Scalevisor: A distributed hypervisor for rack-scale computing

Supervisors: Gaël Thomas, Mathieu Bacou

Pipereau Yohan

Télécom SudParis - Institut Polytechnique de Paris

July 7, 2021

How does VM hosting works ?

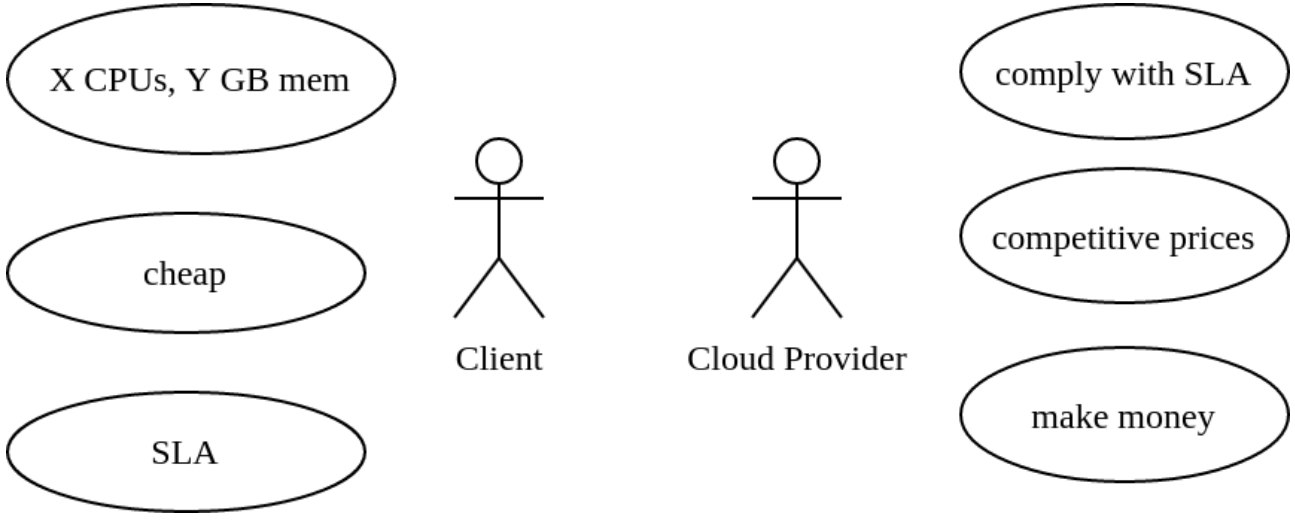


Figure: Use case

Cloud provider problem: fragmentation

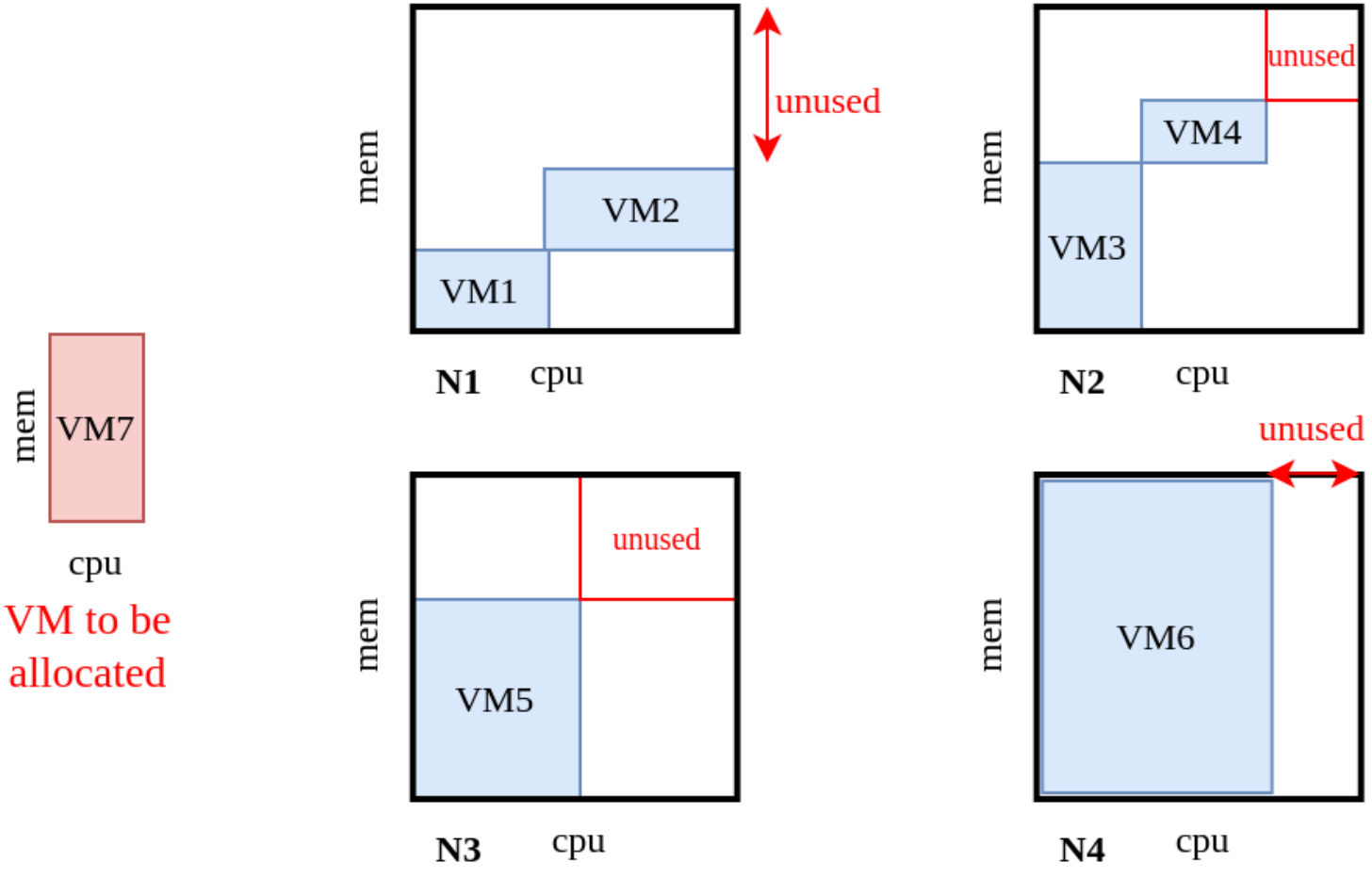


Figure: Resource manager - VM Scheduler

Fragmentation

Wasted resources

Eolas reported 40% unused resources in its datacenter !

Improving consolidation

- Saving money
- Reducing Energy Consumption

Problem - Complexity

Finding minimum number of servers for a set of VM is NP-hard !

