Fast switching between operating systems using a pass-through hypervisor

Sebastian Eydam

Brandenburg
University of Technology
Cottbus - Senftenberg

CYBERUS
TECHNOLOGY

Talking Points

1 Motivation

2 Approach

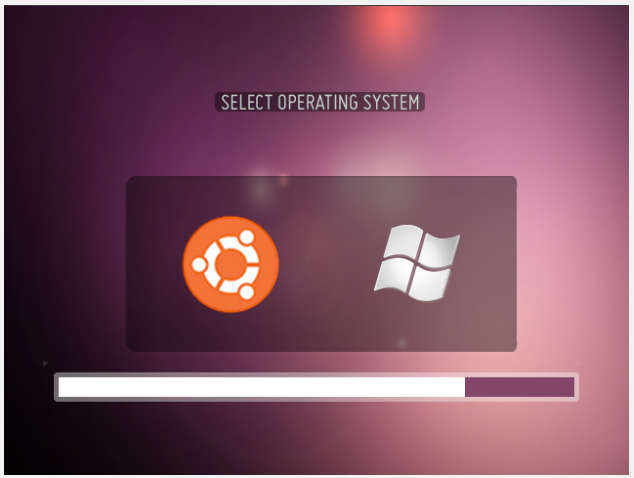
3 Measurements

4 Future Work

5 Conclusion

Motivation

Motivation



Motivation

- multiple OSes on one computer

Motivation

- multiple OSes on one computer
- fast switching without losing our state

Motivation

- multiple OSes on one computer
- fast switching without losing our state
- eliminate or mitigate performance issues

Motivation

- multiple OSes on one computer
- fast switching without losing our state
- eliminate or mitigate performance issues
- eliminate hardware issues

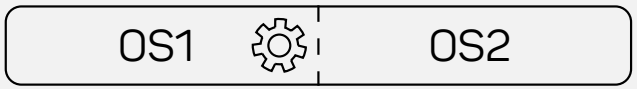
Motivation

- multiple OSes on one computer
- fast switching without losing our state
- eliminate or mitigate performance issues
- eliminate hardware issues
- do not modify the OSes

Approach

Approach

- to switch between multiple OSes, we need multiple OSes



Approach

- to switch between multiple OSes, we need multiple OSes

Approach

- to switch between multiple OSes, we need multiple OSes
- a virtualization layer grants us control over the hardware

Approach

- to switch between multiple OSes, we need multiple OSes
- a virtualization layer grants us control over the hardware
- OSes expect certain memory regions

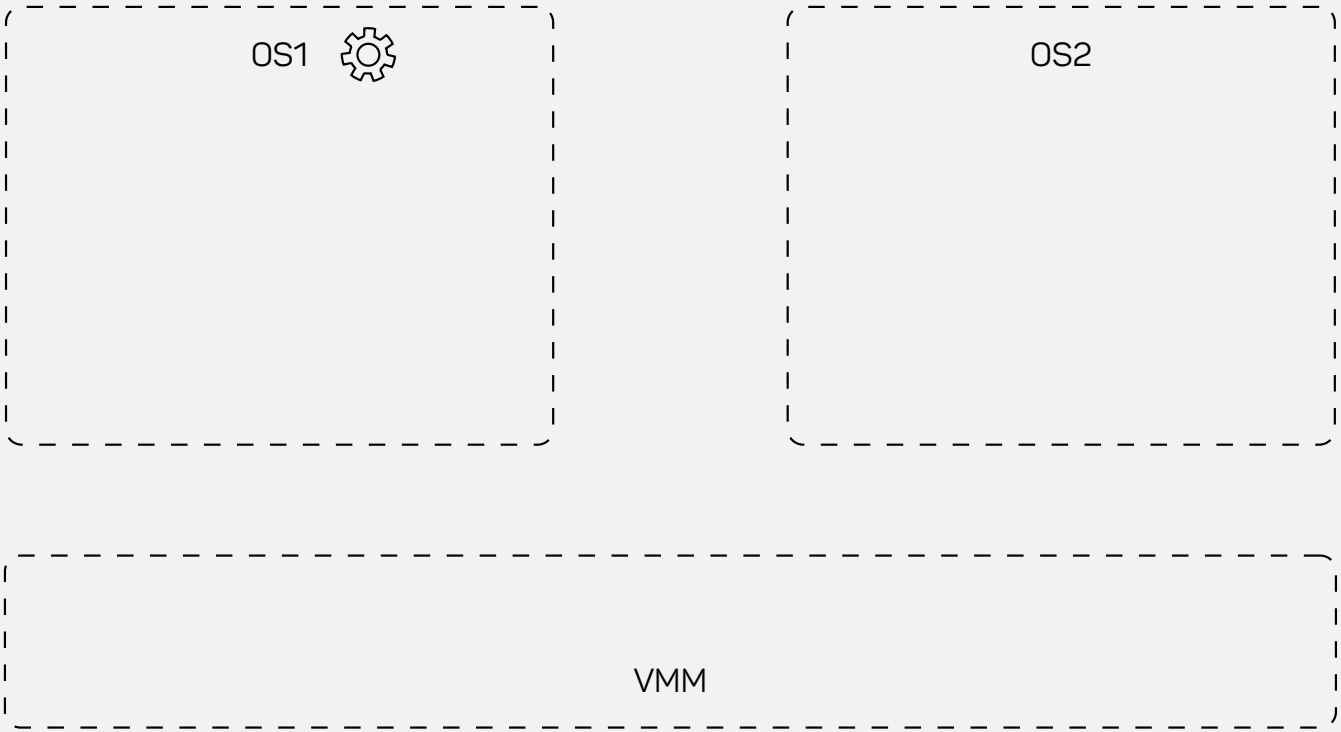
Approach

- to switch between multiple OSes, we need multiple OSes
- a virtualization layer grants us control over the hardware
- OSes expect certain memory regions
- the OSes have to handover their devices

