Automatic Accelerator Virtualization via Language and Compiler Support
Hangchen Yu, Arthur M. Peters, Amogh Akshintala, Christopher J. Rossbach
The University of Texas at Austin University of North Carolina at Chapel Hill VMware Research Group

**Technology Trend**

Hypervisor support falls behind accelerators’ proliferation

**Silos Complicate Virtualization**

- API remoting: interposition, compatibility
- Para-virtual I/O: complexity, compatibility e.g. SVGA translates guest interactions into DirectX
- Full-virtualization: significant overheads by trap-based interposition
- SRIOV: remains lacking by hardware support (< 0.95% NVIDIA GPUs)

**Applications**

- **CPU**
- **GPU**
- **ASIC**
- **NVM**
- **FPGA**
- **CRYPTO**

**Hypervisor**

**Virtualized Accelerators**

<table>
<thead>
<tr>
<th>API</th>
<th>Gen</th>
<th>#</th>
<th>LoC</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenCL</td>
<td></td>
<td>39</td>
<td>7514</td>
<td>NVIDIA GTX 1080</td>
</tr>
<tr>
<td>CUDA 10</td>
<td>√</td>
<td>16</td>
<td>266</td>
<td>AMD RX 580</td>
</tr>
<tr>
<td>CUDART 10</td>
<td>√</td>
<td>93</td>
<td>1358</td>
<td>NVIDIA GTX 1080</td>
</tr>
<tr>
<td>TensorFlow 1.14</td>
<td></td>
<td>111</td>
<td>1865</td>
<td></td>
</tr>
<tr>
<td>NCSDK v2</td>
<td>√</td>
<td>26</td>
<td>479</td>
<td>Movidius NCS</td>
</tr>
<tr>
<td>QuickAssist</td>
<td>√</td>
<td>19</td>
<td>444</td>
<td>Intel QAT 8970</td>
</tr>
</tbody>
</table>

AvA has supported 10 accelerators and 12 framework APIs

**Selected Evaluation Results**

- **Unfairness**
  - Fixed (P=0.5s): 7.5%
  - Fixed (P=1s): 2.6%
  - Adaptive (P=0.5s): 7.4%
  - Adaptive (P=1s): 2.4%

**Specification Example**

```c
ava_throughput_resource memcpy_thp;
CUDAResult cudaMemcpyHostToDevice
  (cudaMemcpy(dstDevice, srcHost, size_t byteCount) {
    ava_argument(dstDevice) ava_handle;
    ava_argument(srcHost) {
      ava_in; ava_buffer(byteCount);
    } ava_consumes_resource(memcpy_thp, byteCount);
  }
```