Exploring VM Introspection: Techniques and Trade-offs

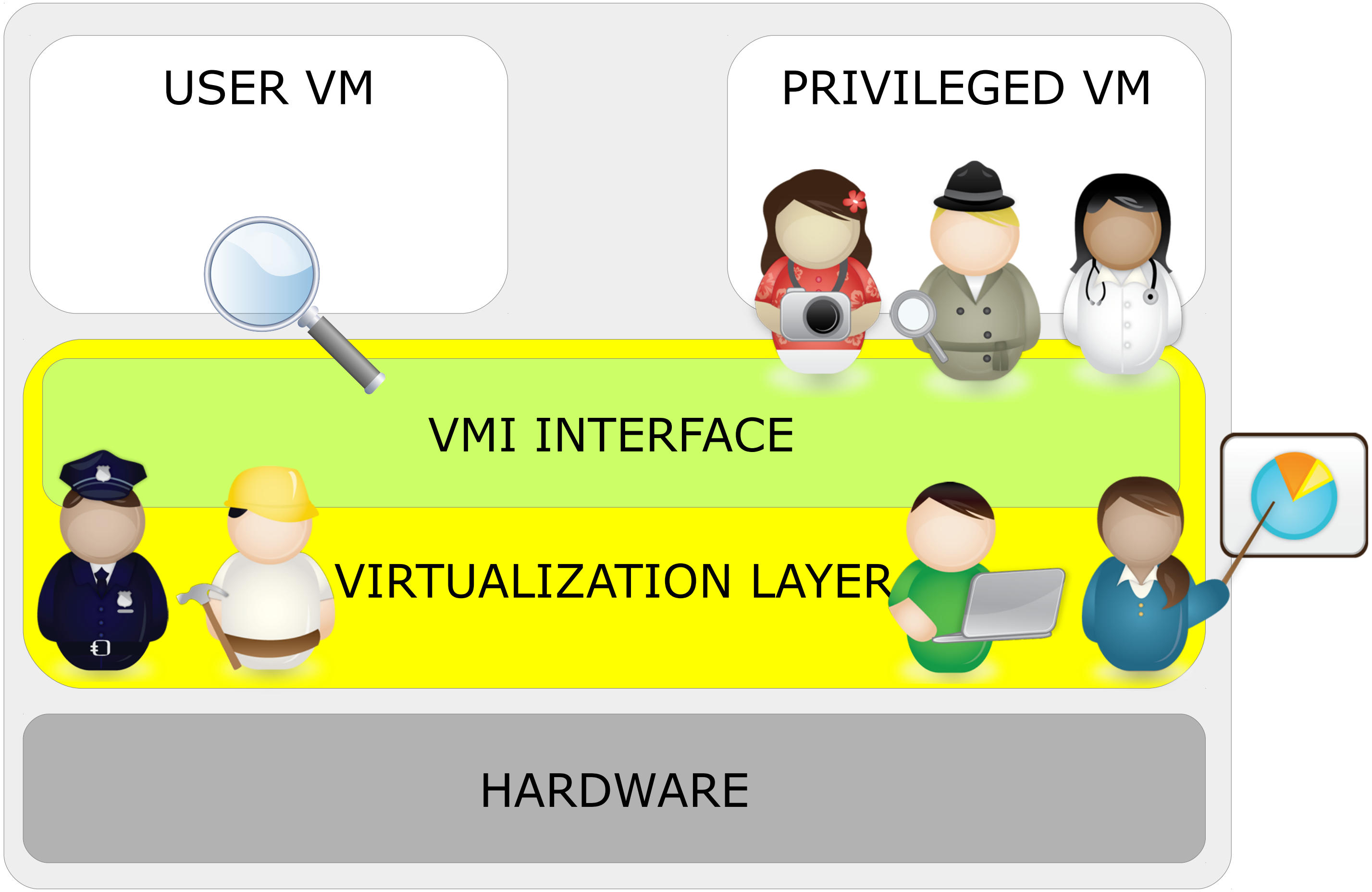
VEE

March 15, 2015

Sahil Suneja
Eyal de Lara
University of Toronto

Canturk Isci
Vasanth Bala
IBM T.J. Watson Research

Virtual Machine Introspection: Uses



VMI Techniques - Expose VM State

XEN

`XenGuestAgent`
`dump-core`
`xc_map_foreign_range()`
`gdbsx`
`...`

QEMU/KVM

`pmemsave`
`migrate`
`hugetlbfs`
`/proc/pid/mem`
`...`

VMWare

`VMSafe()`
`vmss2core`
`.vmem`
`...`

LIBRARY / 3rd PARTY

Libvirt
LibVMI
`...`

Technique Vs. Technique?