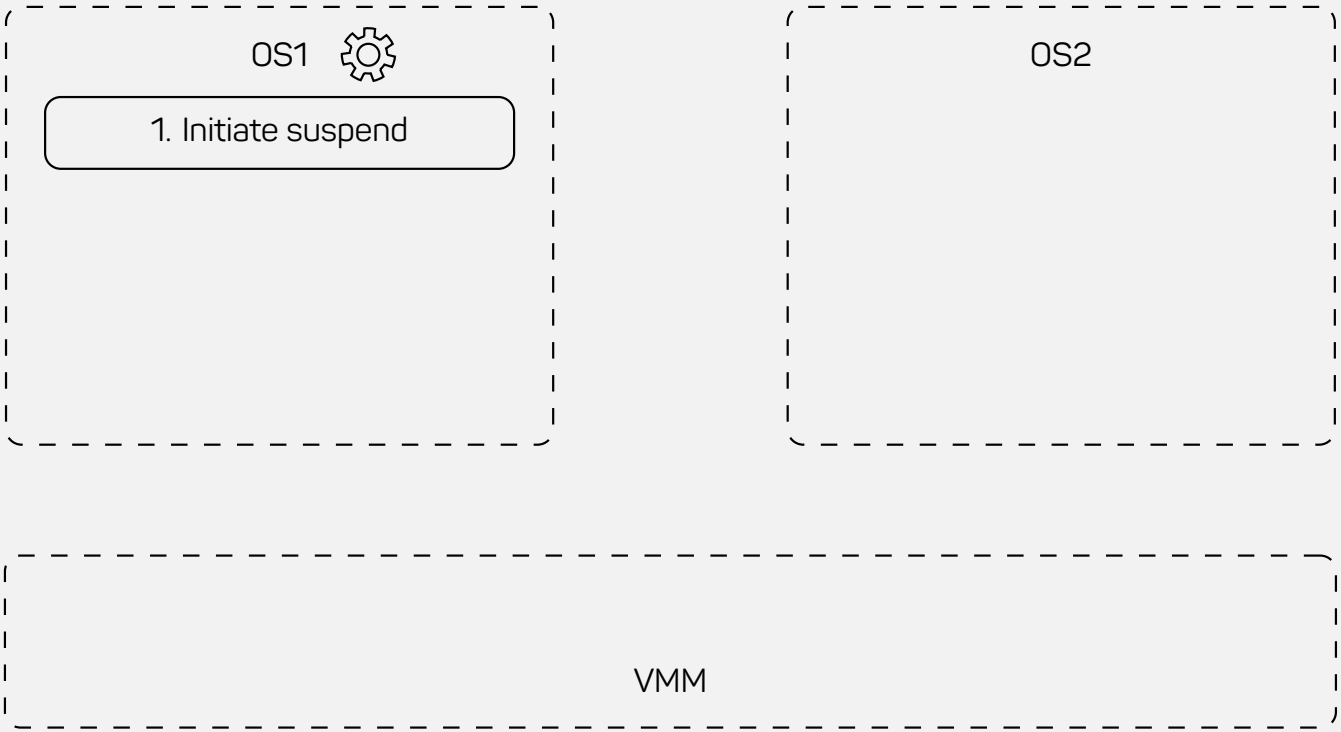
Approach

- to switch between multiple OSes, we need multiple OSes
- a virtualization layer grants us control over the hardware
- OSes expect certain memory regions
- the OSes have to handover their devices
  - use suspend-to-RAM (S3 sleeping state)

