Bring QEMU to Micro Service World

Zhang Yulei <yuleixzhang@tencent.com>
Xiao Guangrong <xiaoguangrong@tencent.com>

Tencent Cloud

Agenda

- Background
- QEMU Adaption
- Future works

Tencent Cloud

Tencent Cloud

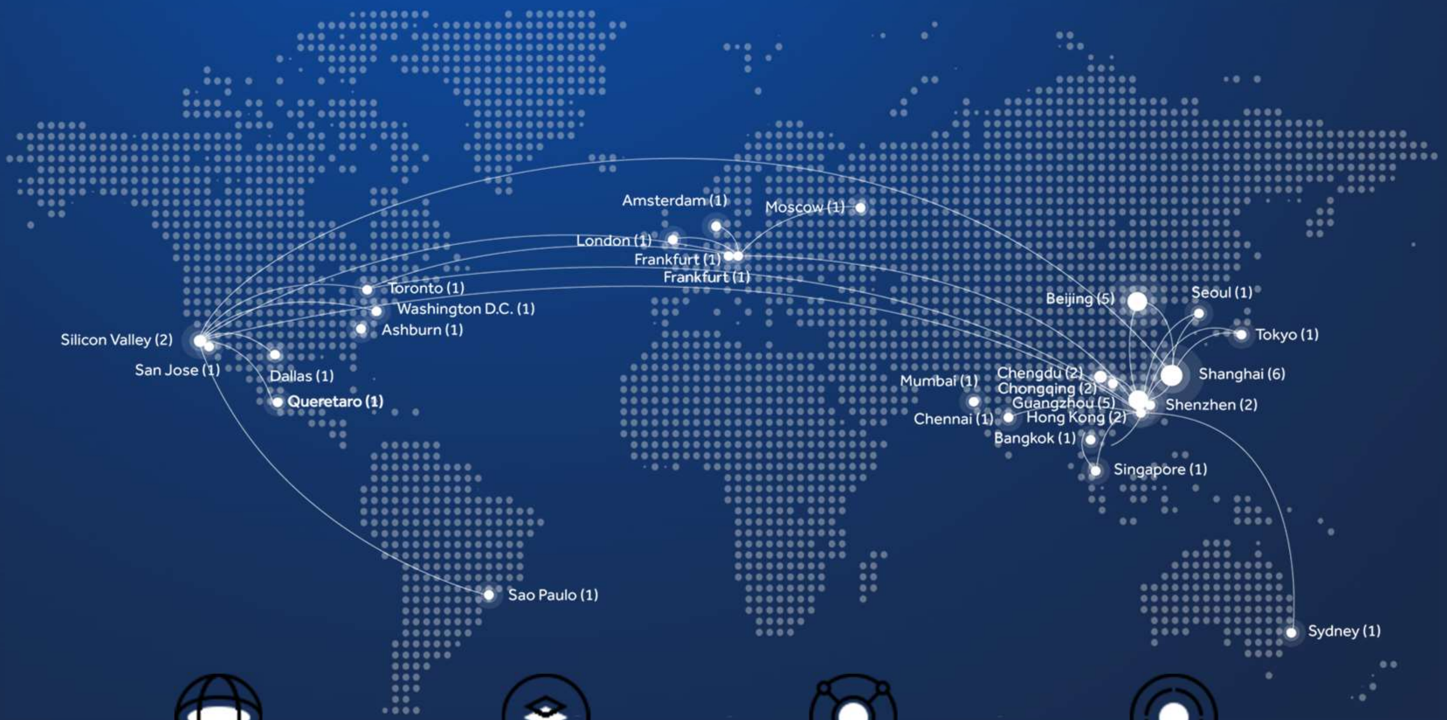
25 Regions

53 Availability zones

1,100+ PoP

1,000,000+ Servers

1,024+ PB Storage



Powerful Network

Highly Customized

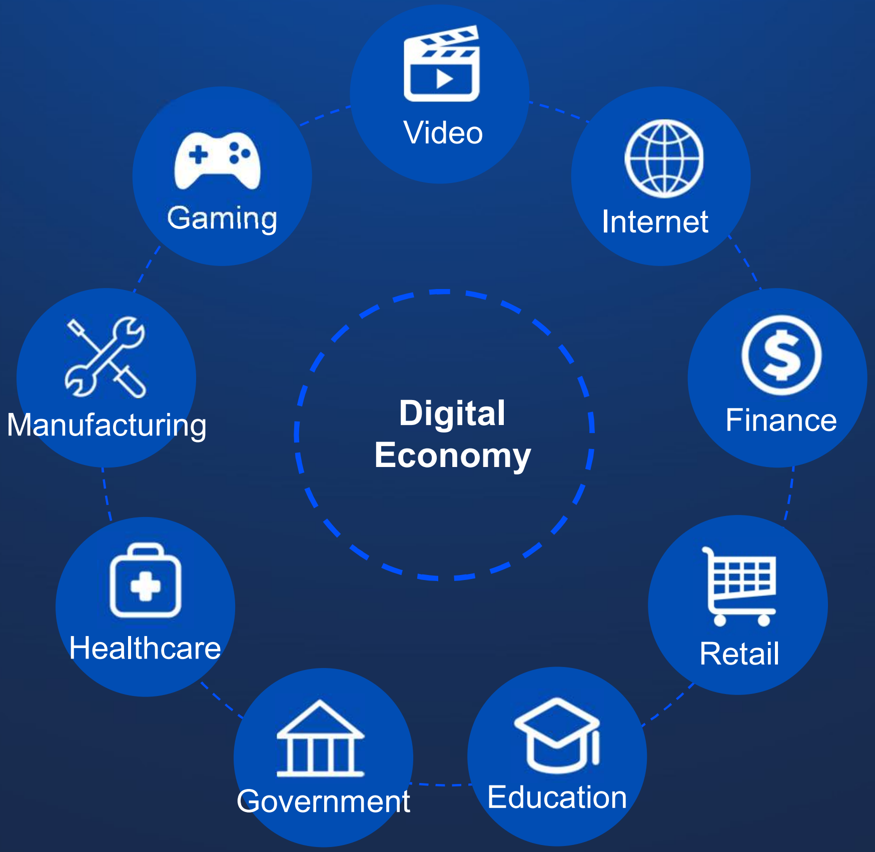
Global Coverage

Ecosystem

Tencent Cloud can serve globally with large scale of resources

Tencent Cloud

Tencent Cloud Clients



Tencent cloud has been contributing to 100000+ clients from different industries for digitalization

