Boosting GPU Virtualization Performance with Hybrid Shadow Page Tables

Yaozu Dong
Mochi Xue
Xiao Zheng
Jiajun Wang
Zhengwei Qi
Haibing Guan

Shanghai Jiao Tong University
Intel Corporation
GPU Usage

• Gaming (2D/3D graphic)
• HD video hardware decoding
• High performance computing

High Performance Computing shifts computation-intensive workloads to cloud environment.

- Machine learning
- Molecular dynamics simulations
- Media transcoding

A new computing paradigm: GPU Cloud
**gHyvi**

An optimized GPU virtualization scheme based on gVirt.

- **GMedia:** A media transcoding benchmark
- **Relaxed Shadow Page Table**
- **Adaptive Hybrid Page Table Shadowing Policies**

Up to **13x** performance of gVirt and **85%** of native.
GPU benchmarks

2D/3D graphic
For OpenGL and DirectX commands
Such as 3DMark, PassMark

GPGPU
For CUDA and OpenCL commands
Such as Rodinia [ATC 2013], Parboil
GMedia

A hardware media transcoding benchmark based on Intel’s MSDK (Media Software Development Kit).

- MSDK provides APIs for hardware acceleration.
- A wrapper invokes media functions form MSDK.
- Test cases can run with assigned threads and settings.
- The FPS results reflect the performance.
Massive update issue

93% performance degrade.
Shadow page table

- **512 * 1024 * 4KB = 2GB**
- Resource partition
- User space isolation
Strict shadow page table

Trap-and-emulation

Write protect the page table  Page fault happens  Update shadow page table

Page table

Shadow page table

gVirt
Case profiling

Proportion changes
21.43% → 79.45%

VM_EXIT

5, 720p  7, 720p  15, 720p  3, 1080p  4, 1080p  10, 1080p

- EXCEPTION_NMI
- EXTERNAL_INTERRUPT
- VIRT_INTR
- VMCALL
- I/O_INSTRUCTION
- PAUSE
- APIC_ACCESS
- EPT_VIOLATION
Breakdown of EPT_VIOLATION

Take “15thread-720p” as an example:

- 62% of all VM_exit is due to EPT_VIOLATION
- 83% of EPT_VIOLATION is caused by PTE access
Frequency

The frequency of update in 10s.

PTE access frequency

- Up to 7.5k times
Hot page tables

![Diagram showing PTE access pattern (in 10s)]

- **PTE index**
  - 0 to 512
- **CPU cycle**
  - 0 to 10

**Legend:**
- Red plus signs represent hot page table accesses.
Continuous updates

PTE access pattern (in 0.02s)

CPU cycle

PTE index

Same table
Conclusion of massive update issue

VM is frequently swapping graphic memory pages. It modifies the entries of page table massively.

- Large amount of updates (7.5k in 10s)
- Updates focus on certain pages (hot pages)
- Updates are continuous (on the same page)

Modifications lead to busy trap-and-emulations. Eventually, overhead happens. (Up to 95%)
gHyvi architecture

Host VM
- gVirt
- Native GFX Driver
  - Page Table

Guest VM
- Native GFX Driver
  - Page Table

VMM
- gHyvi
  - Strict SPT
  - Relaxed SPT

GPU

- Hyper call
- Pass Through
- Trap
The commands fed by CPU won’t take effect until they are fetched by the GPU.
Relaxed shadow page table

Remove write protection

Page table

Shadow page table

gHyvi
Relaxed shadow page table

Page table

Shadow page table

gHyvi
Relaxed shadow page table

Page table

gHyvi

Shadow page table

Reconstruct
Page table reconstruction

**Step 1:** Take a snapshot of guest page table

```
Guest

Page table

Host

Strict shadow page table

Snapshot
```
Step 2: Massive update

Page table reconstruction

- Page table
- Strict shadow page table
- Snapshot
Step 3: Compare with snapshot
Step 4: Reconstruct the different part

- **Guest**: Page table
  - Relaxed shadow page table

- **Host**: Snapshot
4 Policies for Performance Tuning
Static partial relaxed

Select 50, 100, 200, 300 hot pages
Switch them into relaxed mode
Dynamic partial relaxed

- All the pages are in strict mode at first
- Switch to relaxed mode once it’s touched
Dynamic segmented partial relaxed

- All the pages are in strict mode at first
- Switch to relaxed mode once it’s touched
- Reset the relaxed pages after reconstruction
Evaluation

4 policies of page table shadowing

• Full reconstruction
• Static partial reconstruction
• Dynamic partial reconstruction
• Dynamic segmented partial reconstruction

Linux 2D/3D performance

Windows 2D/3D performance
## Configuration

### Hardware

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>4th Intel Haswell i5 (2.4Ghz)</td>
</tr>
<tr>
<td>RAM</td>
<td>8GB</td>
</tr>
<tr>
<td>HDD</td>
<td>Seagate 500GB</td>
</tr>
<tr>
<td>GPU</td>
<td>Intel Processor Graphics&lt;br&gt;With 2GB global graphics memory</td>
</tr>
</tbody>
</table>

### Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux VM</td>
<td>64bit Ubuntu 12.04</td>
</tr>
<tr>
<td>Windows VM</td>
<td>64bit Windows 7</td>
</tr>
<tr>
<td>Xen</td>
<td>4.3</td>
</tr>
<tr>
<td>VM configuration</td>
<td>4 VCPUS&lt;br&gt;and 2GB system memory</td>
</tr>
</tbody>
</table>
Evaluation: full relaxed

The performance of normal cases are **degraded**.

The performance of issue cases are **improved**.
Evaluation: static partial relaxed

The performance of normal cases becomes worse with more relaxed pages.

The performance of issue cases are improved. The coverage of hot pages affects the performance.
Evaluation: dynamic relaxed

Dynamic segmented partial works fine on normal cases.

Up to 13x of gVirt
85% of native
Linux 2D/3D performance

Slightly better than gVirt.
Windows 2D/3D performance

Discrepancy is acceptable.

![Bar chart showing performance comparison between SM2.0, HDR/SM3.0, Pass2D, Heaven, and Heaven fps categories. The chart indicates that the discrepancy is acceptable across all categories.]
Summary

An optimized GPU virtualization scheme based on gVirt.

- New shadow page table: relaxed shadow page table
- Adaptive hybrid page table shadowing policies

Up to 13x performance improvement.

Source code is available
https://01.org/igvt-g
Q&A

Boosting GPU Virtualization Performance with Hybrid Shadow Page Tables