

Enhance memory utilization for cloud providers

Zhang Yulei <yuleixzhang@tencent.com>



Agenda

- **Background**
- **Design**
- **Future works**

Tencent Cloud



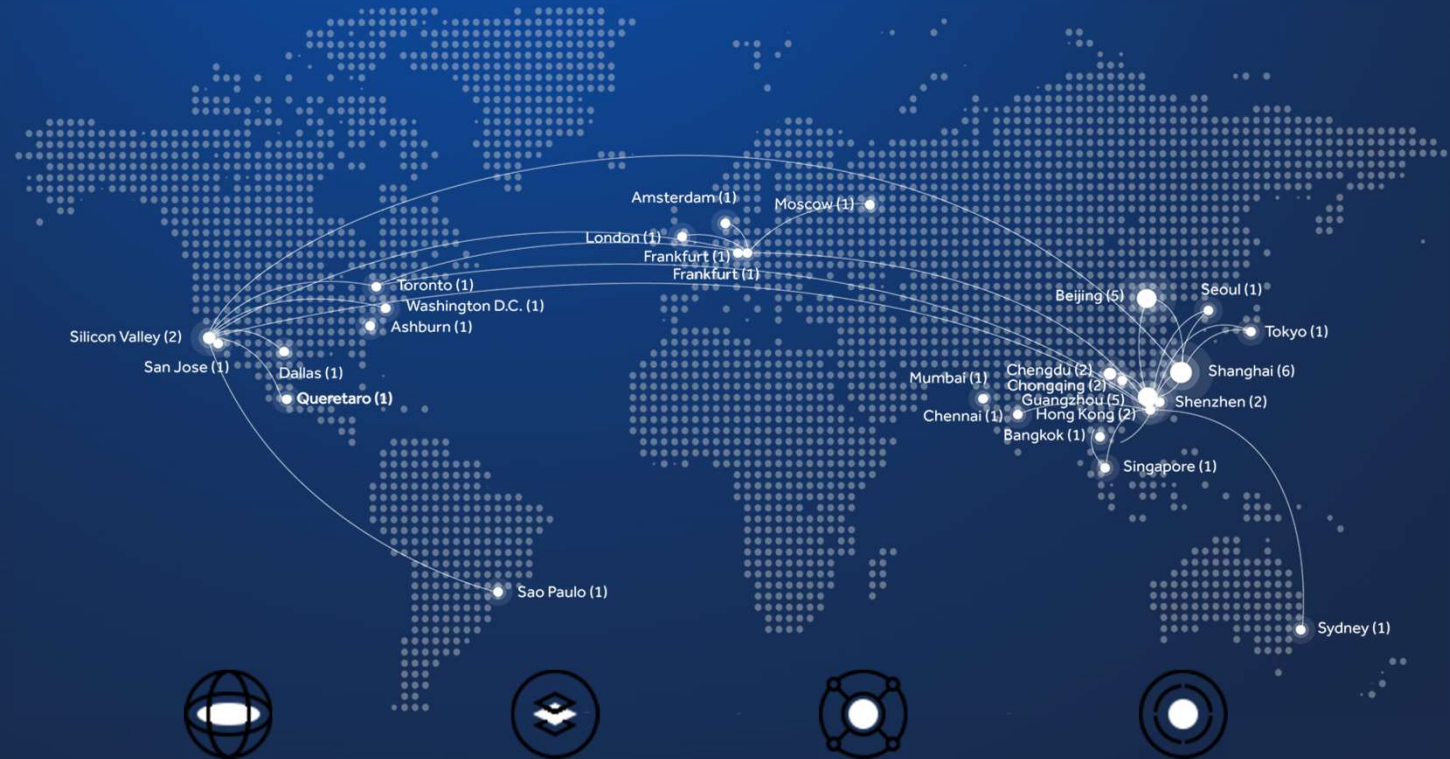
25 Regions

53 Availability zones

1,100+ PoP

1,000,000+ Servers

1,024+ PB Storage



Powerful Network



Highly Customized



Global Coverage



Ecosystem

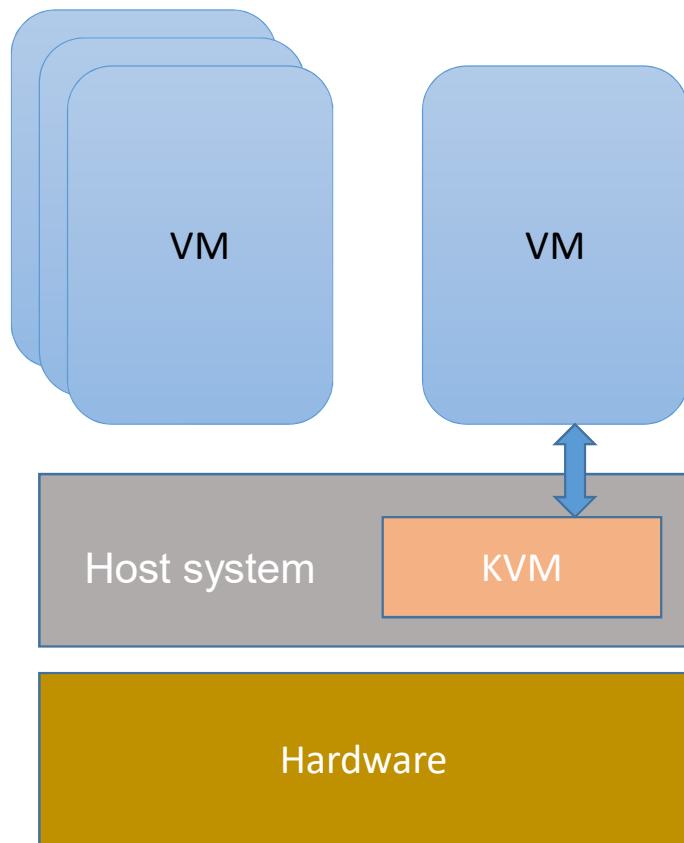
Tencent Cloud can serve globally with large scale of resources

