Differential Virtual Time (DVT): Rethinking I/O Service Differentiation for Virtual Machines

Mukil Kesavan, Ada Gavrilovska, Karsten Schwan
Background

• Cloud customers – QoS Requirements
• Need to provide service differentiation
• CPU & Mem Sharing – Widely Supported
• Is I/O Sharing Straightforward?
  – Do Existing Approaches Apply Cleanly?
I/O Service Differentiation Challenges

- Work Conserving Scheduling
- Bursty I/O
- High VM Consolidation

Randomly Varying Service Latencies over Time in VM
Fn(Device Delay, Other VMs’ I/O Activity)

- VMs != Processes
  - Internal resource mgmt (TCP, Disk Sched)
False TCP Congestion

- Lower Data Transfer Rates: Lower Avg. CWND
- Varying Service Levels => RTT Jitter = Congestion
Deceptive Idleness in VM Disk I/O

- Writes Starving Reads
- Anticipation Fails: Rand. Latency Variation

Read time in log scale

- Noop (RO)
- Noop
- Deadline
- Anticipatory
- CFQ

Dom0 (NoDelay)  DomU (NoDelay)  DomU (delay = Rand(20, 50) ms)
Key Insights

• Service Latency
  – Key to VM Res. Mgmt
  – Variation: Ignored by Curr. Sched. Methods

• Insufficient Performance Isolation

• Inability to Differentiate I/O Service Levels

• Complicates Cloud Adoption & Billing
Differential Virtual Time (DVT)

- Desirable Service Latency Properties for I/O Sharing

  - P1: Gradual Service Latency Change
  - P2: Principle of Conservation of DVT

- P1 => Maintain Isolation (Friendly to RM)
- P2 => Maintain Proportionality
DVT Illustration

Service Latency

Time

VM I/O at Regular Intervals

DVT Dilation: P1

DVT Conservation: P2
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

![Diagram showing epochs and VM states]

**I/O Service Level**

0 — 100

**VM States:**
- VM W
- VM X
- VM Y
- VM Z

**Epochs:**
- **Epoch m:** Active: X
  - r1
  - r2
  - r3
  - r4

- **Epoch n:** Active: W, X, Y, Z
  - r1
  - r2
  - r3
  - r4

- **Epoch o:** Active: X
  - r1
  - r2
  - r3
  - r4
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

I/O Service Level

![Diagram showing active VMs and I/O service levels across epochs.]

VM W
VM X
VM Y
VM Z
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

I/O Service Level

| VM W | | | |
| VM X | | | | |
| VM Y | | | | |
| VM Z | | | | |

Service Level Dropped to 75% instead of 25%
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

I/O Service Level

0 ——— 100

VM W
VM X
VM Y
VM Z

Epoch $m$
Active: X
r1 r2 r3 r4

Epoch $n$
Active: W, X, Y, Z
r1 r2 r3 r4

Epoch $o$
Active: X
r1 r2 r3 r4
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

Service Level increased to 100% right away
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

I/O Service Level

0 —— 100

<table>
<thead>
<tr>
<th>VM</th>
<th>Active</th>
<th>Epoch m</th>
<th>Epoch n</th>
<th>Epoch o</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td></td>
<td>Active: X</td>
<td>Active: W, X, Y, Z</td>
<td>Active: X</td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>r1 r2 r3 r4</td>
<td>r1 r2 r3 r4</td>
<td>r1 r2 r3 r4</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization
LS-IO Scheduling

- Built on top of WF2Q+ Algorithm
- Round: Basic Proportional Service
- Epoch: Multiple Rounds, DVT Realization

I/O Service Level
0 100

VM W
VM X
VM Y
VM Z

Epoch $m$
Active: X
r1 r2 r3 r4

Epoch $n$
Active: W, X, Y, Z
r1 r2 r3 r4

Epoch $o$
Active: X
r1 r2 r3 r4
Throughput Improvement with DVT

- Isolates Streaming (iperf) from Bursty (web)
- No loss of Performance For Web Server

![Vegas: Throughput Improvement vs. Tunables](chart1)

![Compound: Throughput Improvement vs. Tunables](chart2)
Avoid Unnecessary Congestion Collapses
Avoid Unnecessary Congestion Collapses
DVT Perf for Diff App Mixes (Vegas)

- Effective against most Bursty Profiles

<table>
<thead>
<tr>
<th>Workload Mix</th>
<th>VM (Weight)</th>
<th>WF2Q+ Thput</th>
<th>LS-IO Thput</th>
<th>Thput Change</th>
<th>(RPE, DF, TH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Streaming</td>
<td>Iperf1 (1)</td>
<td>26.8</td>
<td>31.1</td>
<td>16%</td>
<td>(10, 10, 15)</td>
</tr>
<tr>
<td></td>
<td>Iperf2 (1)</td>
<td>26.8</td>
<td>30.7</td>
<td>14%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web (6)</td>
<td>75.5</td>
<td>73.7</td>
<td>-2%</td>
<td></td>
</tr>
<tr>
<td>High Streaming</td>
<td>Iperf1 (1)</td>
<td>13.1</td>
<td>14.8</td>
<td>13%</td>
<td>(10, 10, 7)</td>
</tr>
<tr>
<td></td>
<td>Iperf2 (1)</td>
<td>13.1</td>
<td>15.1</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Iperf3 (1)</td>
<td>13.1</td>
<td>15.9</td>
<td>21%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web (9)</td>
<td>81.4</td>
<td>81.5</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Streaming &amp; Bursty</td>
<td>Iperf1 (1)</td>
<td>34.8</td>
<td>44.9</td>
<td>29%</td>
<td>(10, 10, 7)</td>
</tr>
<tr>
<td></td>
<td>Iperf2 (1)</td>
<td>34.8</td>
<td>45.1</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web1 (1)</td>
<td>42.4</td>
<td>43.3</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web2 (1)</td>
<td>42.4</td>
<td>42.3</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>High Bursty</td>
<td>Iperf1 (1)</td>
<td>39.8</td>
<td>48.2</td>
<td>21%</td>
<td>(10, 10, 7)</td>
</tr>
<tr>
<td></td>
<td>Web1 (1)</td>
<td>23.7</td>
<td>23.7</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web2 (1)</td>
<td>23.7</td>
<td>23.7</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Web3 (1)</td>
<td>23.6</td>
<td>23.7</td>
<td>0%</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

• I/O Service Differentiation
  – VM != Process

• Rate of Change of Service Latency

• DVT & LS-IO
  – Performance Isolation Vs. Proportionality

• Results: Bursty & Streaming Network I/O Mix
Future Work

• DVT for Disk I/O
  – VM Ensembles
  – Isolating Entire Cloud Applications
  – Different Resource Sharing Scenarios
    • SAN Deployments
    • Distributed File Systems