Scaling non-elastic applications using virtual machines

Thomas Knauth, Christof Fetzer (TU Dresden)
Introduction

- efficient, application agnostic elasticity
- OS virtualization and live migration
- data stream processing
Scalability vs. elasticity

• scalable: double x \rightarrow double y

• elastic: online scalability
  • can double x, while continuing to run the system

• beg you to differ!
Non-elastic applications

- configuration is static
- new nodes cannot be added at runtime
- static state partitioning
- investigated one example application: StreamMine
Event stream processing

• provide high throughput, low latency continuous query execution with bounded data access

• must be fast - avoid touch disk at all cost

• perfect for migrating, because very little on-disk state
StreamMine

• data stream processing system developed at TU Dresden

• limitation: no dynamic reconfiguration
  • must deploy statically for peak throughput

• still aiming for efficiency
Application setup

**source**

TCP

processing

sink

virtual machine

physical machine
Efficiency

- want to run on as few nodes as possible
- save operating costs
- achieve higher hardware utilization by consolidation
Live migration

- minimal downtime and service interruption
- depends on working set size of applications
- source and destination share stable storage
- only in-memory state is copied
- only in same subnet
Key idea

• start non-elastic application instances in virtualized OS instance

• use minimal set of physical resources when load is minimal

• use live migration to scale out/in when load in-/decreases
Key idea (in pictures)
Evaluation setup
<table>
<thead>
<tr>
<th>Latency [s]</th>
<th>Latency [us]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^3$</td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td></td>
</tr>
<tr>
<td>$10^7$</td>
<td></td>
</tr>
</tbody>
</table>

![Latency Scatter Plot](image)

**Latency**
Throughput

![Graph showing throughput over time](image)

Throughput [events/s] vs. Time [s]
Energy-efficiency

![Graph showing energy efficiency with dynamic and static lines.](image-url)
Shortcomings

• blocking senders
• crude migration strategy
Big(ger) picture

• investigate application level effects of live migration
• when is live migration beneficial
  • stateful vs. stateless VMs
  • OS vs. application level migration
• impact on data center architecture
Summary

• using live migration feature of modern hypervisors enables resource efficient application elasticity

• demonstrated using an event stream processing system

• energy savings of ~80% under minimal load