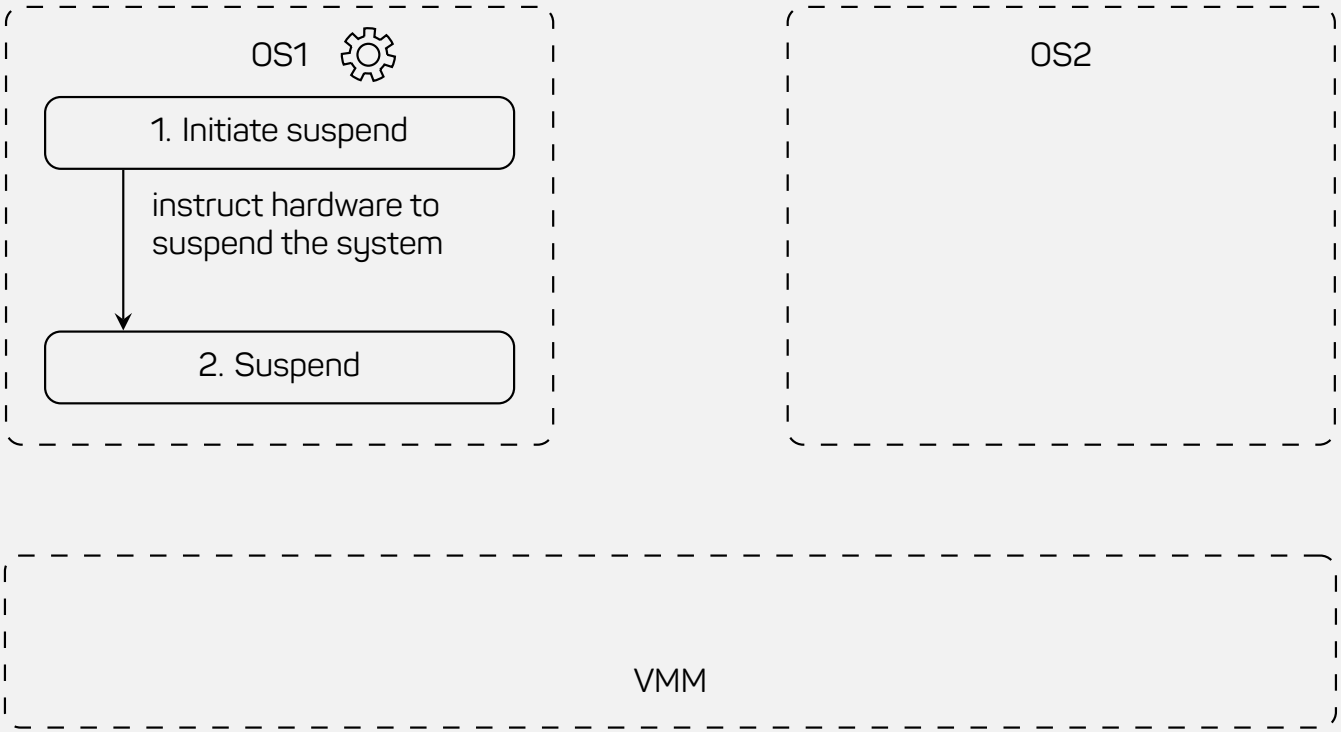
Approach



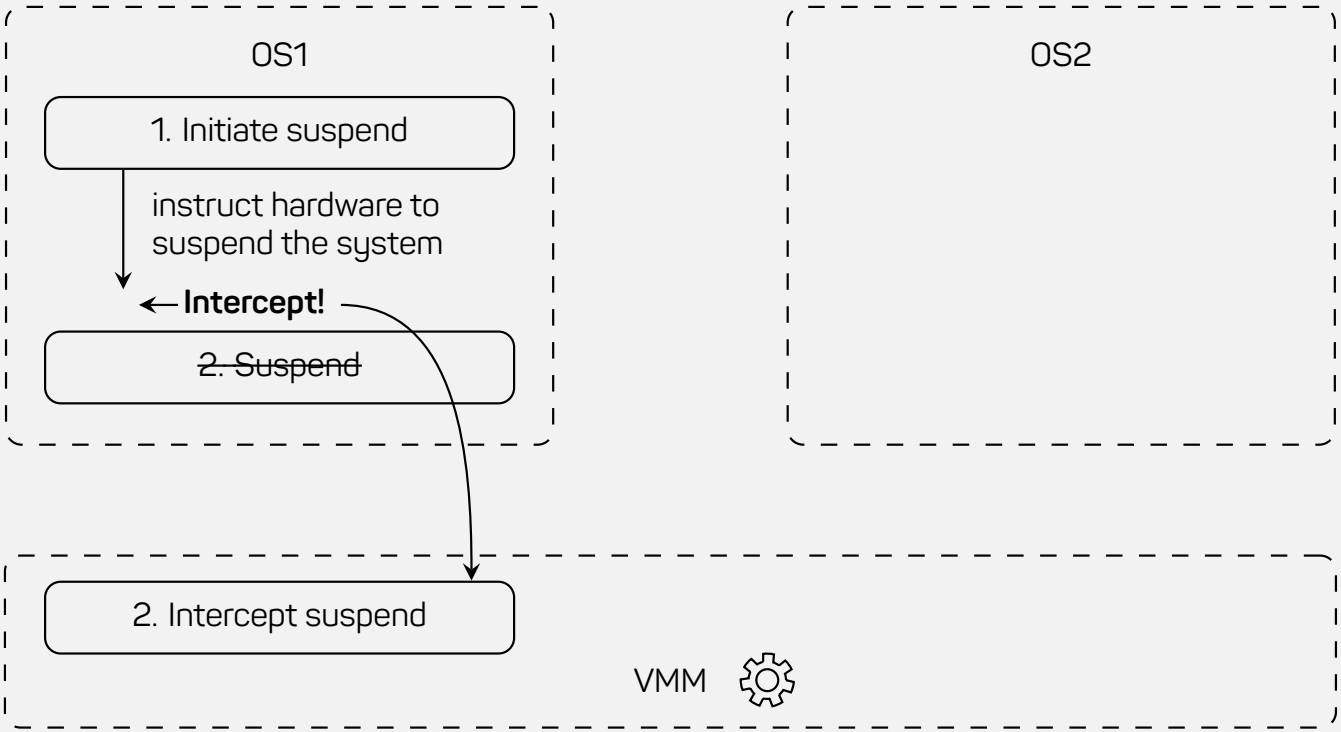
Approach



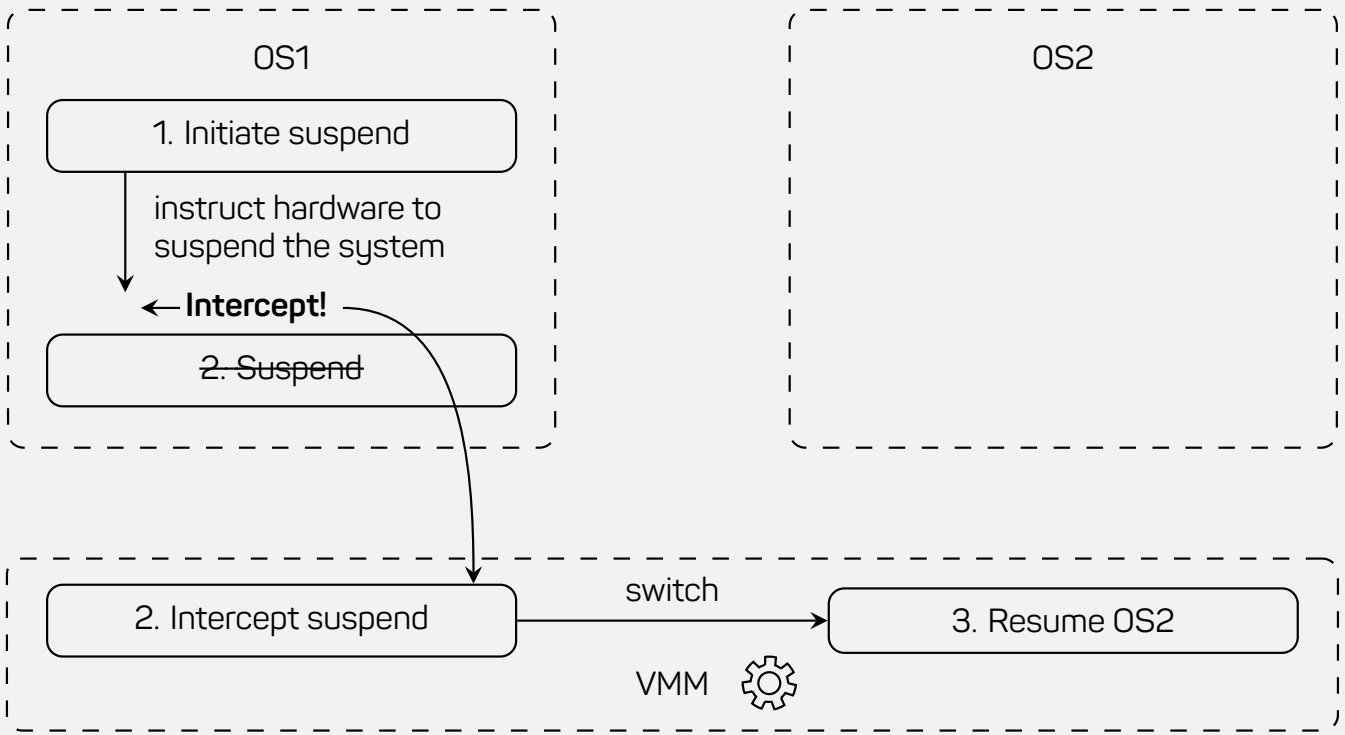
Approach



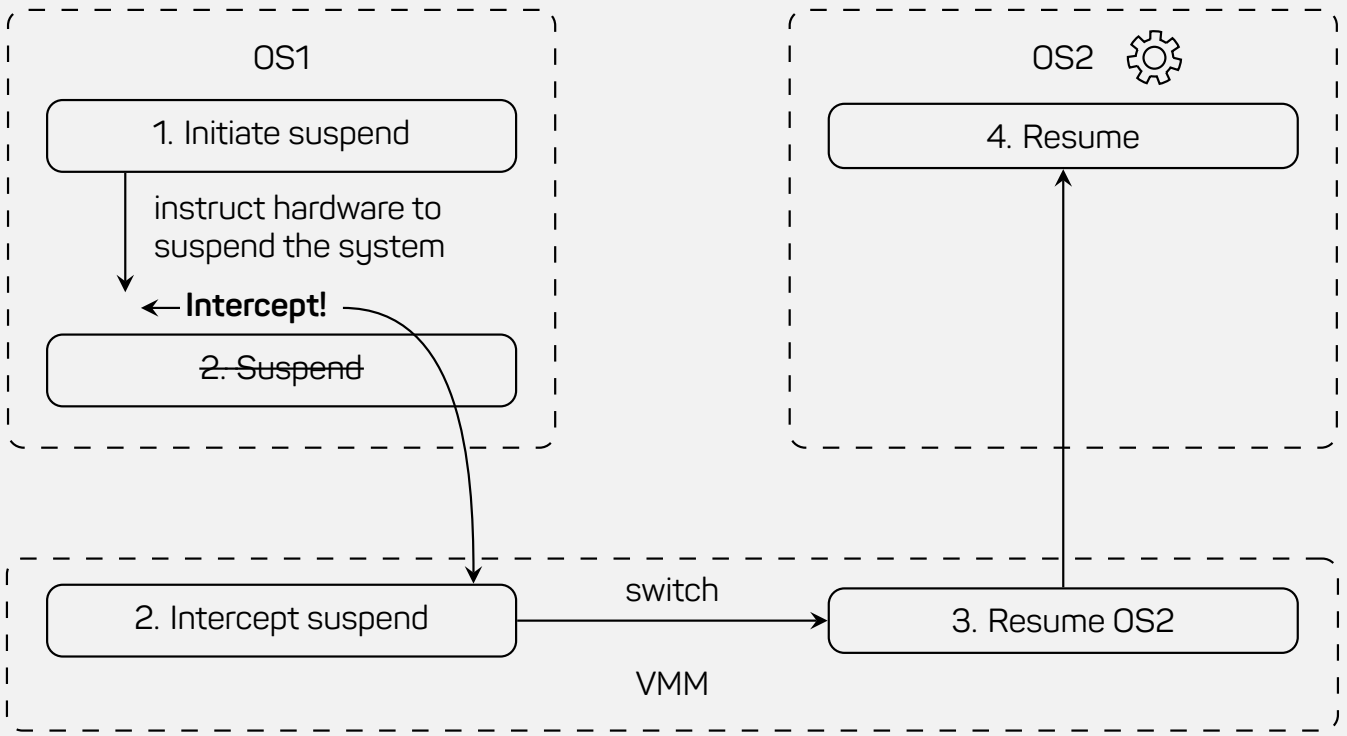
Approach



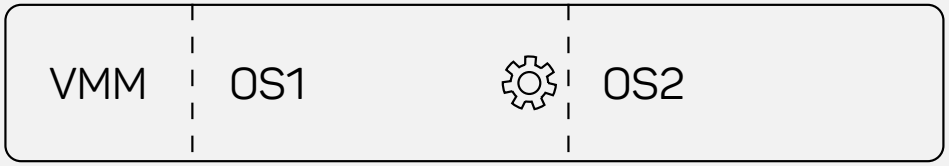
Approach



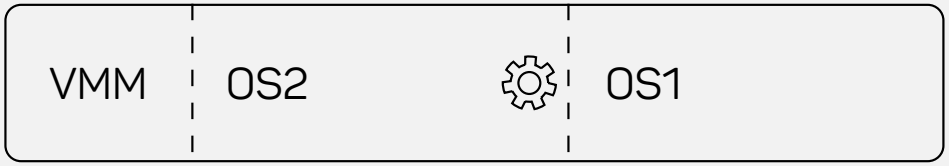