Tencent Cloud Clients



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

Background

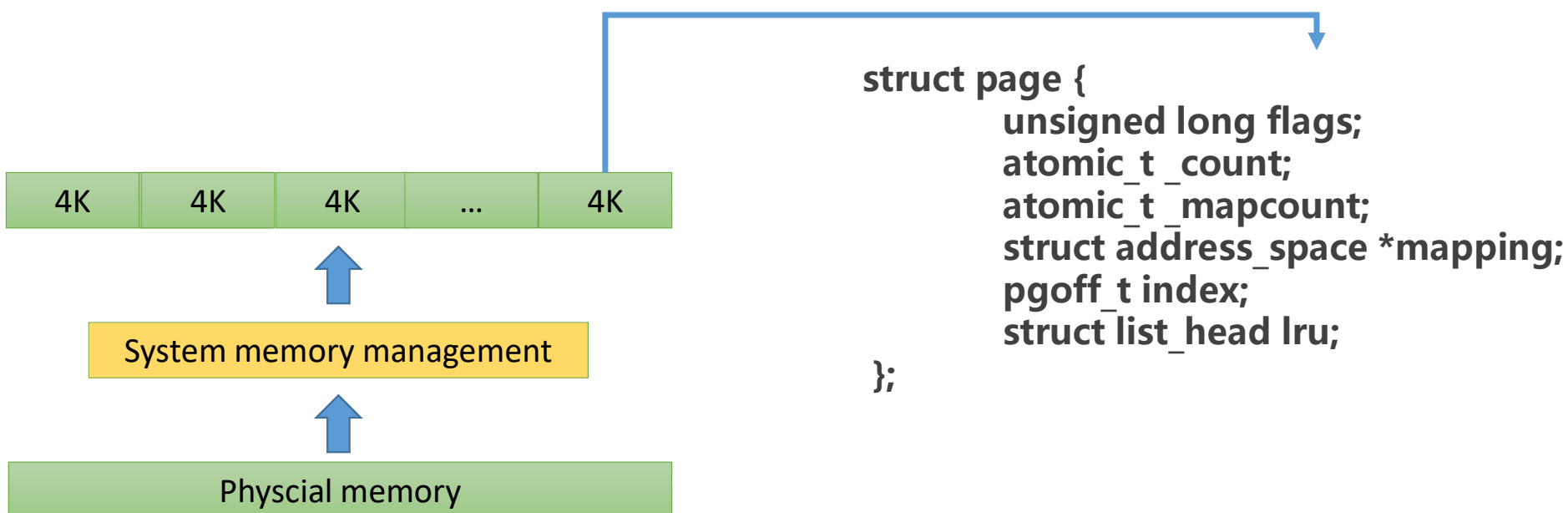


Host system memory overhead

HOST memory	Available memory
384G	375G
512G	501G
768G	753G

Background

In current system each physical memory page is associated with a page structure which is used to track the usage of this page.



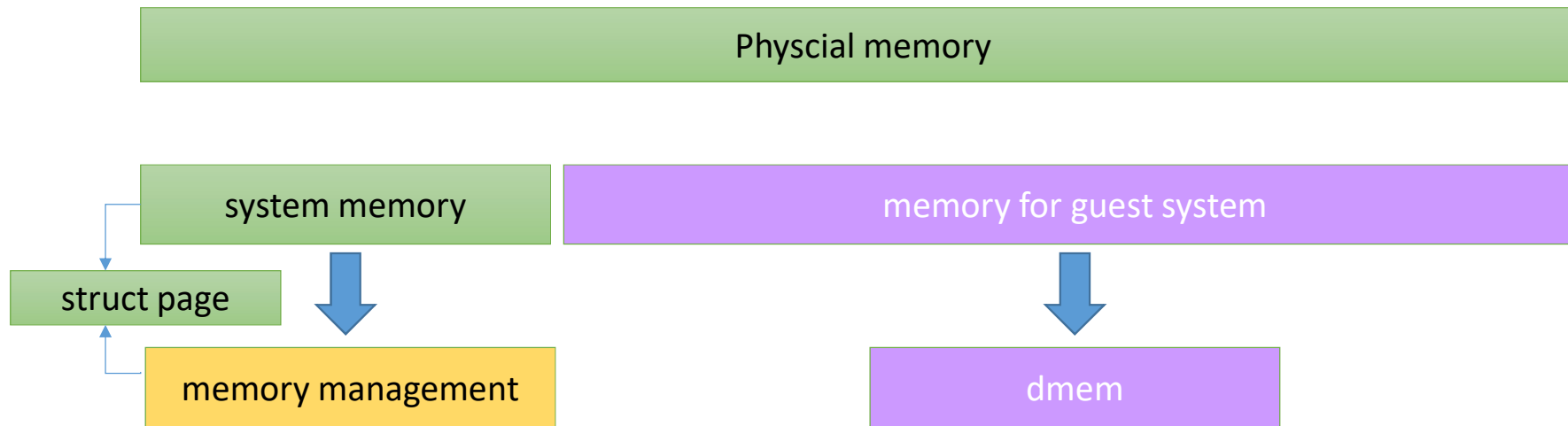
Existing approach

- kernel parameter “mem=” to reserve memory for host system
- mmap the remaining system memory with /dev/mem to user space for usage which doesn't have struct-page backed