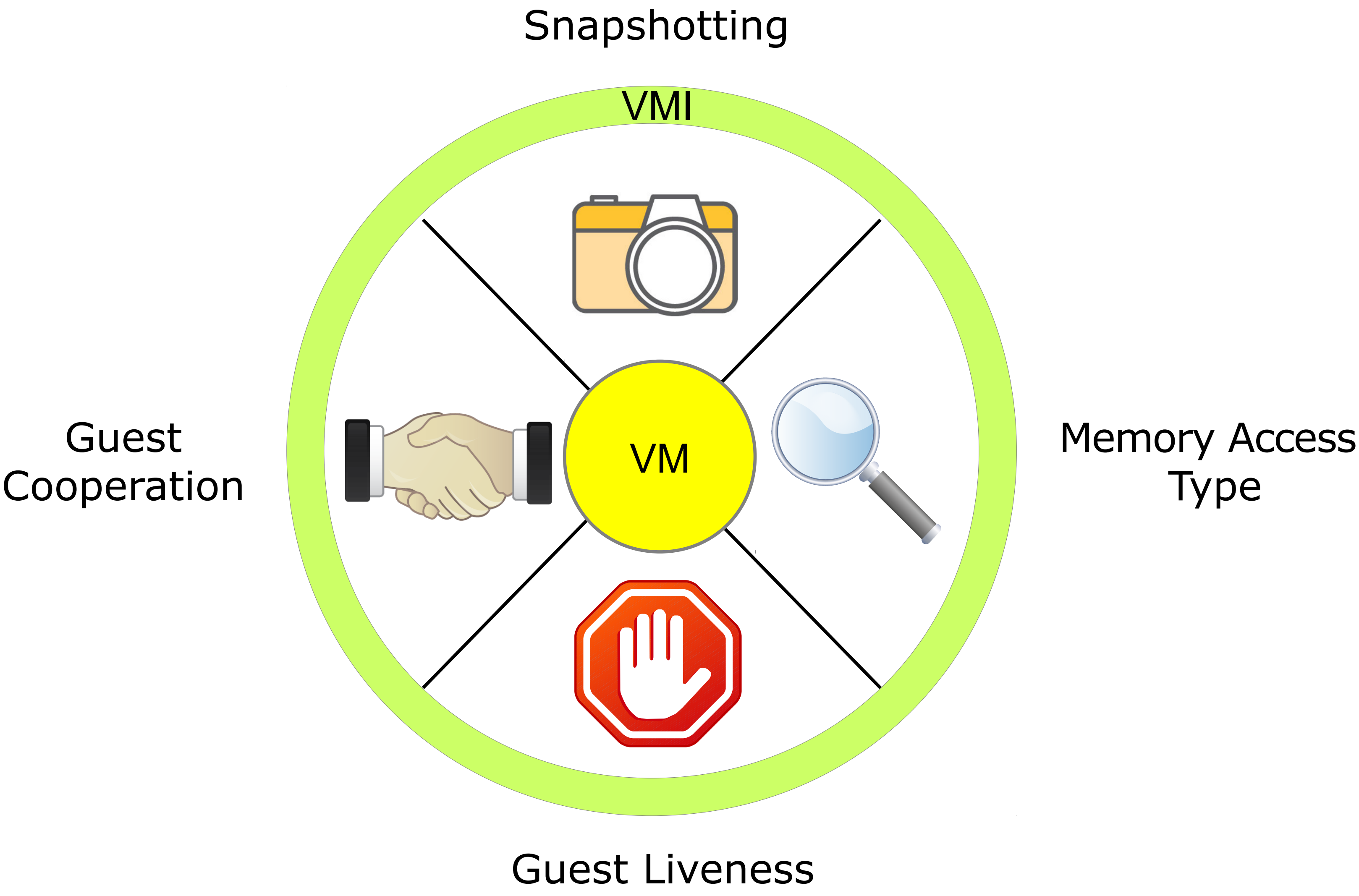
- Latency, overhead, complexity, …

Qualitative Comparison

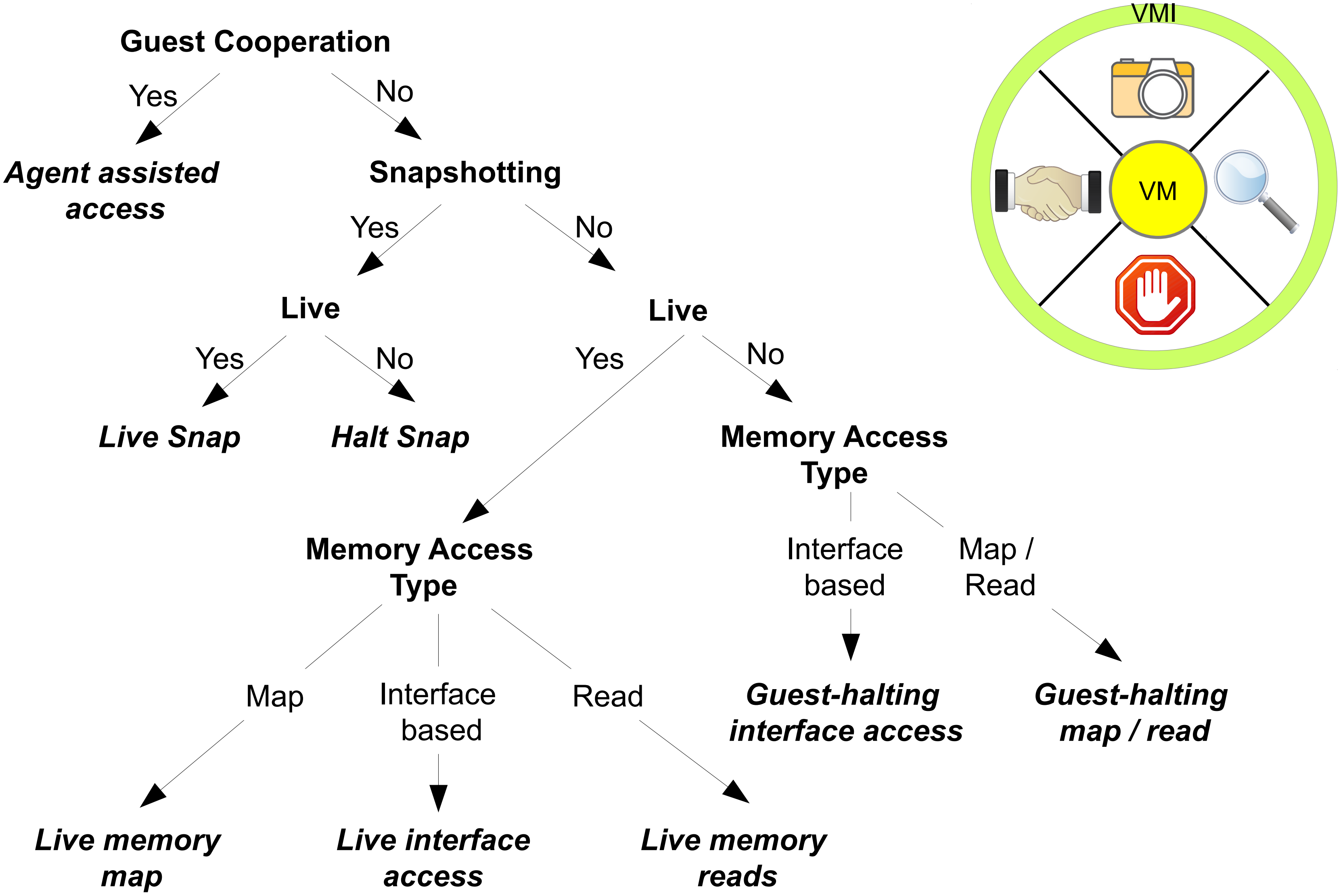
+ Hypervisor / Host Compatibility

Properties Techniques	Live	View Consistency	Speed	Resource Cost	VM perf impact
Guest cooperation / agent assisted access	✓	✓	🟨🟨🟨	🟦🟦🟦	🟥🟥
Halt Snap		✓	🟨🟨	🟦🟦🟦🟦	🟥🟥🟥🟥
Live Snap	✓	✓	🟨🟨🟨	🟦🟦	🟥🟥
Live Memory Mapping	✓		🟨🟨🟨🟨🟨	🟦	🟥
Live Memory Reads	✓		🟨🟨🟨🟨	🟦🟦	🟥
Guest-Halting Memory Map and Reads		✓	🟨🟨🟨	🟦🟦	🟥🟥🟥
Live Interface Access	✓		🟨	🟦🟦🟦🟦🟦	🟥🟥
Guest Halting Interface Access		✓	🟨	🟦🟦🟦🟦🟦	🟥🟥

Characterizing VMI Techniques: Dimensions



Proposing VMI Taxonomy



- **Guest Cooperation**
  - Yes → ***Agent assisted access***
  - No → **Snapshotting**
    - Yes → **Live**
      - Yes → ***Live Snap***
      - No → ***Halt Snap***
    - No → **Live**
      - Yes → **Memory Access Type**
        - Map → ***Live memory map***
        - Interface based → ***Live interface access***
        - Read → ***Live memory reads***
      - No → **Memory Access Type**
        - Interface based → ***Guest-halting interface access***
        - Map / Read → ***Guest-halting map / read***

