Problem

In cloud ...

- One machine is rented to multiple tenants
- Machine is sliced into many VMs running on a Virtual Machine Manager (call it: Host OS)
- To protect customer data host OS has to be stable and secure. Therefore regular updates are required
- Updating host OS disturbs VMs, and thus reduces the availability
- How do we update it without disturbing VMs?
- The problem is also applicable to regular processes, and containers.
Existing methods

Seamless update methods

- Hot patching
  - Not suitable for complex fixes
  - Can’t provide full stack update

- Live Migration
  - Requires extra hardware
  - May permanently slow down VMs

Mitigations

- Faster reboot: kexec to skip firmware
- Keep VM state in memory for quicker reattaching after the reboot
Possible Solution #1: cooperative multi-OSing

Allow the old VMM run while the new VMM is booting.

Problems:

- If suspend/resume + device quiesce is expensive, update is not truly seamless
- Conflicts with shared resources like networks, FSes
Possible Solution #2: VM to bare metal

- Boot an updated version of host OS in a virtual machine with the same memory layout as the host and present cpus to be exactly the same as the physical CPUs: all emulated instructions would return true values: true cpuid, true tsc etc.
- Migrate the running VMs inside the new host OS (can actually be done after the transition to bare metal)
- Kexec into a special entry point from old host OS to the new host OS.
- At the entry point fix the ept translations: either relocate the physical pages where the new host OS expects them (safer, requires more copying), or fix page tables within the OS itself (fragile, potentially faster).
- A process of quiescing and initializing the devices after the kexec in a new host OS need to be designed
Other solutions?