Hardware Support for Efficient and Secure Resource Harvesting in the Cloud

YArch 2023

Jovan Stojkovic, Chunao Liu*, Muhammad Shahbaz*, Josep Torrellas
University of Illinois at Urbana-Champaign
*Purdue University
Resources are Underutilized in the Cloud!

- Azure: an average VM has 15.6% CPU utilization, 3.2 cores are most of the time idle
  - Users want to be safe for their worst-case scenario and peak load → resource overprovisioning
- Increases the cost for cloud providers, worsens sustainability issues
- Need to make use of allocated but idle cores
A Harvest VM has a minimum size for its physical resources

- It can dynamically grow and shrink beyond this minimum by stealing cores from primary VMs
A Harvest VM has a minimum size for its physical resources.
It can dynamically grow and shrink beyond this minimum by stealing cores from primary VMs.
Harvest VMs to the Rescue

- A Harvest VM has a minimum size for its physical resources
- It can dynamically grow and shrink beyond this minimum by stealing cores from primary VMs

VM1 (Regular VM)

- C1
- C2
- C3

VM2 (Harvest VM)

- C4
- C5
- C6
Harvest VMs to the Rescue

- A Harvest VM has a minimum size for its physical resources.
- It can dynamically grow and shrink beyond this minimum by stealing cores from primary VMs.

VM1 (Regular VM):
- C1
- C2
- C3

Need core back!

VM2 (Harvest VM):
- C4
- C5
- C6
A Harvest VM has a minimum size for its physical resources.

It can dynamically grow and shrink beyond this minimum by stealing cores from primary VMs.

VM1 (Regular VM)  VM2 (Harvest VM)

C1  C2  C3  C4  C5  C6
Challenges!

- Maintaining security is expensive

<table>
<thead>
<tr>
<th>Warm Execution</th>
<th>Idle</th>
<th>Flush Cache</th>
<th>Cgroup Change</th>
<th>Execution</th>
<th>Flush Cache</th>
<th>Cgroup Change</th>
<th>Cold Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular VM</td>
<td></td>
<td>Context Switch</td>
<td>Harvest VM</td>
<td></td>
<td>Context Switch</td>
<td>Regular VM</td>
<td></td>
</tr>
</tbody>
</table>
Challenges!

- Overheads not tolerable for the emerging software in the cloud – microservices and serverless computing
- Functions are short running, often in microsecond scale
- Context switch cost is from 200us (L1D flush) to 1.5ms (whole cache flush)
- In addition, cold execution is slow due to many cold cache misses
Proposal: $\mu$Harvest

- Three novel techniques:
  - Per-VM partitioned hardware request queues
  - Cache bypassing when running on a stolen core
  - Disaggregated virtual caches
Conclusion

- There are plenty of underutilized resources in the cloud
- Harvesting them in a safe and efficient manner is challenging
- Need for hardware support → $\mu$Harvest