Background

- Aim to micro service
- Fast bootup
- massive deployment
- Short life cycle
- Less memory footprint
- High level of security
- Current solutions
    - Firecrack, crosvm, RustVMM based hypervisor

Why QEMU adaption?

Good for devops

Good for developer's knowledges

QEMU Adaption

Pro

1. Strong hardware isolation
2. Rich existing features

Con

1. Redundant code path
2. Slow guest start up within seconds
3. More resource used

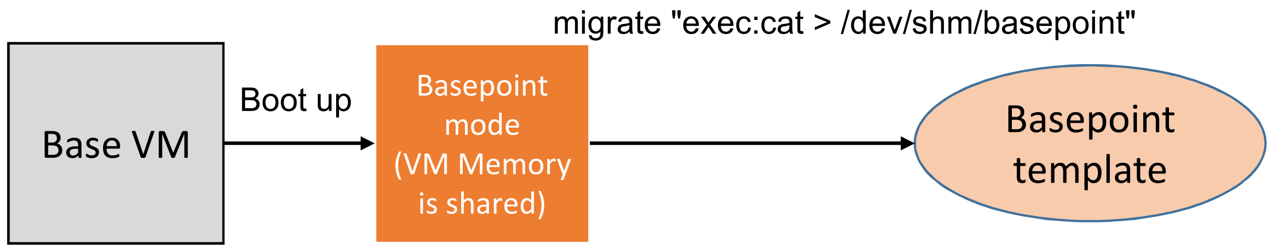
Our Solution

QEMU Basepoint

- Bypass Guest initialization
- Bypass QEMU initialization

QEMU basepoint: Bypass Guest init.