Solution: memory mutualization

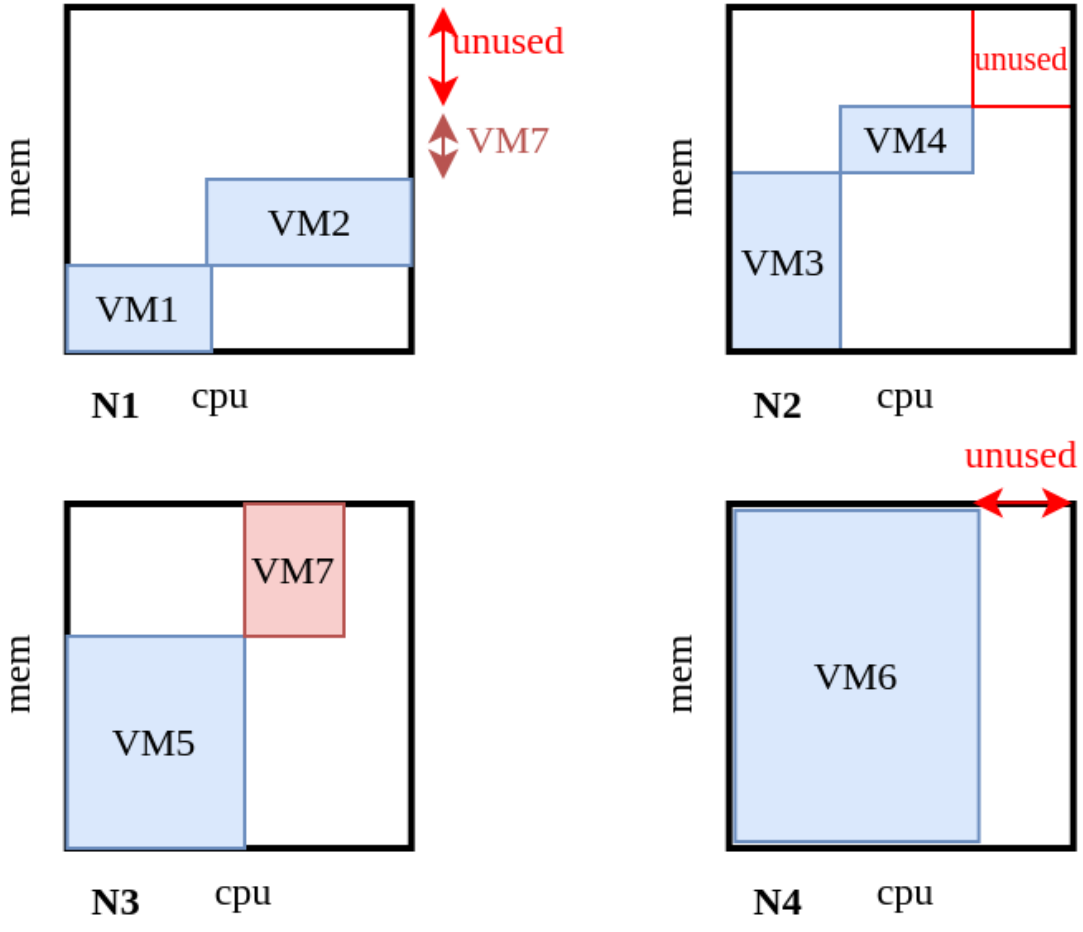


Figure: Resource manager - VM Scheduler

Solution: memory disaggregation

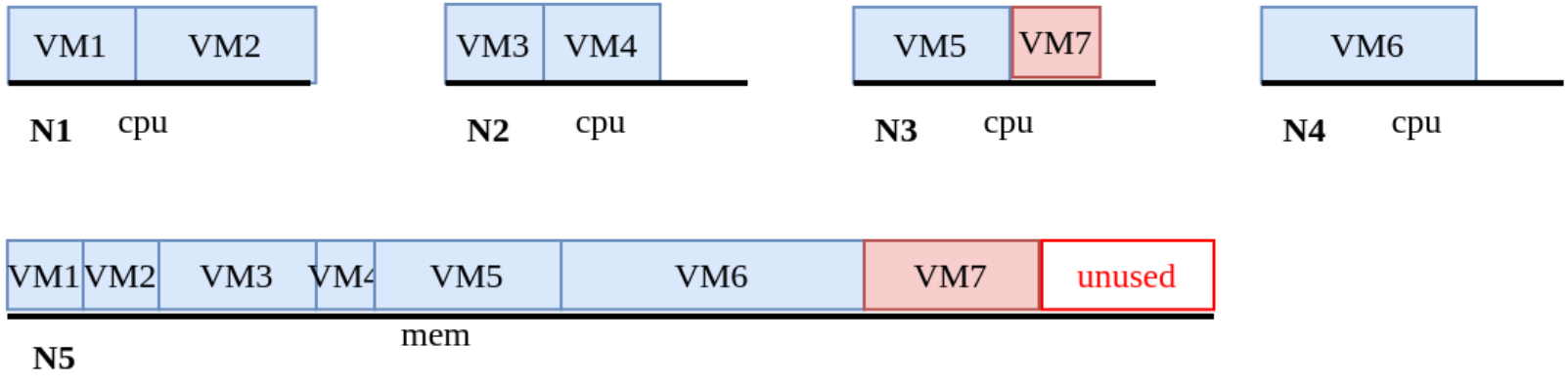


Figure: Resource manager - VM Scheduler

Virtual machines consolidation

Definition - Consolidation

- **scenario 1**: run more VMs on same number of PMs
- **scenario 2**: run same number of VMs on less PMs

2 configurations for VM placement

- **Static VM Consolidation**: initial placement of VMs
- **Dynamic VM Consolidation**, reallocation of existing VMs

Some techniques

- **VM migration**: move 1 VM from PM1 to PM2
- **memory overcommitment**: $\Sigma VM_{memory} > PM_{memory}$ (memory ballooning, transparent page sharing, memory compression, swapping)

Dynamic resource management - VM migration

Similar Problem

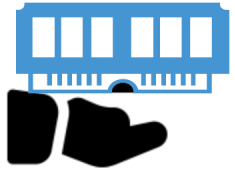
Similar to **static** resource management for VM allocation

Useful

- Upgrade a server
- Consolidate virtual machines

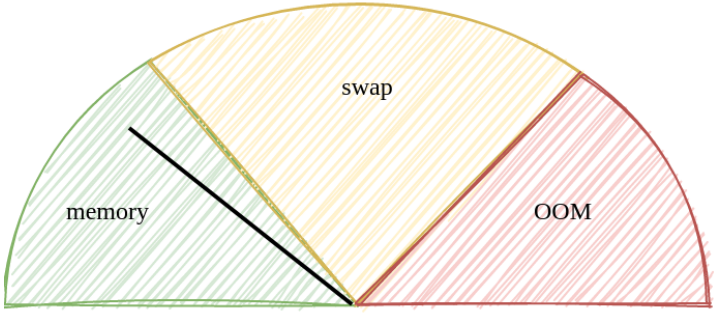
Problem Statement

How can we mutualize virtual machines memory in a datacenter ?