Approach



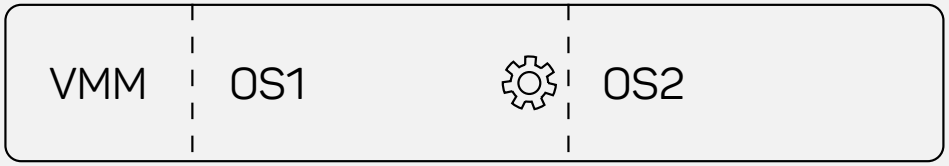
Approach



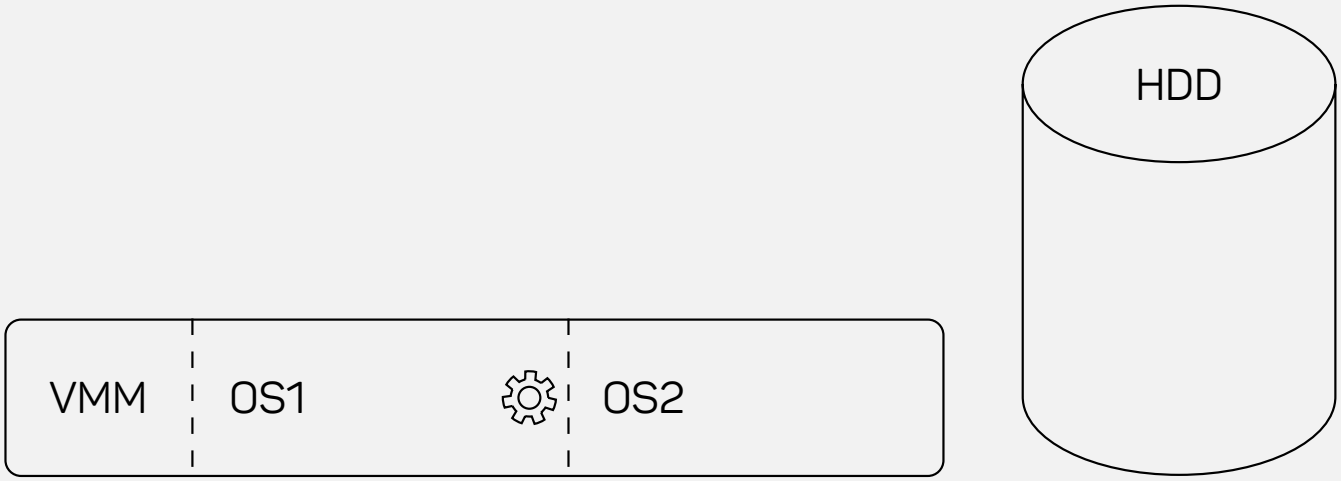
Approach



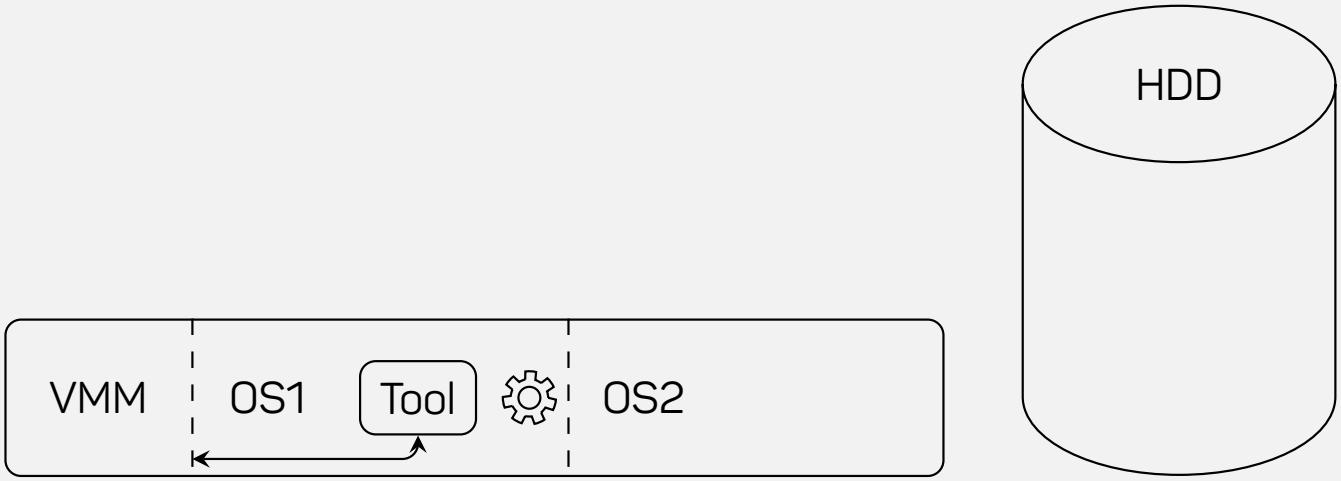
Approach



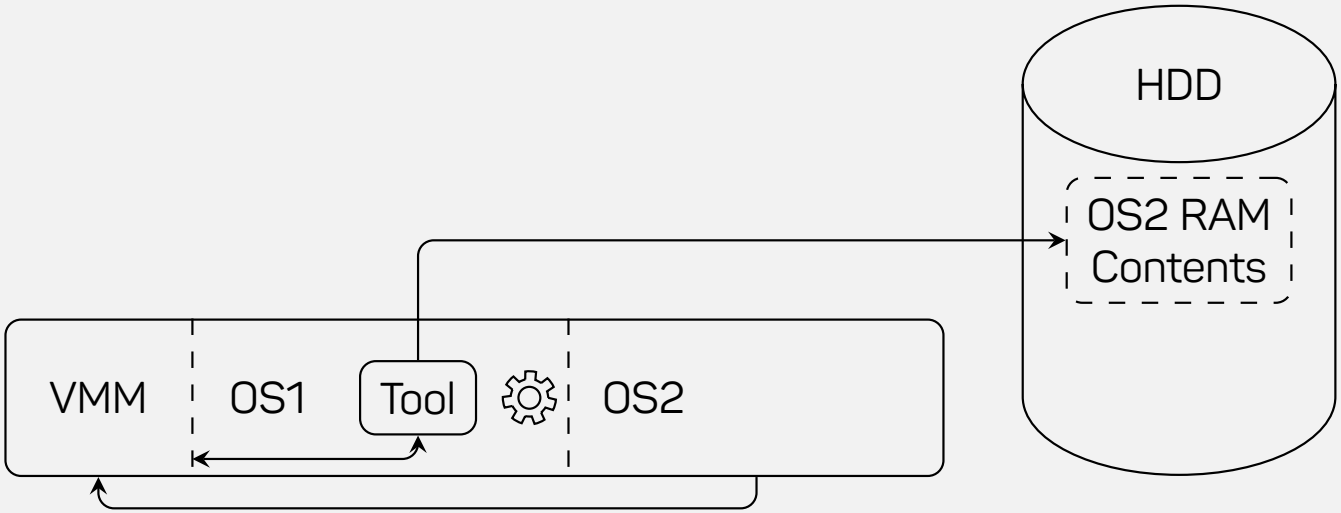
Approach



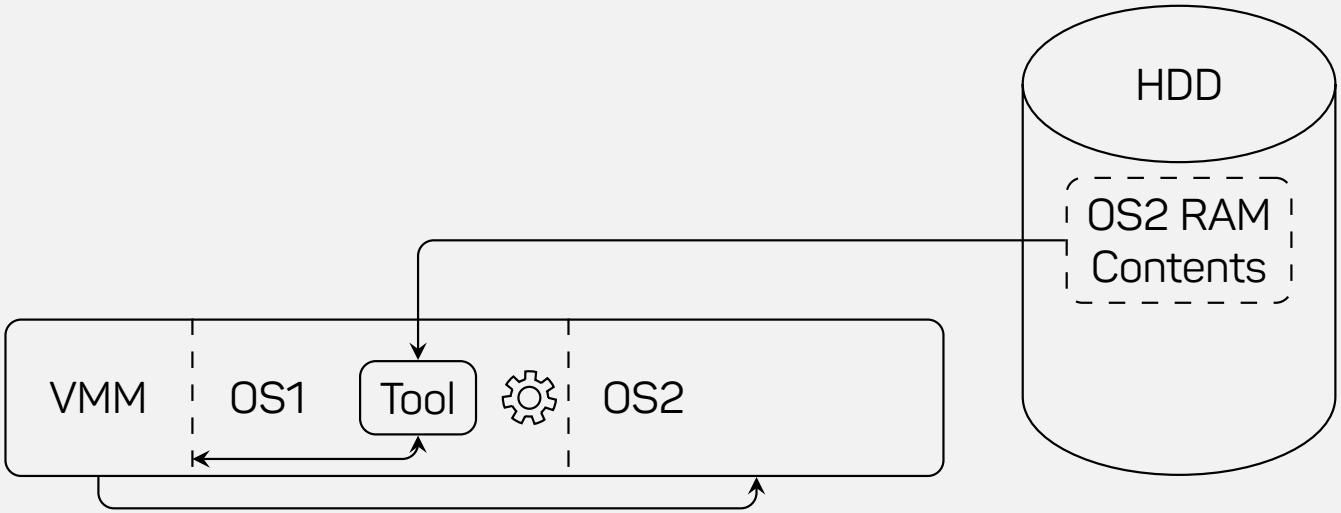
Approach