Limitation for existing approach

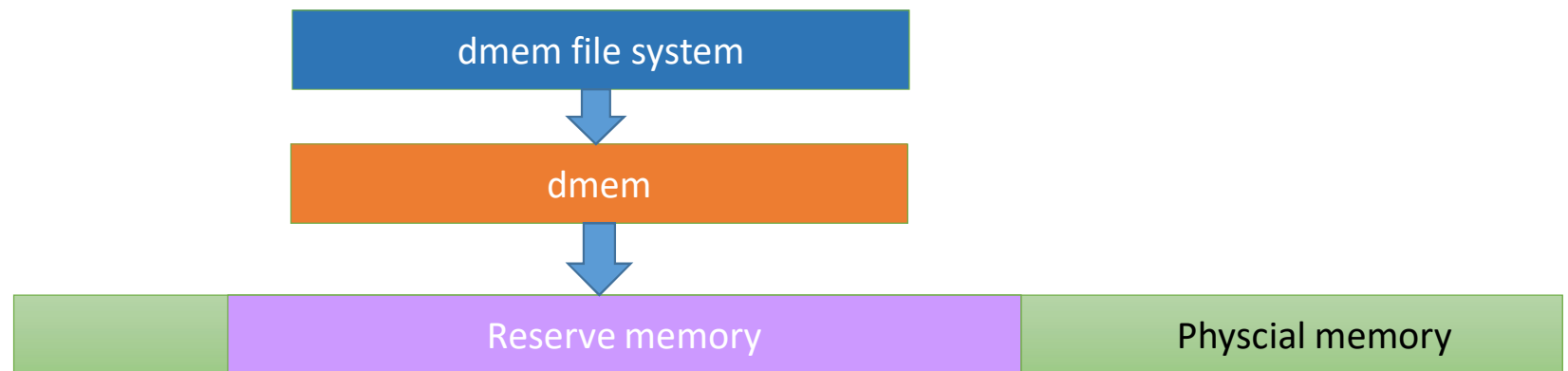
1. Access to `/dev/mem` is restricted due to the security requirement, but usually applications are unprivileged processes.
2. what we get from `/dev/mem` is a whole block of memory, as dynamic applications running on `/dev/mem` will cause memory fragment, it needs additional logic to manage the allocation and recovery to avoid wasted memory.
3. Can't support hugepage with different page size granularity.
4. MCE recovery is missing.

Our proposal – Dmem(direct memory management)