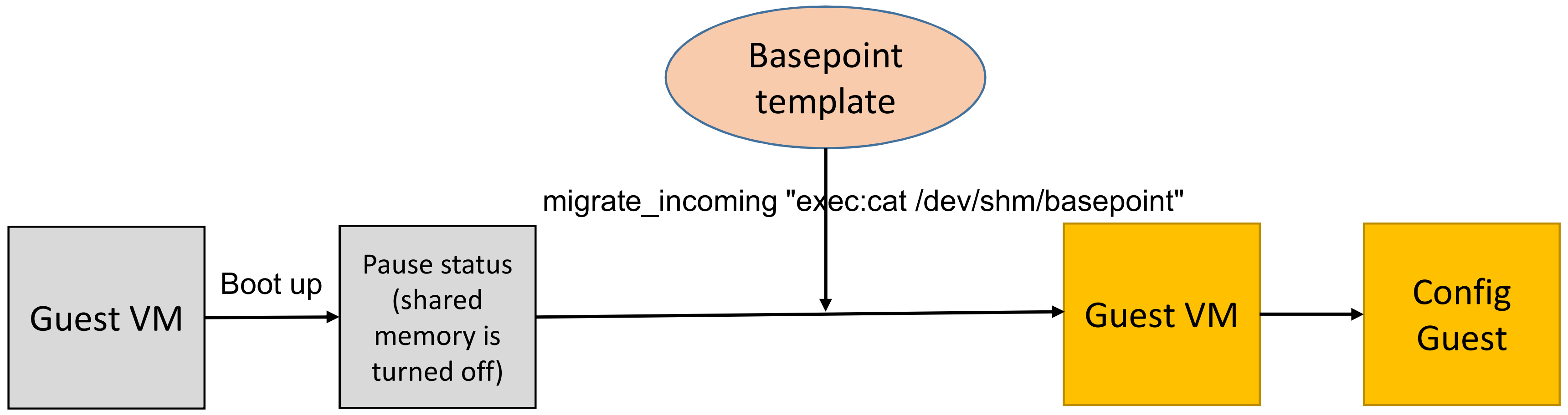
Leverage the existing migration strategy to create the basepoint template which could be used to restore the Guest instead of booting guest kernel every time.


migrate "exec:cat > /dev/shm/basepoint"
Base VM
Boot up
Basepoint mode (VM Memory is shared)
Basepoint template

- **Base on shared memory migration with minimum system resource**
- **(Inspirit from kata container)**

QEMU basepoint: Bypass Guest init. (Cont.)

Restore the guest VM from the basepoint image will significantly speed up the guest boot up which eliminates the cost for guest kernel boot up.