Quantitative Comparison: Use-case

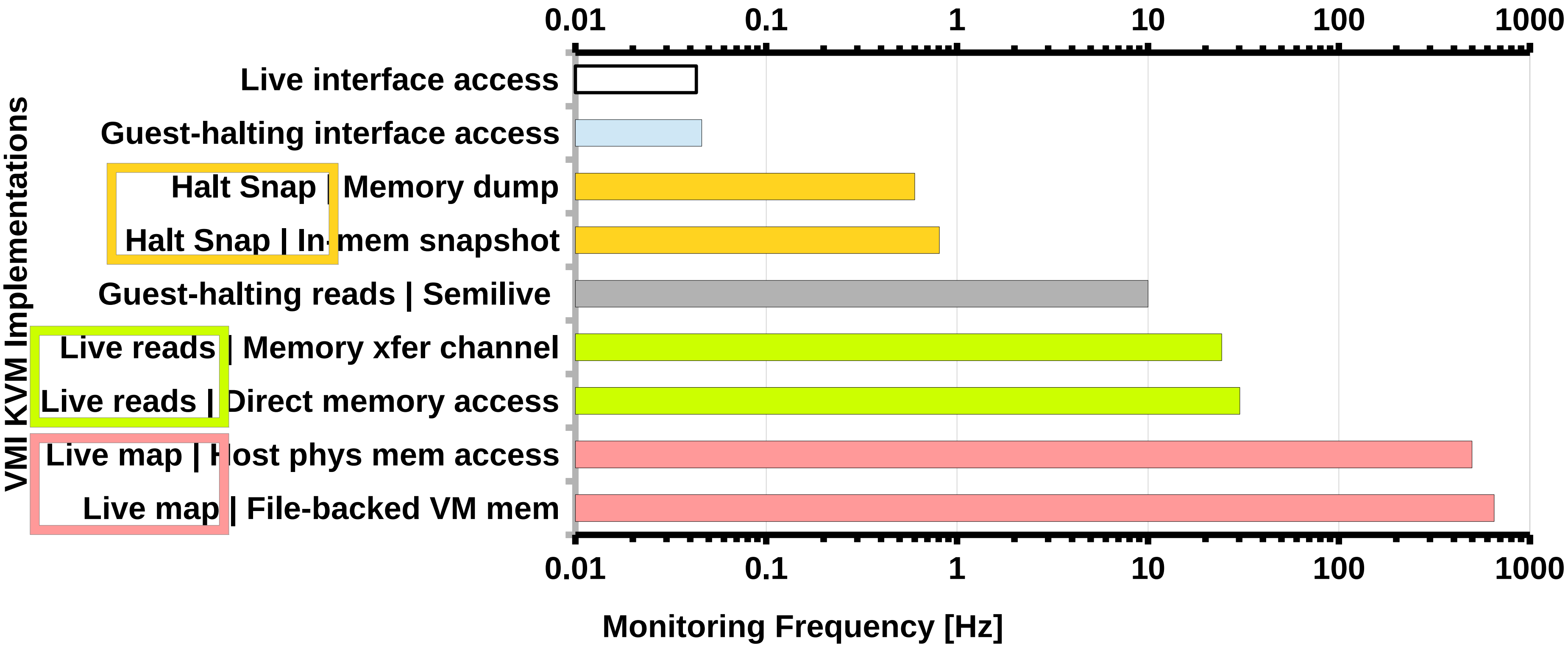
- Periodic generic monitoring via VMI
  - In-memory kernel data structure traversal
    - `task_struct, mm_struct, files_struct, net_devices ...`
  - Extract 700KB volatile VM memory state

CPU	NumCores, Hz, CacheSize, …
OS	Nodename, Release, Arch, …
N/W device	HWaddr, Ipaddr, TX/RX bytes, …
Modules	Name, State, …
Process	PID, Command, RSS, …
Open files	FD → filename, …
Memory Mapping	MappedFiles, VA → PA mappings, …
N/W connections	SocketState, {Src, Dst, Ports}, …