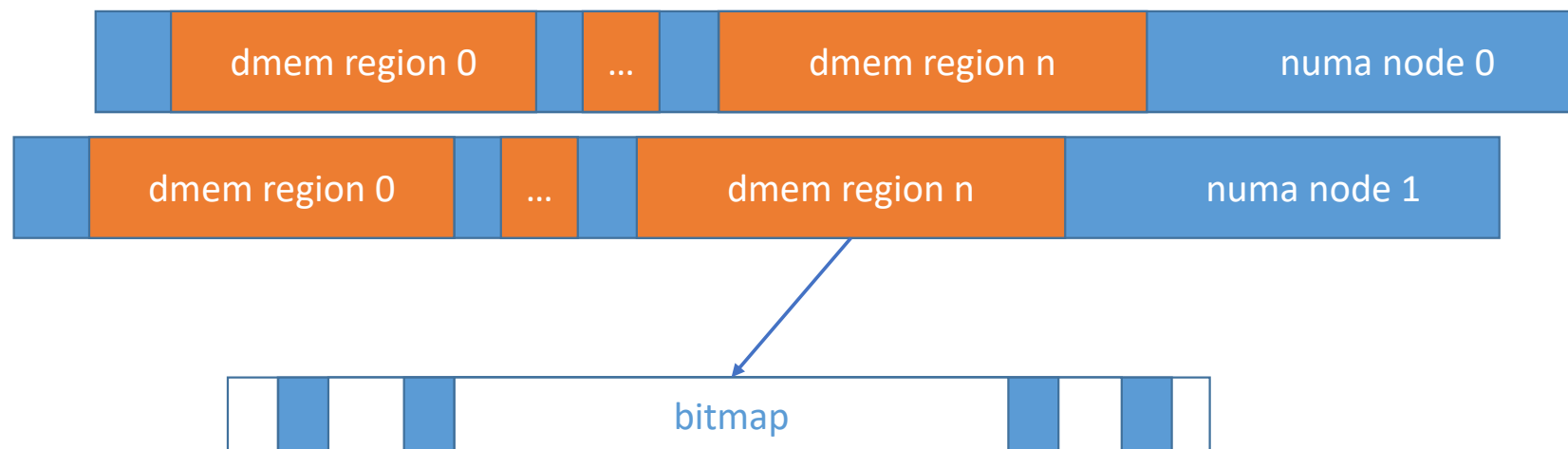
Framework Overview

1. Reserve the memory from host kernel when system boots up with kernel parameter `dmem=size`
2. Dmem will manage the allocation and recovery
3. Dmemfs provide interface to access the reserved memory region