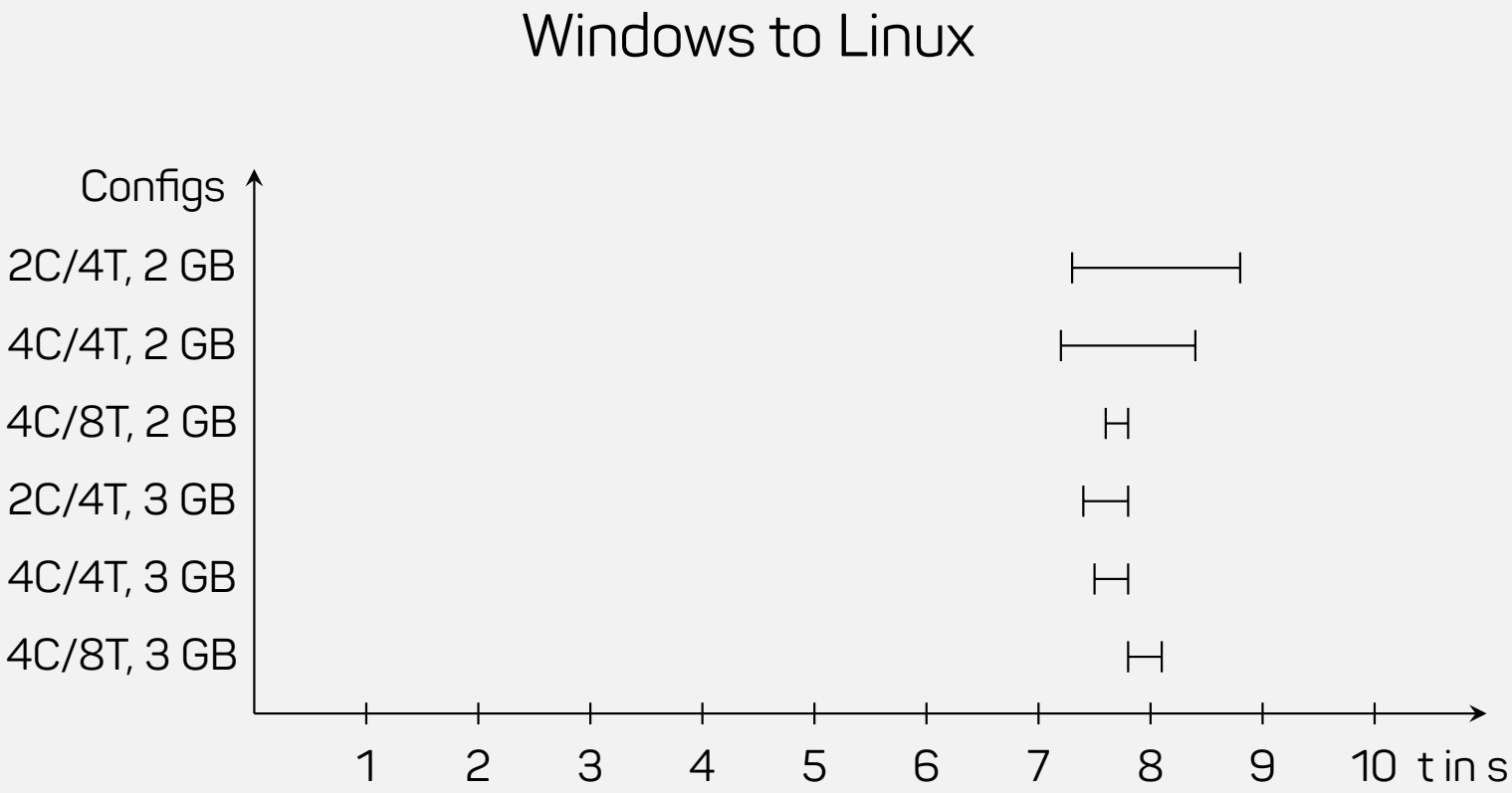
Approach



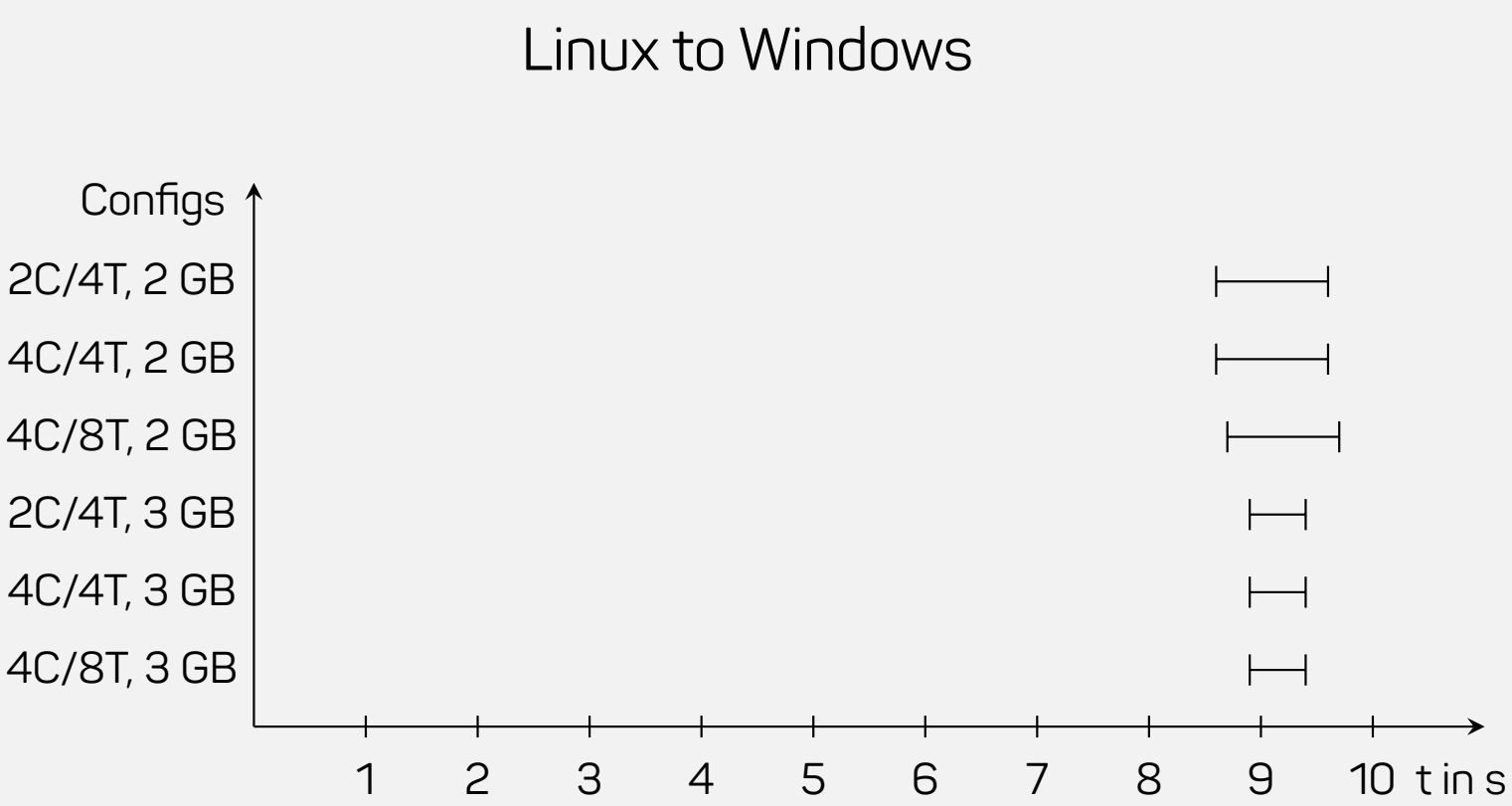
Approach



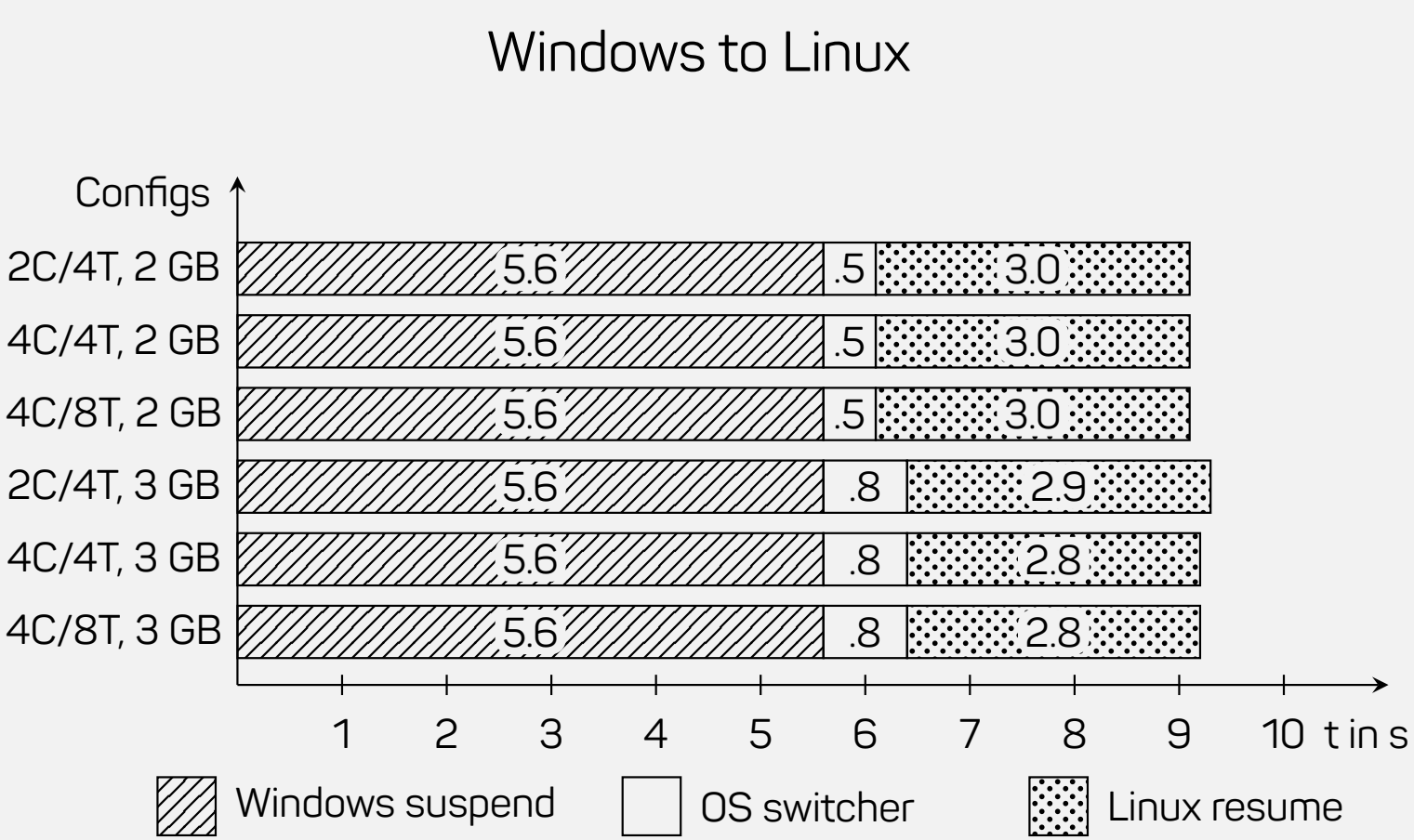
Measurements



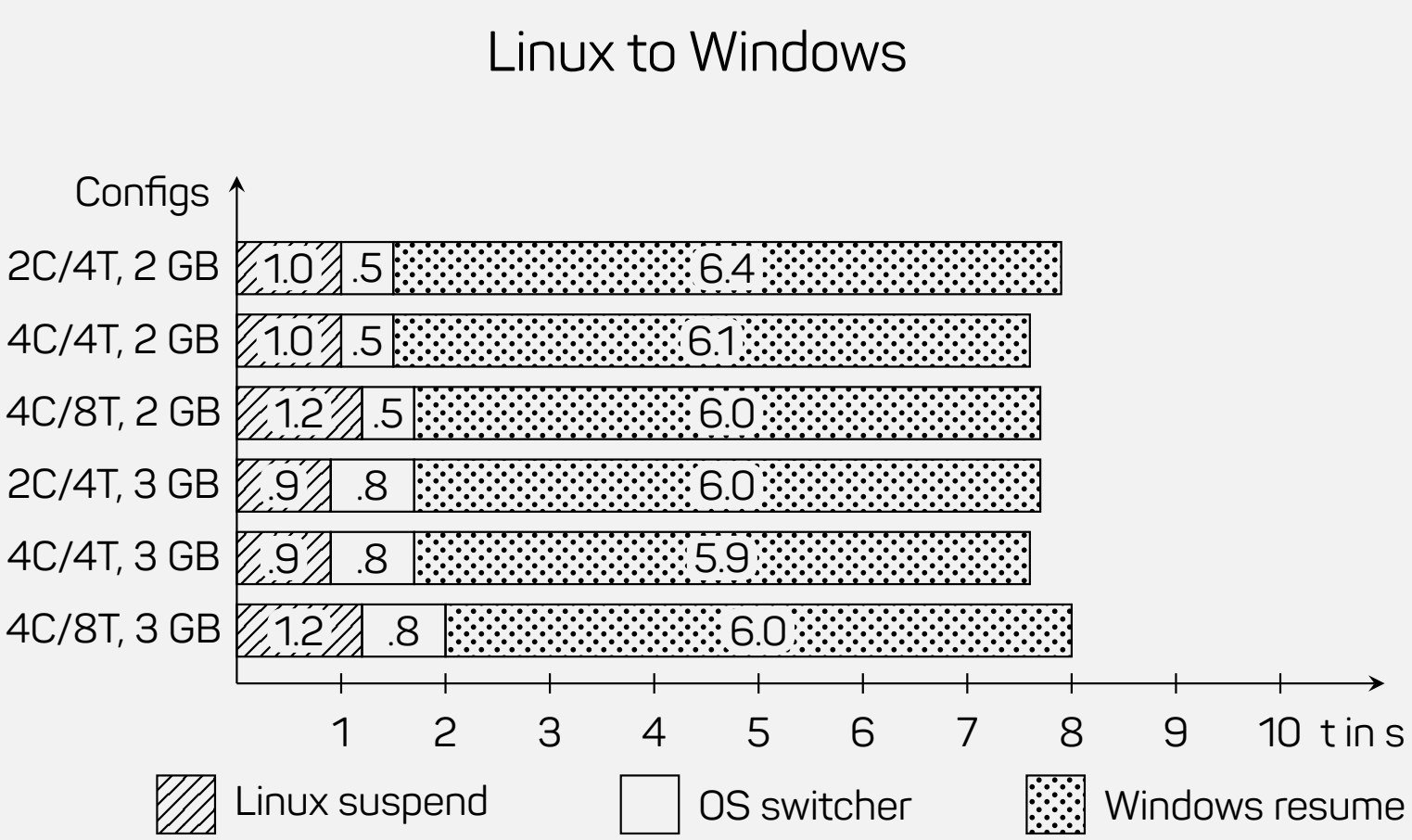
Measurements



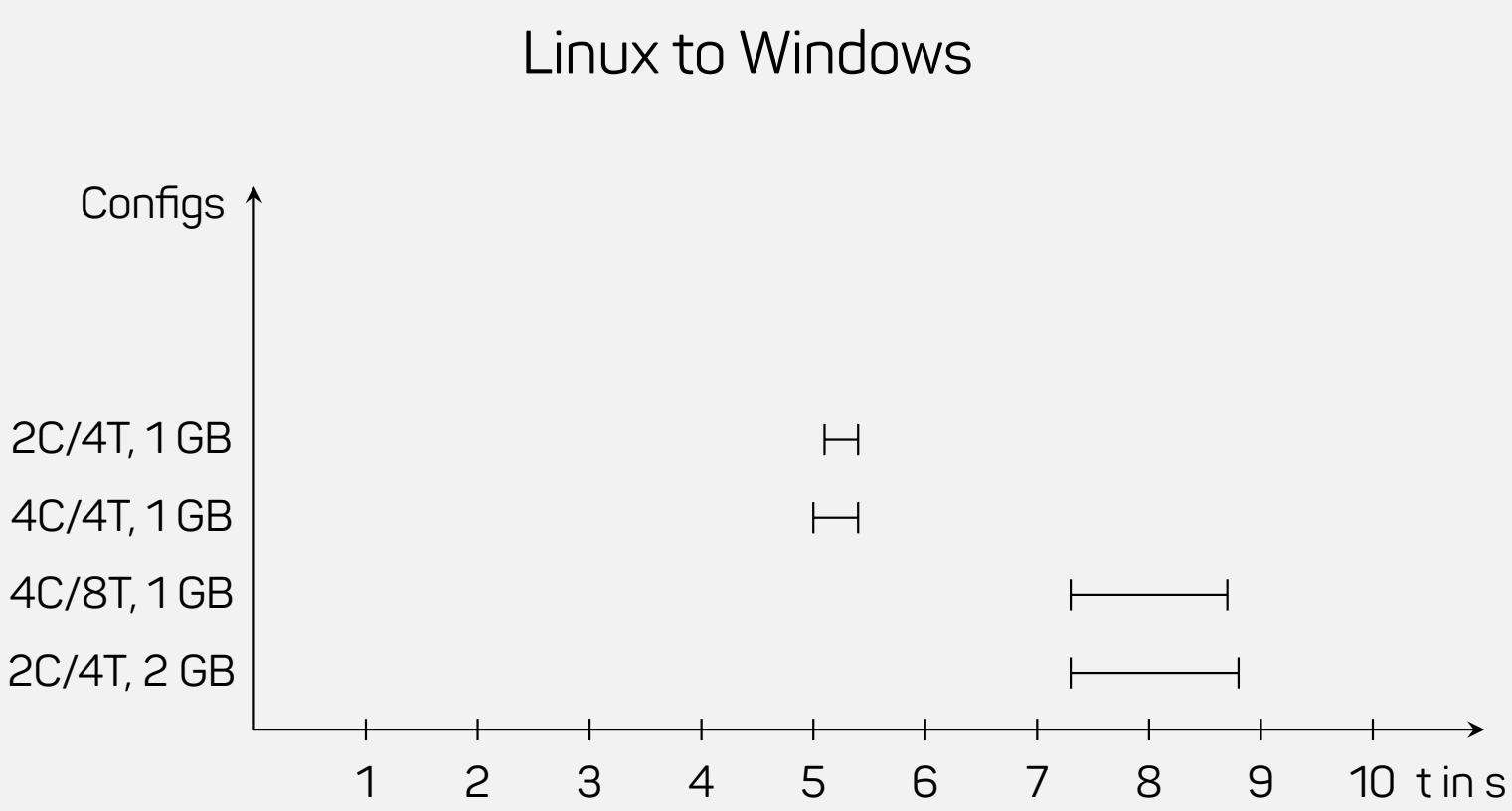
Measurements