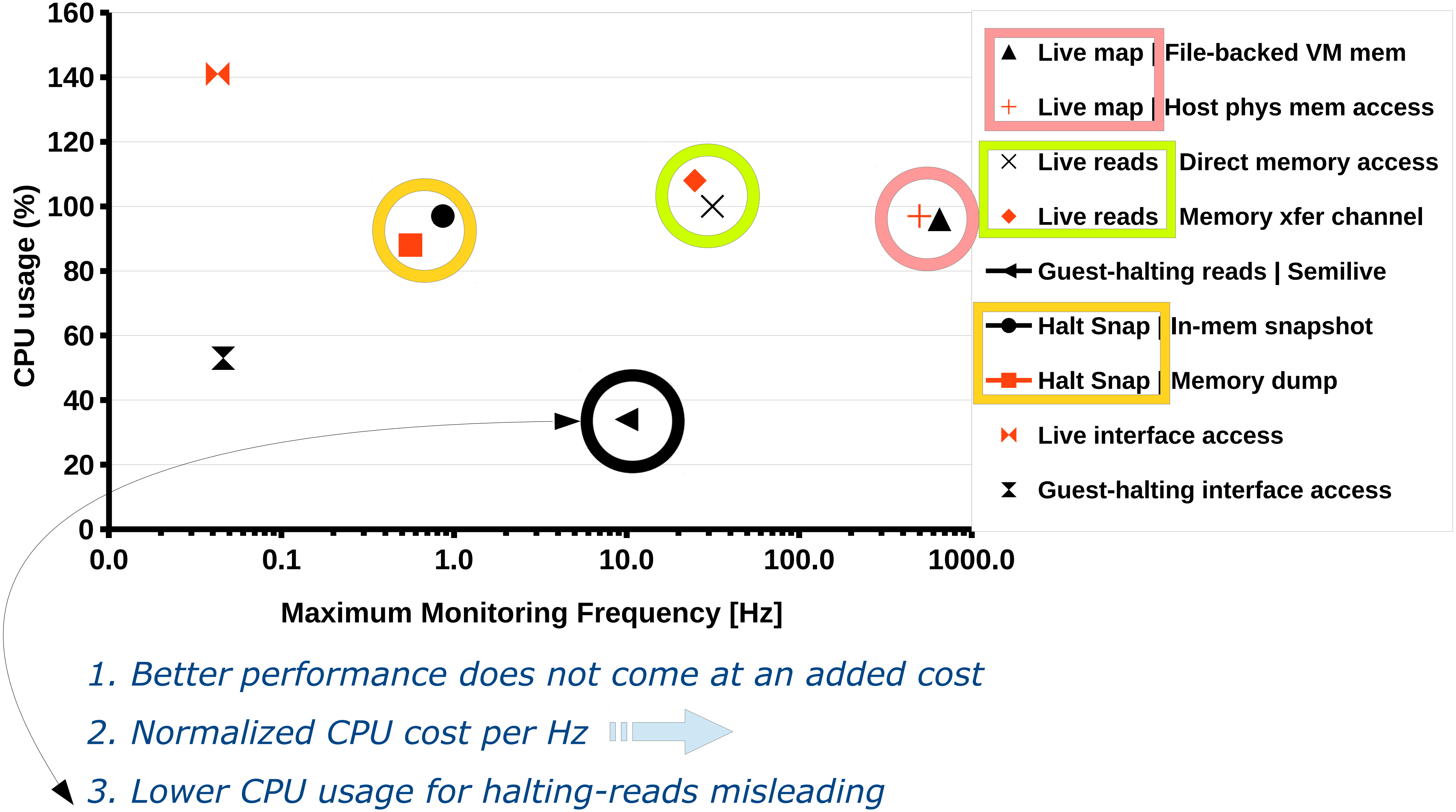
Quantitative Comparison: Dimensions

- Maximum frequency of monitoring?
- Resource usage cost on host?
- Impact on VM’s workload?