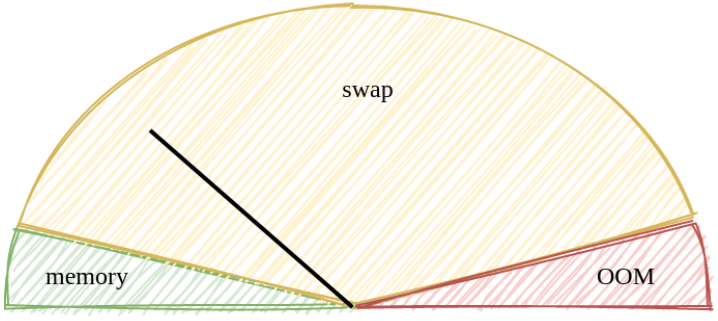


Design & Implementation

Guest memory view



(a) Classical VM memory usage



(b) Our memory usage model

Overall design

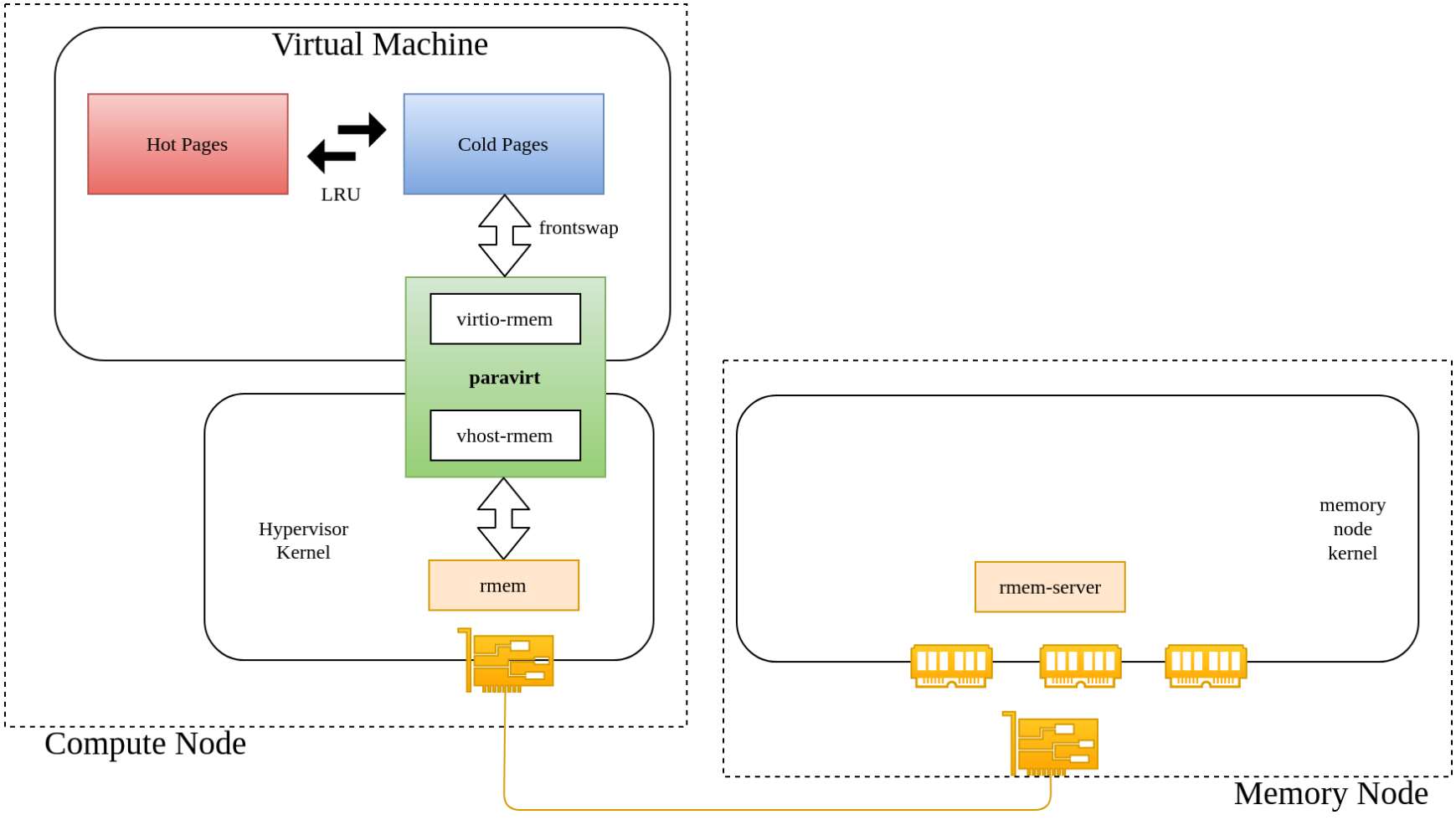


Figure: Overall design

Overall design

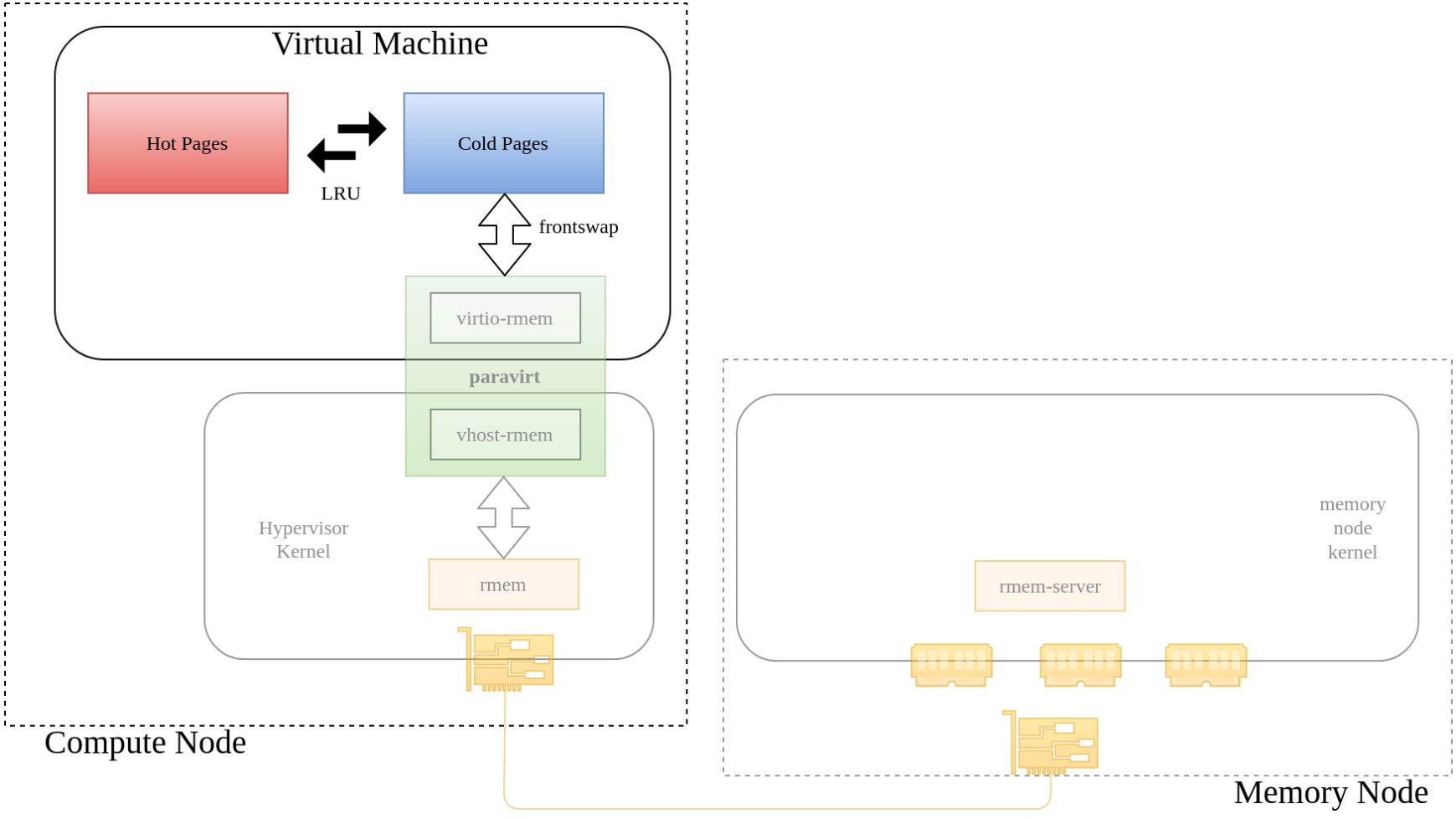


Figure: Overall design

Overall design

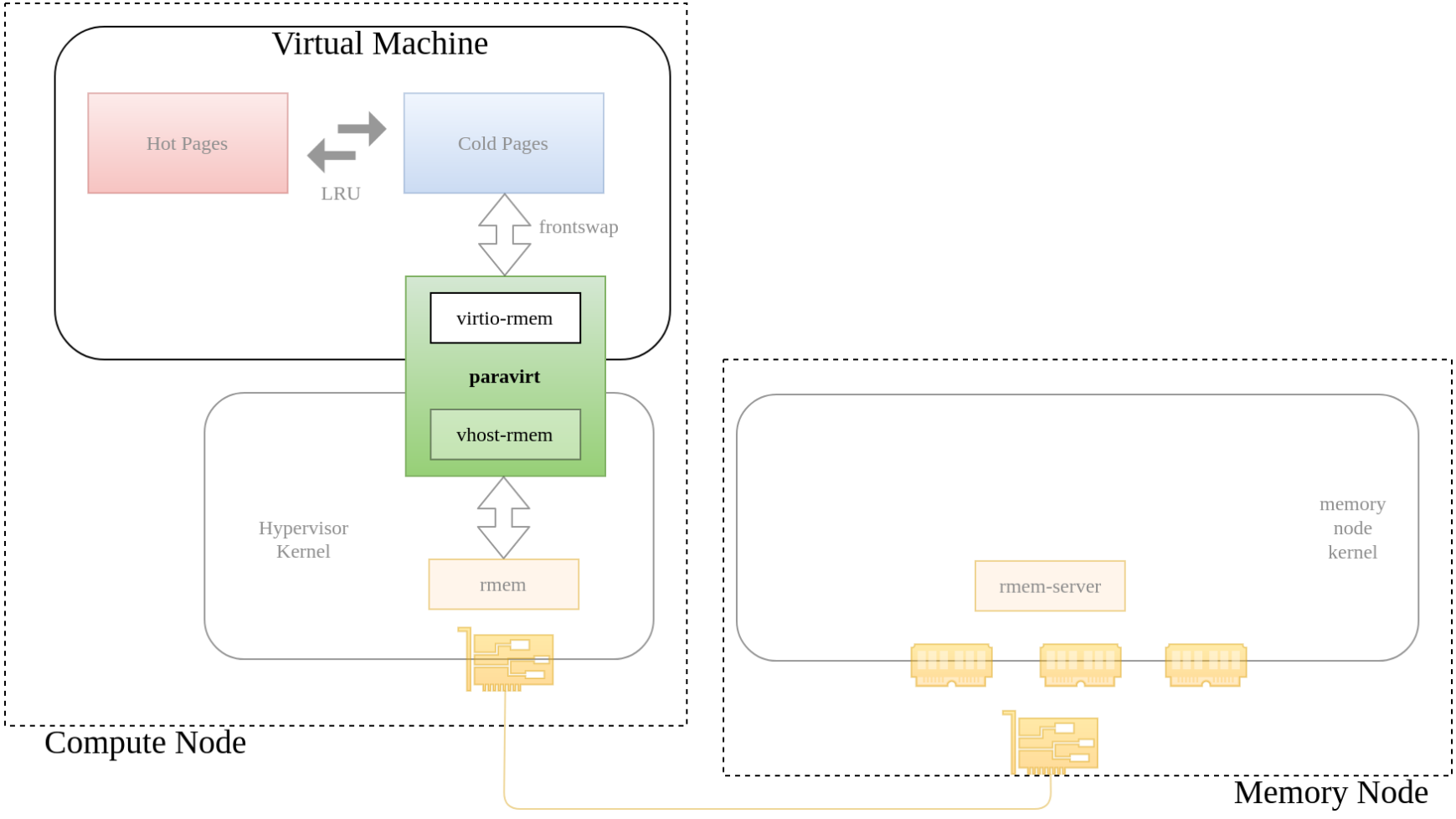


Figure: Overall design

Overall design

