Speed Up Boot-up Time for Guest in Alibaba Cloud

Li Weinan william.lwn@alibaba-inc.com
Guo Cheng hanyu.gc@alibaba-inc.com
Agenda

- Background
- Async dma map
- Guest boot process with async dma map
- Optimization design
- Achievements
What is the problem?

- Dma_map all the guest memory when there is passthrough device

- 8G→384G
  Dma_map time is one big problem!
Guest boot & QEMU initialization time vs memory size

VM Boot Time

QEMU Initialization Time

Time/s

VM Memory Size/G

Time/s

VM Memory Size/G
Conditions

- More time costs along with more memory
- No DMA no dma_map
- DMA specific range memory
Options

- virtual IOMMU

- Async dma map
  - Only map necessary memory first
  - Map asynchronously in the background
Async dma_map

★Balloon memory before allocated for DMA

Overview of memory access with a passthrough device
Architecture Overview

- **QEMU**
  - Trigger vfio_dma_map
  - Trigger balloon change
  - Track ballooned pages

- **virtio_balloon driver**
  - Balloon pages
  - Tell to host (QEMU)

- **VFIO driver**
  - do vfio_pin_map_dma
virtio_balloon communication

Related functions and struct
- inflate_vq
- deflate_vq
- virtio_balloon_handle_output
- VirtQueueElement:
  - guest PFN
  - page_num
Balloon range tracking workflow

Inflate process

Deflate process
Guest boot process with async dma map

<table>
<thead>
<tr>
<th>Phase 1: Initialization</th>
<th>Phase 2: dma_map asynchronously</th>
<th>Phase 3: Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init guest balloon size</td>
<td>Dma_map non-ballooned ranges</td>
<td>Trigger deflate balloon</td>
</tr>
<tr>
<td>dma_map IOVA ranges below 4G</td>
<td>Generate ballooned ranges table</td>
<td>balloon empty?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update ballooned ranges table</td>
</tr>
<tr>
<td>Load virtio_balloon driver</td>
<td>Dma_map released ranges</td>
<td>deflation notification</td>
</tr>
<tr>
<td>Balloon_target</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fill balloon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balloon completed?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inflation notification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load passthrough device driver</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balloon_target</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Optimization design

Auto-combination

Problem:
- one page at a time
- 256 pages per cycle

Practice:
✓ Combine adjacent pages

- Most of the memory ranges are adjacent
- dma_map after inflate balloon process finished
Optimization design

Increase balloon page size

```c
#define VIRTI0_BALLOON_ARRAY_PFNS_MAX 256

struct page *balloon_page_alloc(void)
{
    struct page *page = alloc_page(balloon_mapping_gfp_mask() | __GFP_NOMEMALLOC | __GFP_NORETRY | __GFP_NOWARN);
    return page;
}
```

- 4K page is too small which will import heavy but unnecessary communication between guest and host

- 4KB -> 2MB, one virtio talk can in/deflate 2 x 256 = 512MB memory
Optimization design

✓ Pre-map to perform dma_map

• Asynchronously dma_map can start early independent of deflating notification

• Insert new dma_map range if the released pages beyond mapped ranges
Achievements

QEMU Initialization Time

Time/s

VM Memory Size/G

async_dma_map

normal_boot
Achievements

![Graph showing VM Boot Time vs VM Memory Size/G for async_dma_map and normal_boot. The graph indicates a significant increase in boot time as memory size increases, with async_dma_map having a slightly higher boot time compared to normal_boot.]
Q&A
Li Weinan
william.lwn@alibaba-inc.com

Guo Cheng
hanyu.gc@alibaba-inc.com