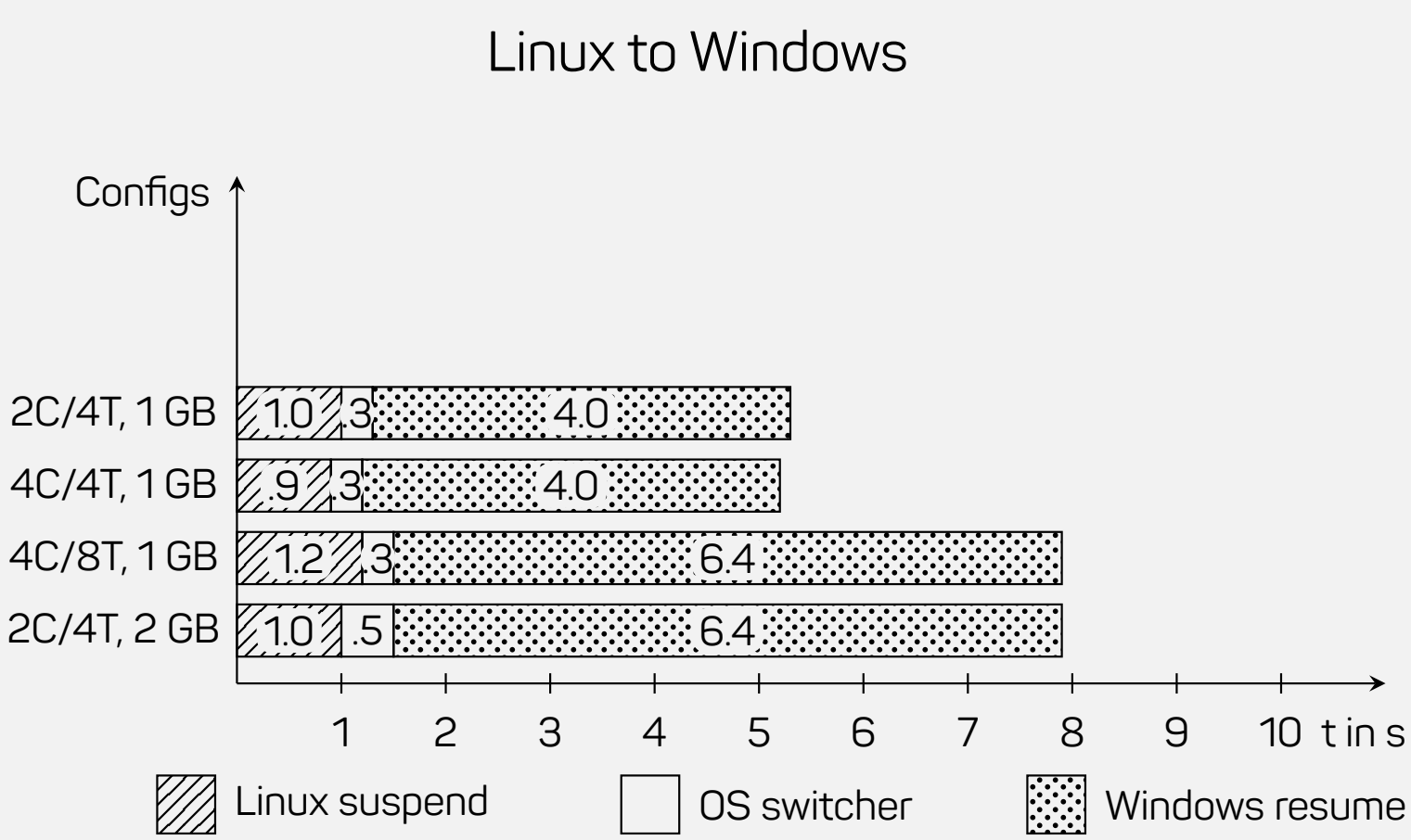
Measurements



Measurements



Measurements



Future Work

- limitations
  - switching time increases with available memory

Future Work

- limitations
  - switching time increases with available memory

- suggestions for future hardware
  - **each** device needs two functions that can be invoked using MMIO
    - `store_configuration`
    - `read_configuration`

Conclusion

Summary

- developed and implemented the OS switcher
  - use ACPI to avoid issues with device state management
  - use NOVA and SuperNOVA to intercept and control the guests

Conclusion

Summary

- developed and implemented the OS switcher
  - use ACPI to avoid issues with device state management
  - use NOVA and SuperNOVA to intercept and control the guests
- switch between OSes in under 10 seconds
- switching works reliably on the tested hardware