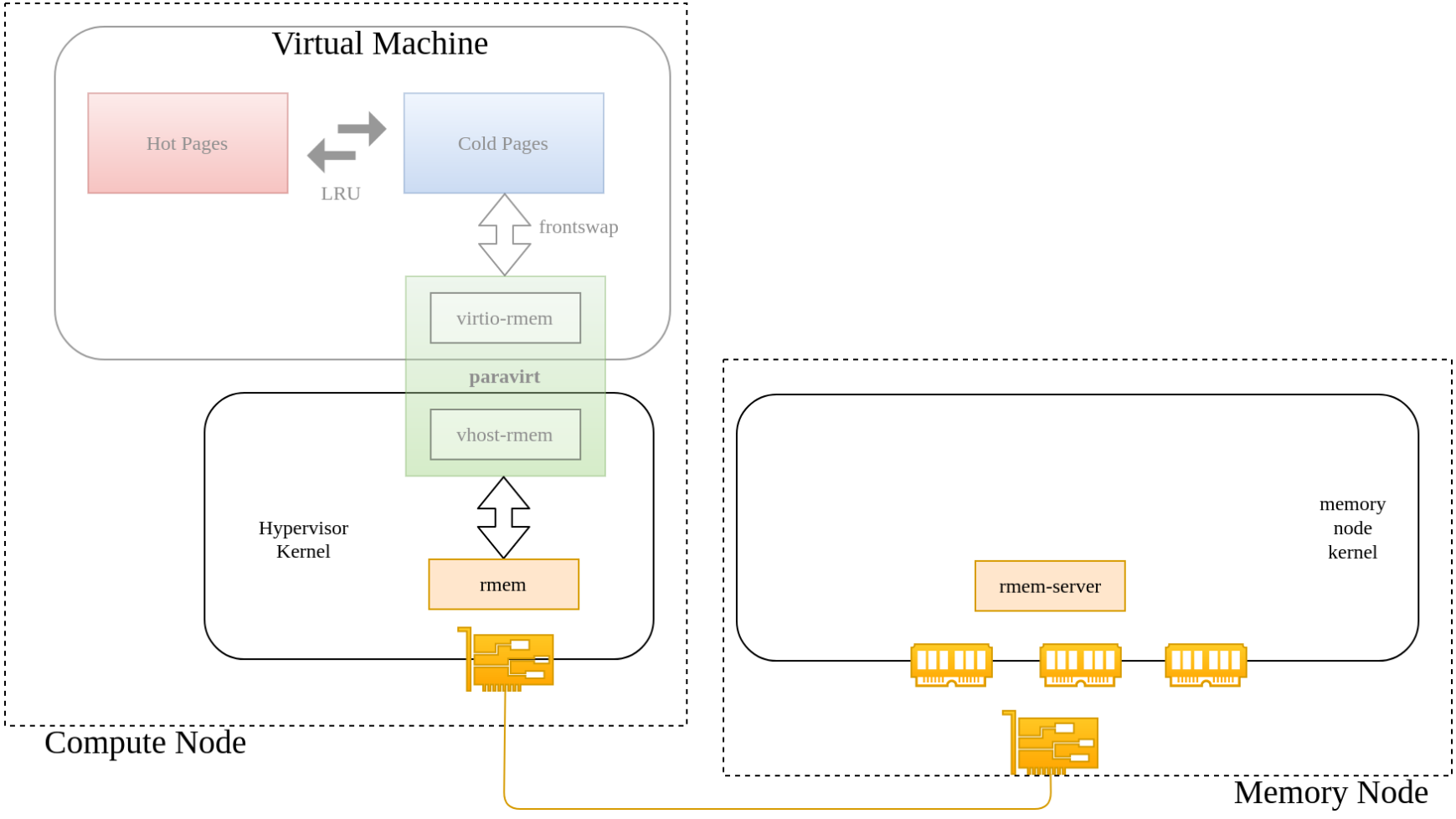


Figure: Overall design

Design & Implementation

rmem layer

rmem API

```
struct remote_memory_region {
    u64 remote_addr;
    u32 remote_rkey;
    u32 remote_len;
};
```

- `alloc` : 1-sided
- `free` : 1-sided
- `write` : 2-sided
- `read` : 2-sided

Design & Implementation

paravirtualization layer

What is paravirtualization

Definition - Paravirtualization