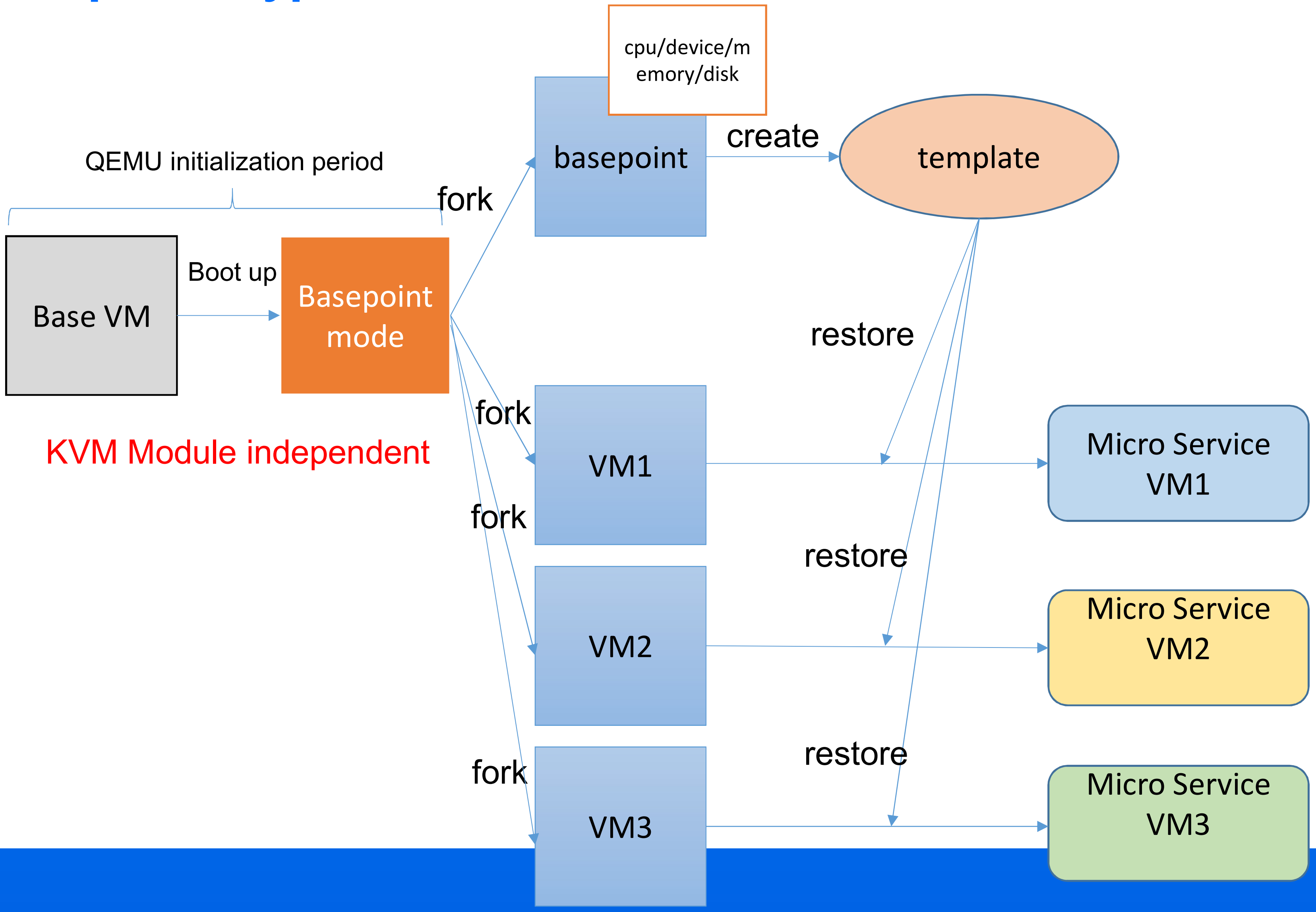


Basepoint template
migrate_incoming "exec:cat /dev/shm/basepoint"
Guest VM
Boot up
Pause status (shared memory is turned off)
Guest VM
Config Guest

QEMU basepoint: Bypass QEMU init.

- Besides skipping the guest kernel boot period, QEMU initial code will introduce appreciable latency.
- Fork the QEMU process after Base VM boots up would be able to further speed up the micro service instance start up

QEMU basepoint: Bypass QEMU init. (Cont.)



Tencent Cloud

VM startup period

	Original Qemu with Optimized linux kernel	+ Bypass kernel init	+ Bypass Qemu init
Boot up (ms)	500	150	35

Security

Tencent Cloud

Security approach

Enhance the Guest security after restore from the same template

- **Use virtio-rng as random number generator for the guest**
- **Reseed the entropy for random number generator after restore the guest from basepoint template**
- **However, kernel randomness (e.g, KASLR) still does not work…**
    - **Each tenant has a dedicated template**

Future works

Tencent Cloud

Future works

- Optimize the current QEMU code
- QEMU modulization
- Guest kernel security enhancement

Summary

1. Based on QEMU code to achieve the micro service requirement to fast deploy intensive micro services in a extremely short period.

2. With minimum modifications and easily adapt to the existing framework in public cloud.

3. Our solution, QEMU baspoint, bypass QEMU & Guest init. completely.

Tencent Cloud

Q&A

Tencent Cloud