Shared Memory Optimization in Virtualized Cloud
Qi Zhang and Ling Liu, Georgia Institute of Technology

Network I/O in virtualized cloud

Disadvantages of co-located inter VM communication via default network stack
- Long communication path
- Multiple data copies
- Context switch between VMs and the host machine

Current shared memory approaches

Current shared memory approaches

[Jian Wang, et al. HPDC 2008]
- Different layers where shared memory is established
  - User libraries and system calls level
  - Above the transportation level
  - Below the IP level

Problems of current approaches

Problems of current approaches

- Static shared memory allocation
  - Inflexibility
  - Low resource utilization
  - Restricted application performance
- Allocating shared memory from inside the VM
  - Increasing VM memory pressure
  - VM crash propagation

Dynamic memory allocation from host

Dynamic memory allocation from host

- Allocating shared memory from the host
- On-demand shared memory allocation to each VM
- Interfaces to VM kernel
  - struct shm* shm_init(size_t size)
  - void shm_exit(void *addr)
  - size_t shm_put(void *buf, size_t offset, size_t len)
  - Etc.

Case study 1: MemPipe

MemPipe is a co-located inter VM communication system built on top of the host machine based dynamic managed shared memory.

MemPipe performance

MemPipe performance

Case study 2: MemFlex

MemFlex is an efficient VM memory swapping system built on top of the host machine based dynamically managed shared memory.

MemFlex performance

MemFlex performance

Table 2: VM disk I/O performance measured by Sysbench

<table>
<thead>
<tr>
<th>S/R</th>
<th>Sequential</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Read (GB/s)</td>
<td>Write (KB/s)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>3.89</td>
<td>19061.90</td>
</tr>
<tr>
<td>MemFlex</td>
<td>3.96</td>
<td>62444.54</td>
</tr>
</tbody>
</table>

(a) Execution time (lower is better) (b) Throughput (higher is better)