NIGHTWATCH: Remoting Accelerator APIs through the Hypervisor

Hangchen Yu, Amogh Akshintala, Arthur Peters, Christopher J. Rossbach

The University of Texas at Austin University of North Carolina at Chapel Hill

VMware Research Group

Accelerator Stacks are Silos
- Hardware Interface: MMIO, mmap’d command queues
- Software Interface: vendor-specific drivers, proprietary protocols

Silos Complicate Virtualization

Automatic Generation

Development Effort

<table>
<thead>
<tr>
<th>API</th>
<th># of API</th>
<th>Lines of Spec</th>
<th>LOC (generated)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenCL</td>
<td>88</td>
<td>4,200</td>
<td>4,150</td>
<td></td>
</tr>
<tr>
<td>CUDA</td>
<td>211</td>
<td>6,350</td>
<td>8,150</td>
<td></td>
</tr>
<tr>
<td>TensorFlow</td>
<td>160</td>
<td>5,350</td>
<td>6,900</td>
<td></td>
</tr>
<tr>
<td>MVNC API</td>
<td>25</td>
<td>910</td>
<td>2,450</td>
<td></td>
</tr>
</tbody>
</table>

A handful of days

Example

```
// marshals data to accelerator shared buffer
switch (guest_base->api_id) {
    case CU_MEMCPY_H_TO_D:
        COPY_FROM_GUEST(arg1_0);
        COPY_FROM_GUEST(arg2_0);
        break;
}
```

LibForward

```
CUresult CUDAPI cuMemcpyHtoD(
    CUdeviceptr dstDevice,
    const void* srcHost,
    size_t byteCount)
{
    INIT_CUDA_PARAM(param);
    param.base.api_id = CU_MEMCPY_H_TO_D;
    // set arguments
    param.arg3_0 = dstDevice;
    param.arg2_0 = byteCount;
    param.arg1_0 = srcHost;
    // compute data size
    param.base.dpool_size += COMPUTE_SIZE(
        param.arg3_0, byteCount);
    // forward call to guest driver
    IOCTL_TO_DRIVER(&param);
    return param.ret_arg0;
}
```

Evaluation

- Compatibility recovered: stack generation is automatic
- Interposition recovered: APIs forwarded over VMM-managed transport
- Near-native performance
- Scales to 16 VMs
- Fair scheduling with heterogeneous workloads
- Migration overhead < 40 ms
- Slowdown under memory over-subscription < 2.5x