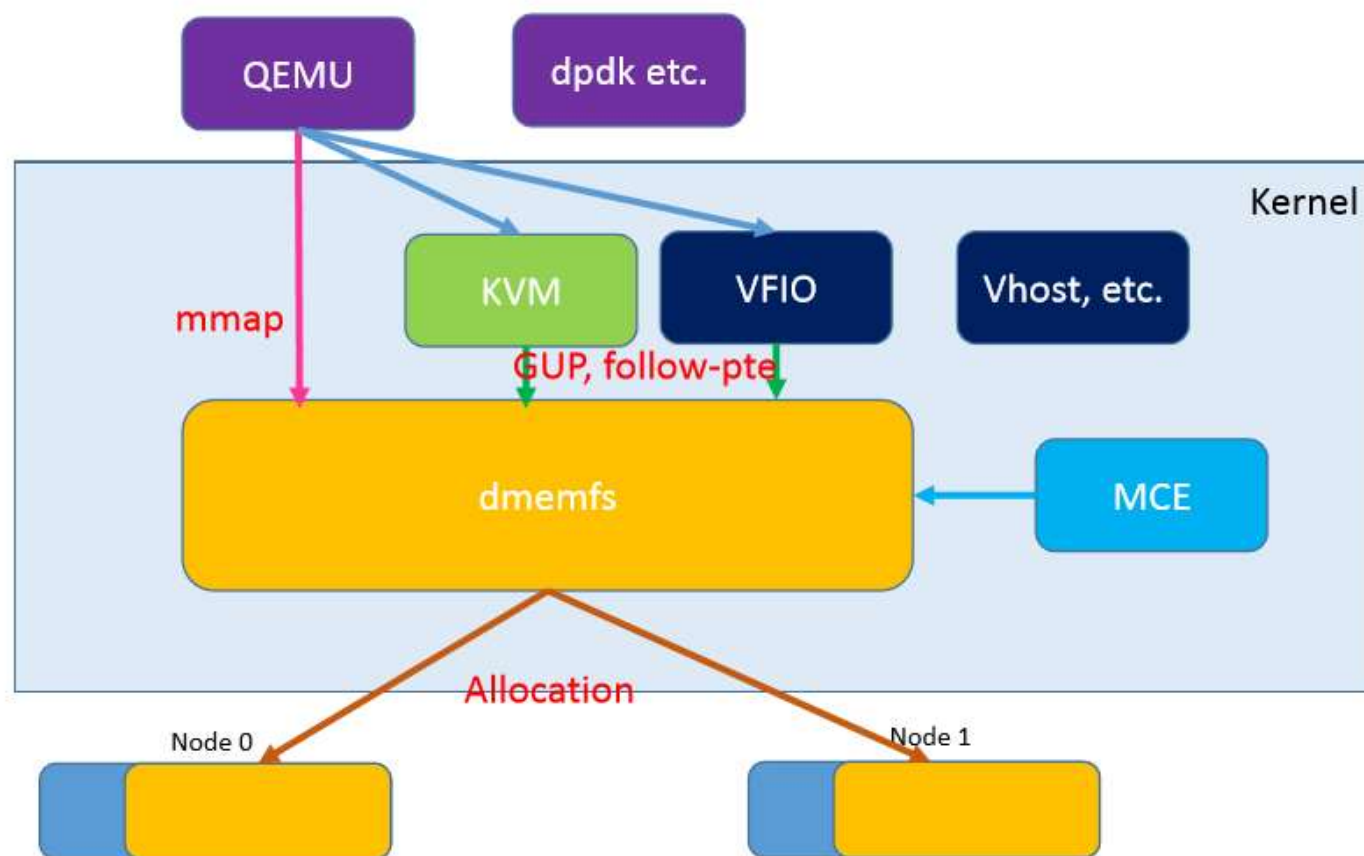


Memory management

Manage the allocation and recovery in `dmem_pool` -> `node-n` -> `region-n`



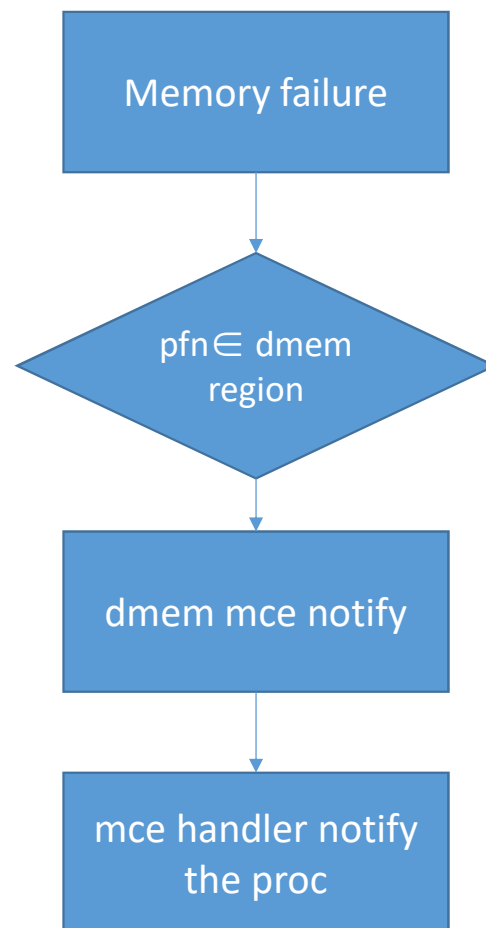
Overview for Direct Memory Management file system



Dmemfs Introduce

- Support mmap, mkdir and create regular files
- Hugepage support
 - 2M, 1G
- Numa allocation
 - MPOL_PREFERRED
 - MPOL_BIND
- Debugfs
- MCE

MCE support



Usage

Dmemfs supports mapping ``4K``, ``2M`` and ``1G`` size of pages to the userspace, for example :

```
# mount -t dmemfs none -o pagesize=4K /mnt/
```

Then it can create the backing storage with 4G size :

```
# truncate /mnt/dmemfs-uuid --size 4G
```

To use as backing storage for virtual machine starts with qemu, just need to specify the memory-backed-file in the qemu command line like this :

```
# -object memory-backend-file,id=ram-node0,mem-path=/mnt/dmemfs-uuid \  
share=yes,size=4G,host-nodes=0,policy=preferred -numa node,nodeid=0,memdev=ram-  
node0
```

Benefit with dmem

`sizeof(struct page) = 64 byte`, for dmem manage 400g reserved memory

totally save $64 * 100 * 2^{20} \approx 6.25\text{G}$ physical memory

Link to the patchset

<https://lkml.org/lkml/2020/10/8/139>

Future works



Future works

- **Dmemfs region dynamic adjustment**
- **COW(copy on write) support**
- **Upstream the feature**

Q&A