A technique of communication between a VM and a hypervisor

Why do we need paravirtualization ?

Semantic gap

Hypervisor has no visibility on memory management in the guest

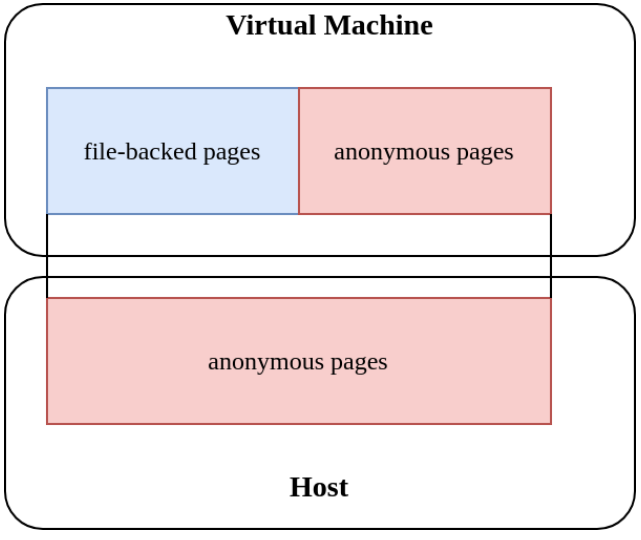


Figure: false page anonymity

Design & Implementation

swap layer

swap layer

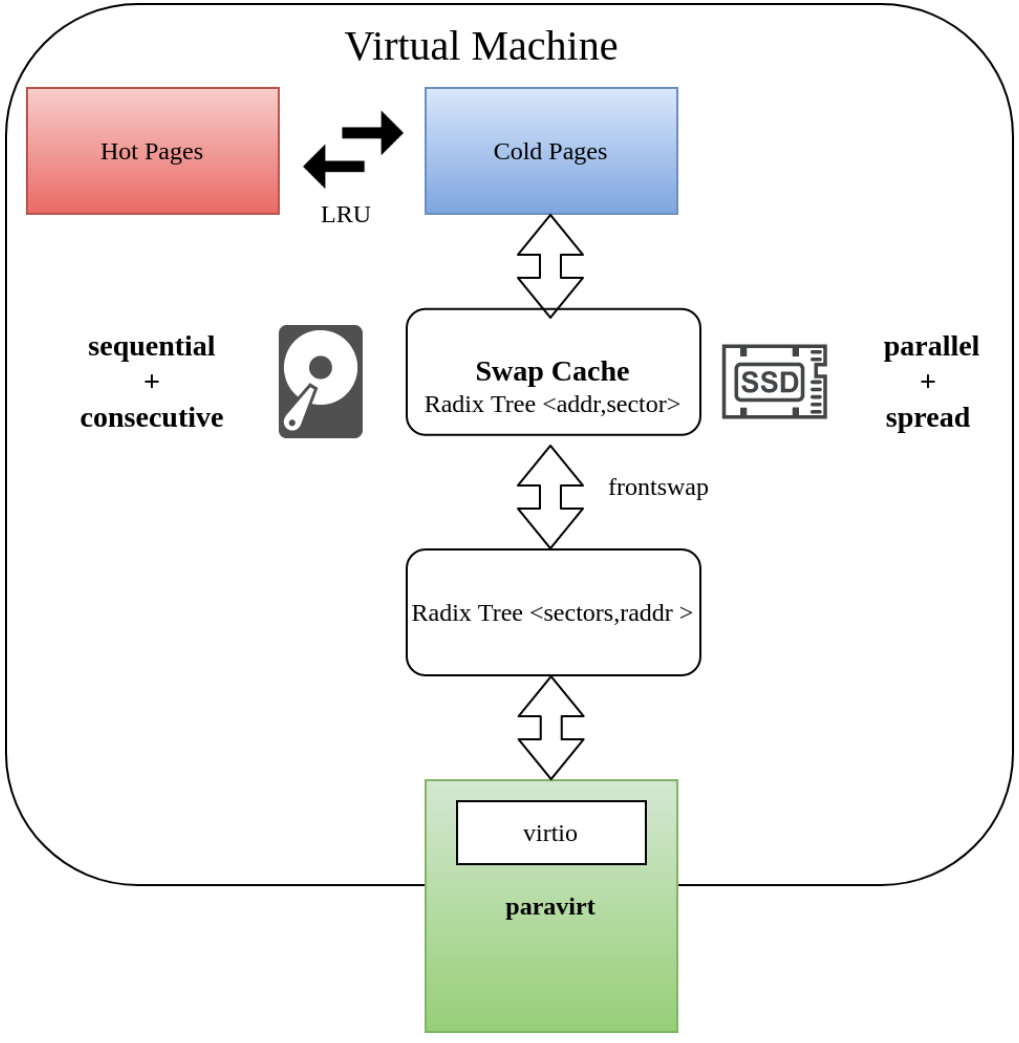


Figure: swap backend

Design & Implementation

How to use ?