Maximum Monitoring Frequency



Resource Cost on Host

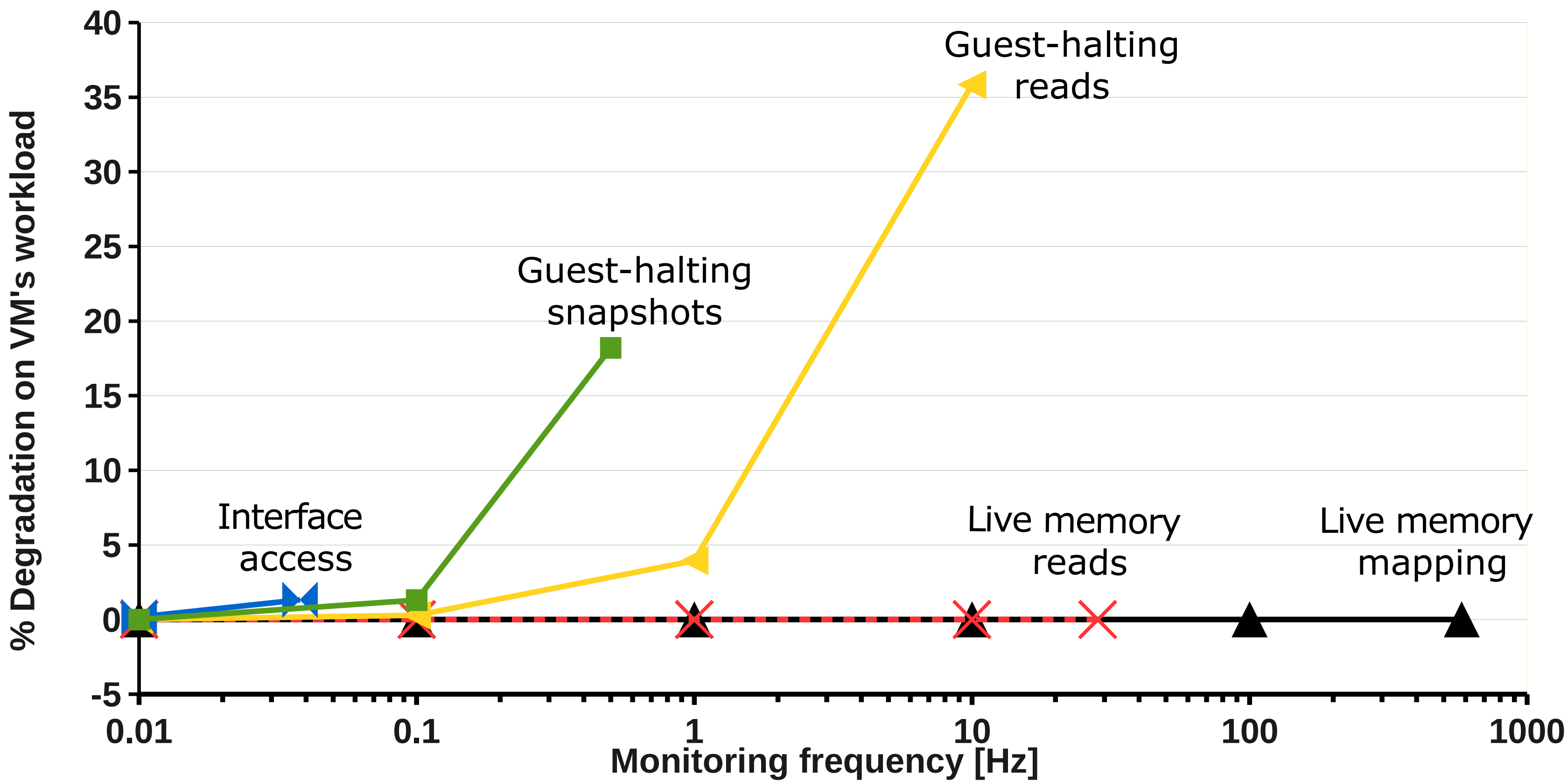


1. *Better performance does not come at an added cost*

2. *Normalized CPU cost per Hz*

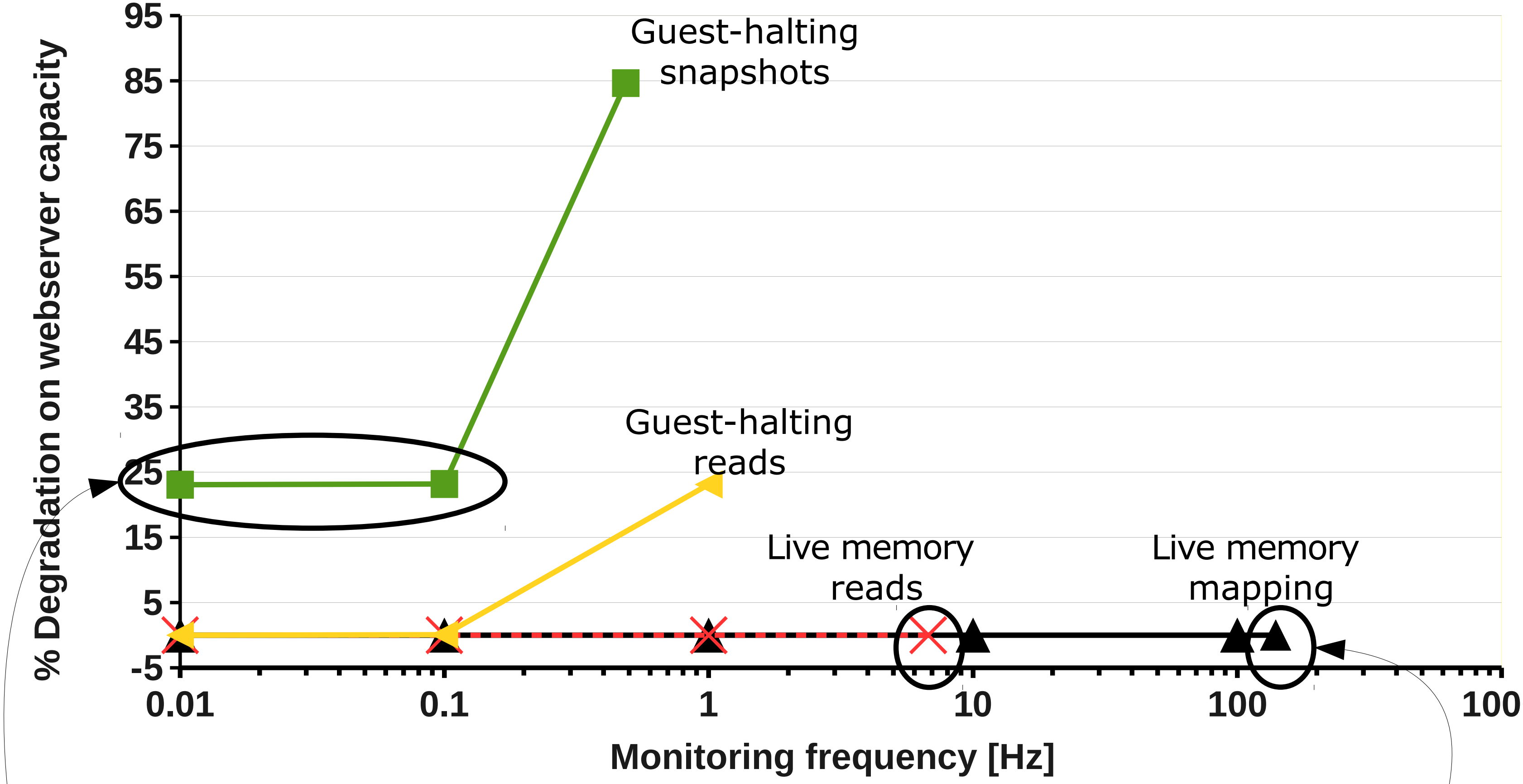
3. *Lower CPU usage for halting-reads misleading*

Impact on VM Perf: x264 CPU Benchmark [1/2]



+ Individual techniques' performance
+ VM impact vs. Technique resource allotment
+ Memory, N/W, Disk benchmarks

Impact on VM’s Perf: httperf Webserver [2/2]



1. *Lower maximum monitoring frequencies:* `apache` *=> in-VM state*

2. *Backlog of pending requests in wait queue*

3. *>100% degradation on response times with halt snap*

} + in paper

Part II: Consistency of VM State

Consistency of VM State [1/3]

- Missing OS-context within VMI scope
  - Inconsistency in observed data structures
- Common solution: pause-and-introspect (PAI)
- Goal:
  - What are these inconsistencies?
  - How ofter do these occur?
  - Is PAI helpful?

Consistency of VM State [2/3]

