval worse the performance.

Emulated Scan cost



- Performance change of application w.r.t. emulated scan costs.
- Normalized wrt to zero-cost scan
- httperf costs worsens with scan cost
  - as it has to hold buffer data for longer periods

CPU usage at scanner host



- Fingerprinter causes high overhead initially but becomes negligible as checkpointing interval increase.

Conclusion

- ScaaS: Framework for security Scanning as a Service.
- Tool for attack detection and forensic analysis on memory.
  - examining memory checkpoints for an attack.
  - highly scalable and fast.

Discussion

- What types of attacks can we detect?
- Do we need to keep a history of checkpoints? Why? How?
- What is a reasonable cost for ScaaS?