How to use

VM allocation steps

1. In the host, connect with a memory server
2. In the host, launch VM vith paravirtualized driver
3. In the guest, swapon a block device or swapfile

Design & Implementation

Live VM migration

Why do we need live VM migration ?

Currently

✓ Memory mutualization for **static VM consolidation**

Remember - Live VM migration

For **upgrade** and **dynamic VM consolidation**

Support for Live VM migration

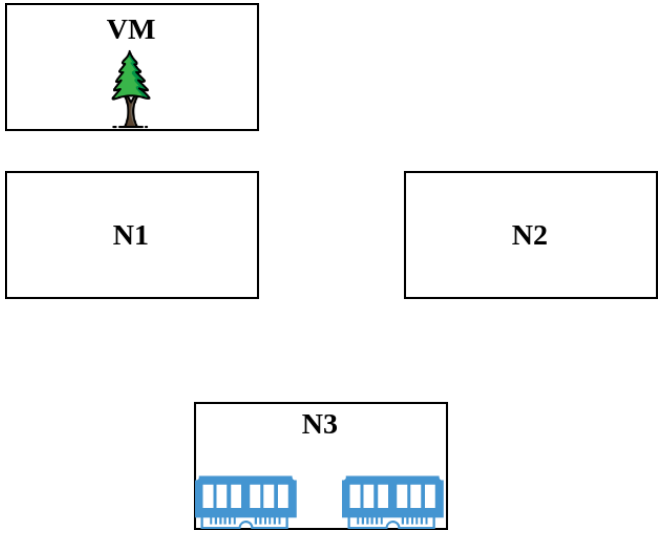


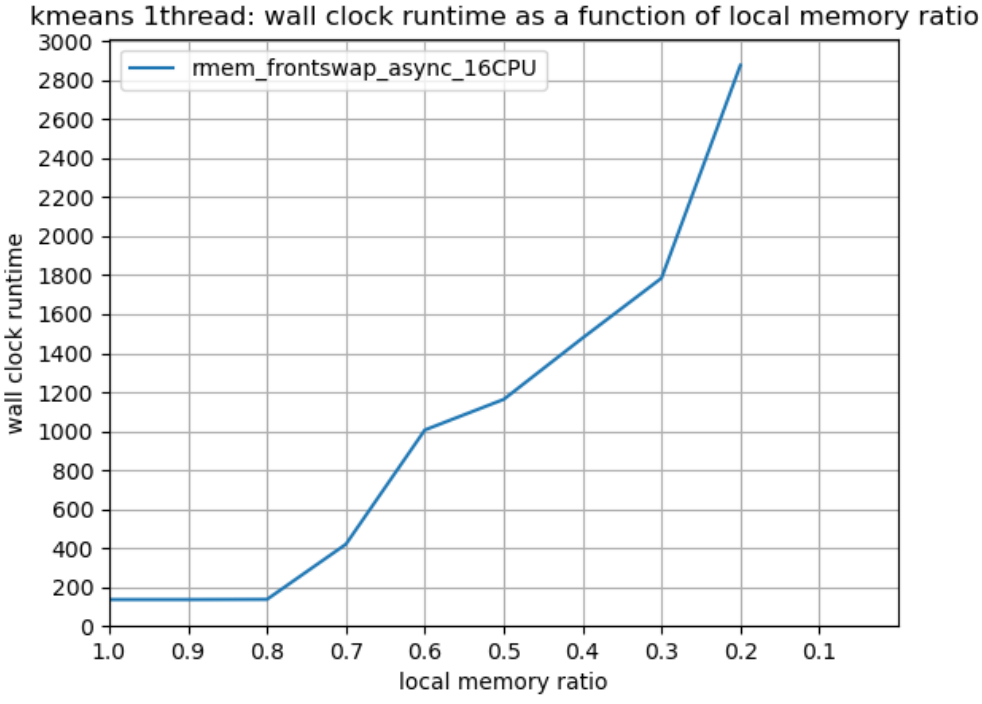
Figure: RX tree in guest for live migration

Evaluation

Number of lines of code

project	SLOC
qemu	713
linux	6
virtio-rmem	993
vhost-rmem	1713
rmem_server	1103

Table: Number of lines of code



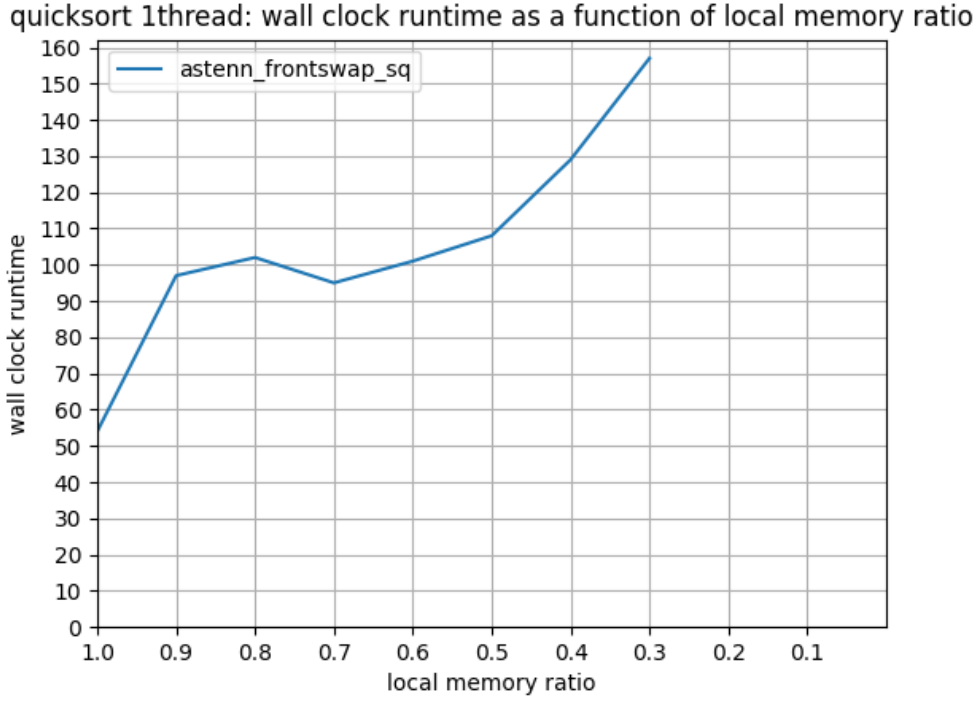


Figure: Wall clock runtime as a function of how much memory is local

Interpretation

Different degradation profile under different memory disaggregation ratio !