- Observation #1: Multiple forms of inconsistencies

Intrinsic Inconsistencies

Zombie tasks

Dying tasks

As-good-as-dead tasks

Fresh tasks

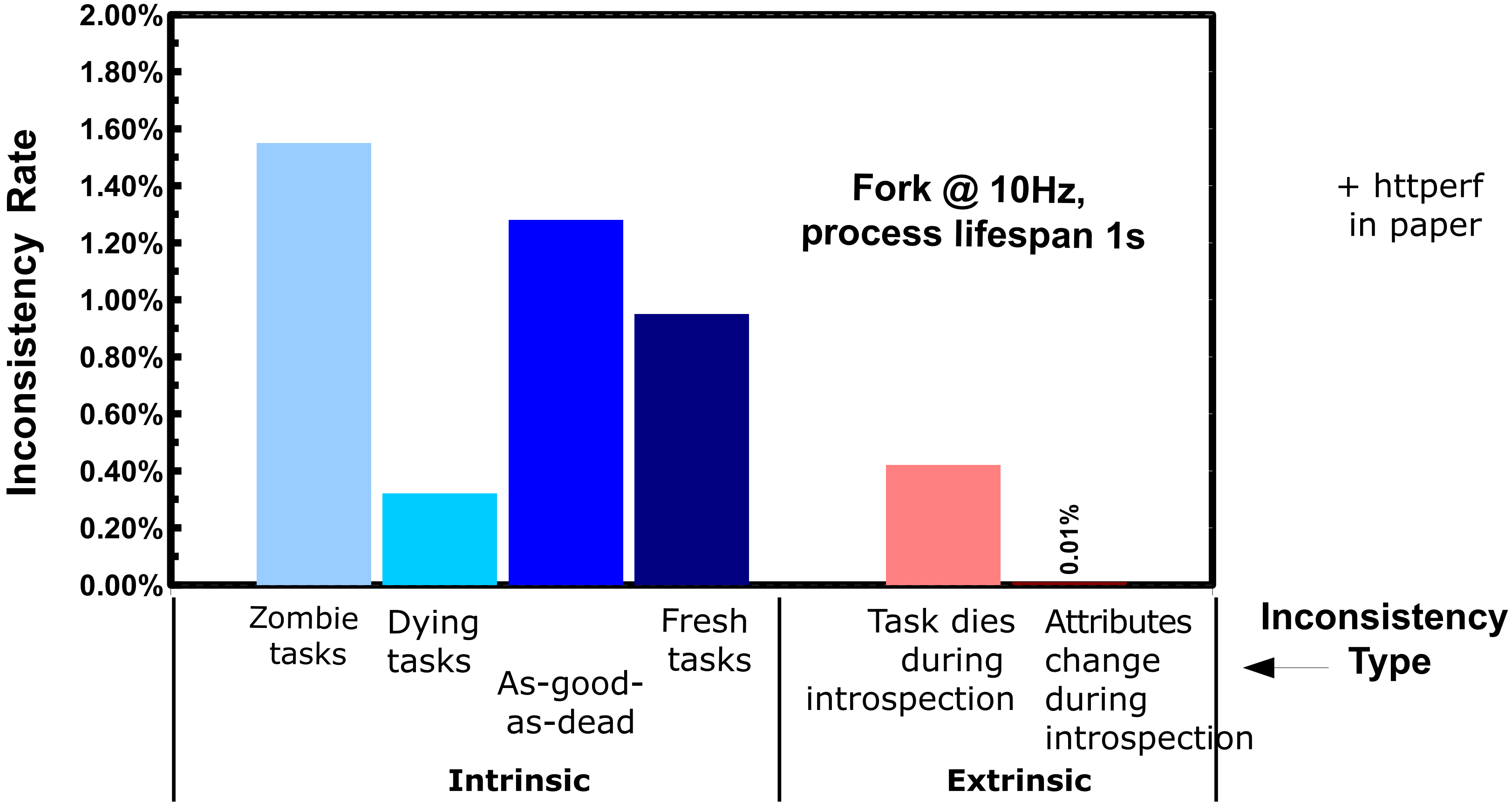
Extrinsic Inconsistencies

Task dies during introspection

Attributes change during introspection

Consistency of VM State [3/3]

- Observation #2: Inconsistencies are rare



+ httperf in paper

- Observation #3: Expensive PAI does not mitigate all inconsistencies
  - Intrinsic remain

Conclusion

- Taxonomy to organize existing VMI techniques
  - Guest Cooperation, Snapshotting, Guest Liveness, Memory Access Type
- Comparative evaluation of techniques
  - Quantitative: Operating frequencies, Resource consumption on host, and Overheads on target systems.
  - + Qualitative: Liveness, Consistency, Compatibility, …
- Memory inconsistency analysis
  - Multiple forms of inconsistencies
  - Inconsistencies are rare
  - PAI have marginal benefits, despite high cost

BACKUP SLIDES -->

Qualitative Comparison

+ Hypervisor / Host Compatibility

Properties Techniques	Live	View Consistency	Speed	Resource Cost	VM perf impact
Guest cooperation / agent assisted access	✓	✓	🟨🟨🟨	🟦🟦🟦	🟥🟥
Halt Snap		✓	🟨🟨	🟦🟦🟦🟦	🟥🟥🟥🟥
Live Snap	✓	✓	🟨🟨🟨	🟦🟦	🟥🟥
Live Memory Mapping	✓		🟨🟨🟨🟨🟨	🟦	🟥
Live Memory Reads	✓		🟨🟨🟨🟨	🟦🟦	🟥
Guest-Halting Memory Map and Reads		✓	🟨🟨🟨	🟦🟦	🟥🟥🟥
Live Interface Access	✓		🟨	🟦🟦🟦🟦🟦	🟥🟥
Guest Halting Interface Access		✓	🟨	🟦🟦🟦🟦🟦	🟥🟥

Observations and Recommendations

- Broad Spectrum of Choices
- Guest Cooperation vs. Out-of-band
- VMI use-case
- VM Workload
- Host / Hypervisor Specialization
- Mapping over direct reads
- Guest-halting map/reads over halting snapshots
- Consistency vs. Liveness, Realtimeness, and VM perf.
- Monitoring Overhead vs. Resource Usage
- Scalability of approaches