Conclusion

Conclusion

Remember one thing

We swap on remote memory to use remaining free memory on other hosts.

Expectations

1. Remote memory latency: 5-10 µs
2. Improve consolidation ratio
3. Reduce VM migration time

Big problems

- Support Live VM migration of zerocopy devices
- What happens when allocations on remote node fails for a size because of memory fragmentation

Questions ?

Appendix

Distributed Shared Memory 1/2

How ?

Deviate TDP page fault

Advantages

- Enable to build a real address space
- More flexibility to implement cache coherency operations

Distributed Shared Memory 2/2

Drawbacks

- Lot of engineering
- Vendor-specific
- what if shadow page table ?
- In-kernel modifications
- No distinction between *hot memory* and *cold memory*
- Cache coherence protocols are expensive

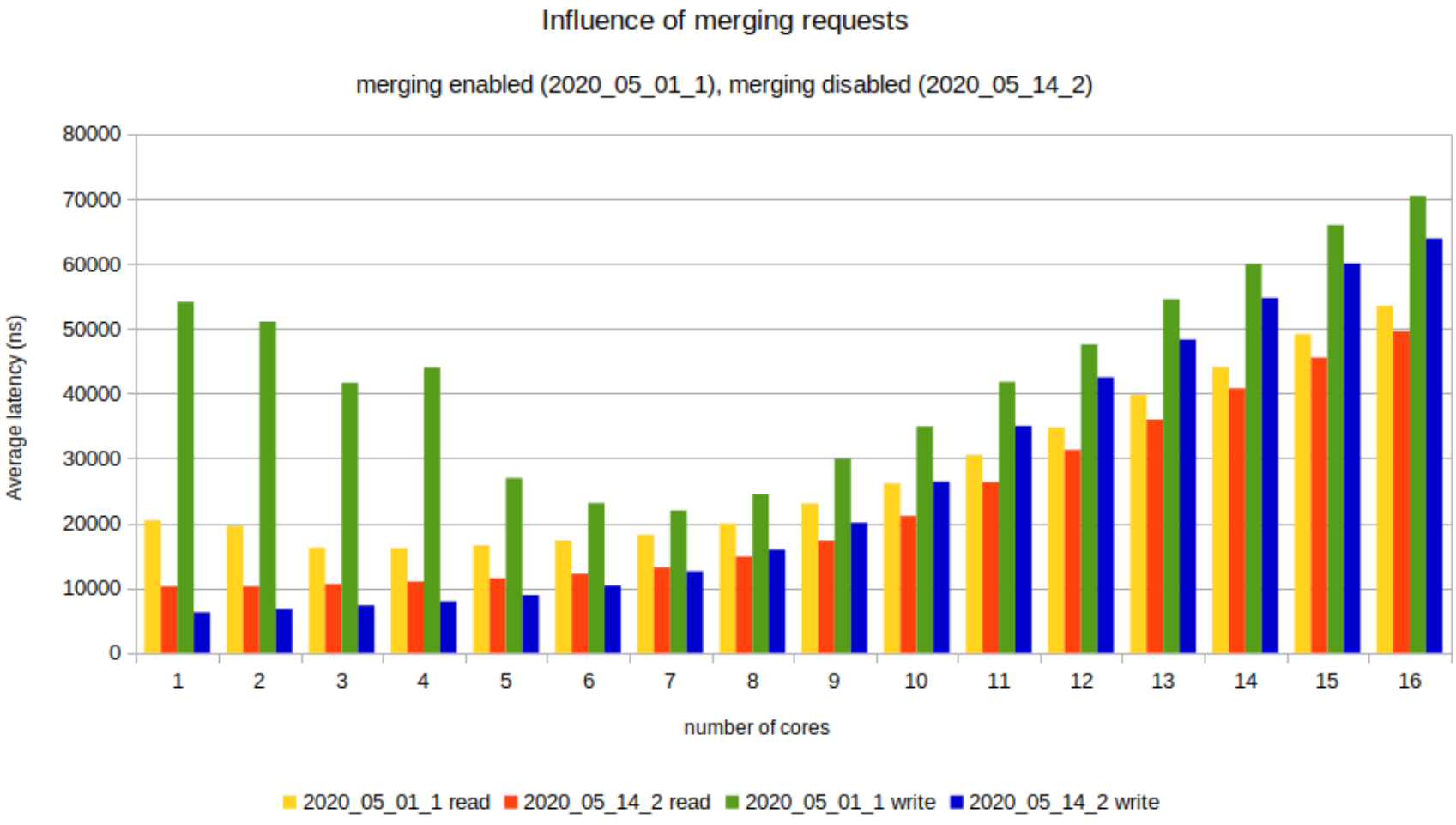


Figure: Enabling/Disabling requests merging

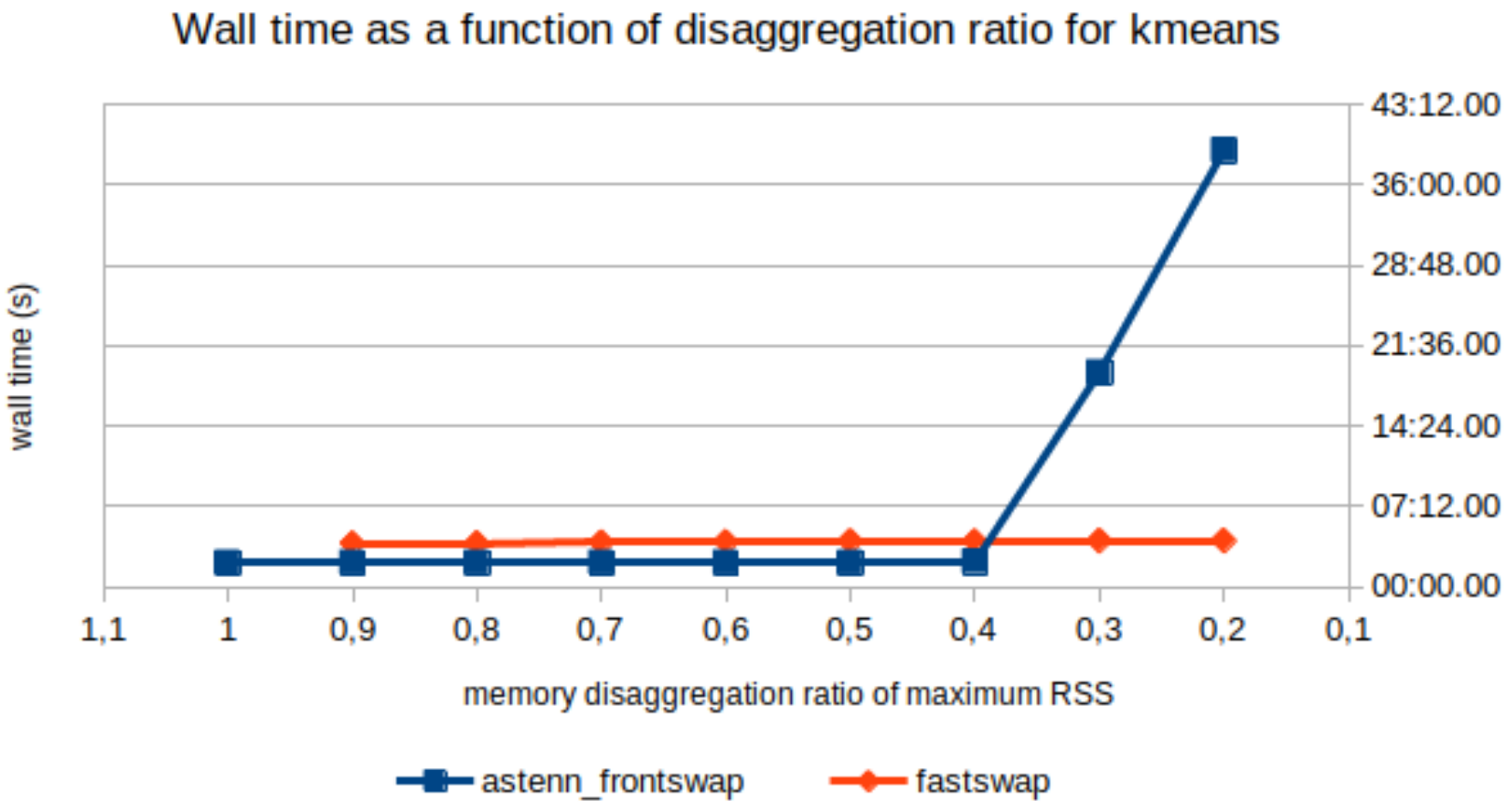


Figure: Enabling/Disabling requests merging

A qemu-kvm virtual machine

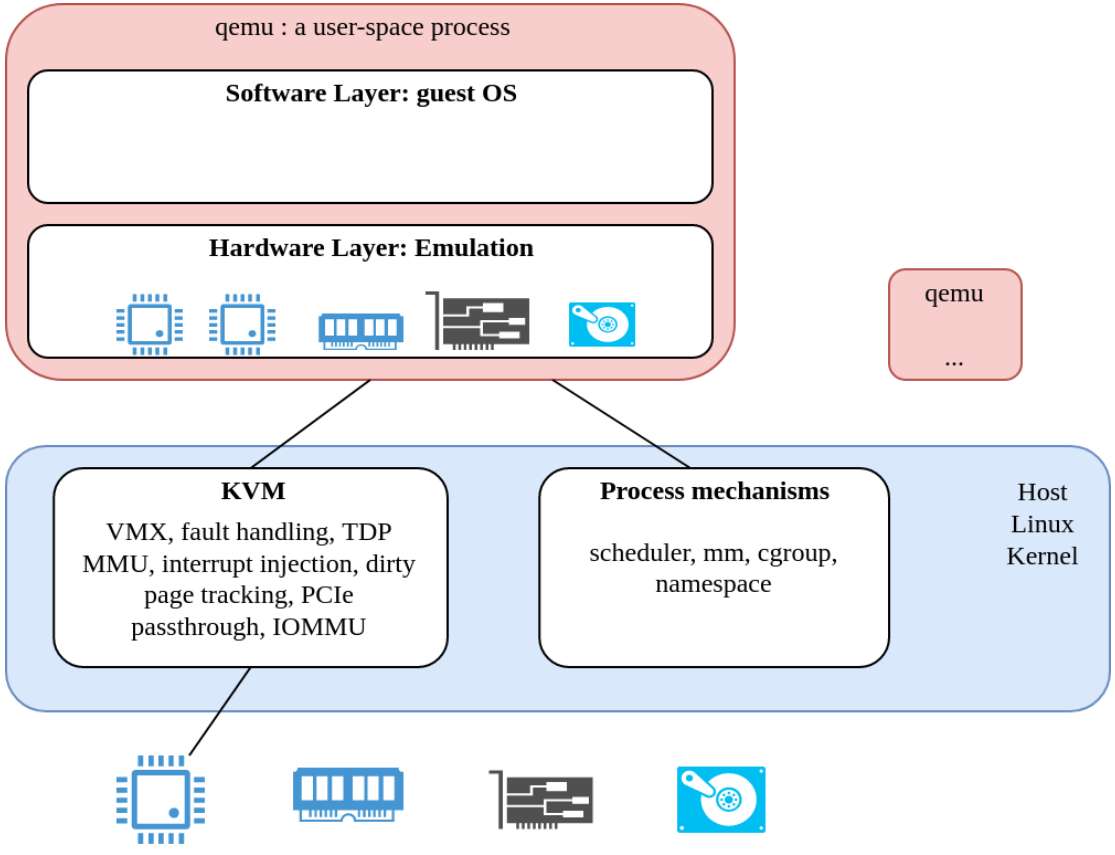


Figure: Enabling/Disabling requests merging

paravirtualization layer

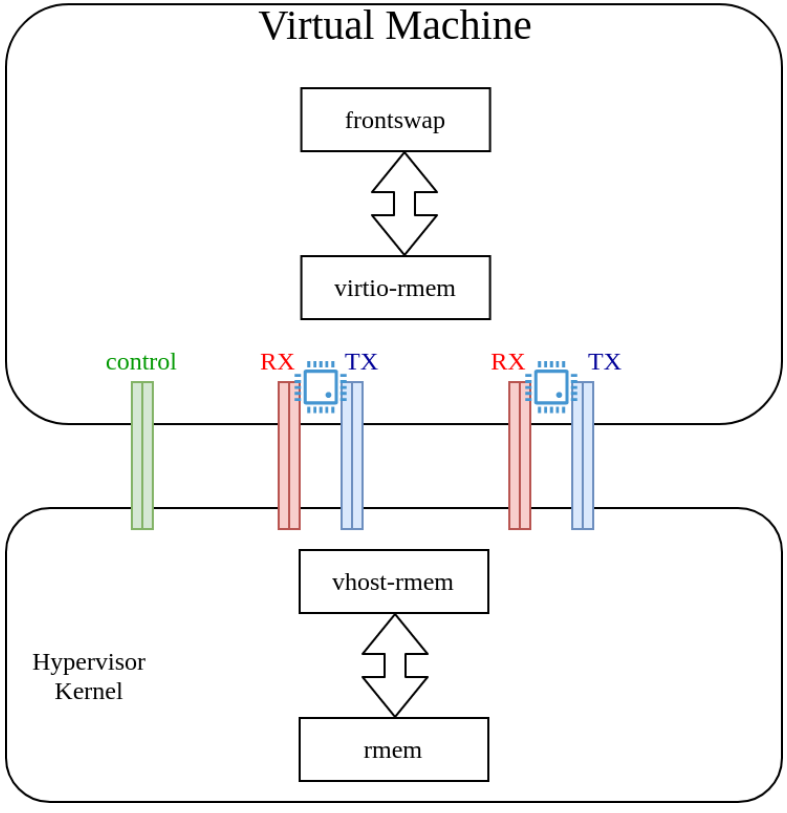


Figure: paravirtualization