Proactive Memory Scaling of Virtualized Applications

Simon Spinner, Nikolas Herbst, Samuel Kounev
University of Würzburg, Würzburg, Germany
Xiaoyun Zhu, Lei Lu, Mustafa Uysal, Rean Griffith
VMware, Inc., Palo Alto, US

July 1st, 2015
IEEE CLOUD 2015, New York, USA
VM memory size:
- Too small → degraded performance or reduced availability
- Too large → higher overheads and costs (e.g. in public cloud)
Memory scaling

Ballooning

- Reclaim memory from VM
- Limit \( \leq \) configured

Hot-add

- Increases memory size of VM
- No restart of VM required
Challenges

- Application memory management
  - Examples: Java, MySQL
  - Optimal configuration depends on VM memory size

- Application elasticity
  - Memory settings may be statically configured
  - Restart of application may be required

- Impact of reconfiguration
  - May cause additional overheads
  - Unreliable under high memory pressure
Contributions

- **splitTs** workload forecasting method
  - Incorporates *calendar information*
    - Improved accuracy for multiple seasonal patterns
    - Forecast demand for 24 hours

- **Proactive control** of VM memory size
  - Determine maximum required memory of next day
  - Leverages *memory hot-add*
    - Schedule reconfigurations during *phases of low load*
    - Minimize impact of reconfiguration on application
Approach overview

1. **Intro**
2. **Approach**
3. **Forecast**

---

### Approach overview

- **Observed arrival rate** ($\lambda_t$)
- **Forecast**
  - $\lambda_{t+1} = h(\lambda_{t+1})$
- **Resource Predictor**
- **Required allocation** ($a_{t+1}$)

- **New VM memory size** ($s_{t+1}$)
- **Limit** ($l_{t+1}$)

---

**Diagram:**

- **vApp**
  - VM\(_1\)
  - VM\(_2\)
  - VM\(_n\)
- **App Sensor**
- **Workload Forecaster**
- **Sizing Controller**

---

**Equation:**

$$a_{t+1} = h(\lambda_{t+1})$$
Workload forecaster: splitTs

- Based on time series analysis
- Issues with state-of-the-art techniques
  - Multiple overlapping seasonal patterns

splitTs addresses this issue

Forecast with tBats

Thu Fri Sat Sun Mon Tue
Workload forecaster: splitTs

Observed time series

Calendar information → Classifier

Type 1 days (e.g., working days)

Type 2 days (e.g., non-working days)

Select time series

WCF forecaster (Herbst et al. 2014)
Sizing controller

Input: required memory $a_{t+1}$

Steps:
1) Stop application (optional)
2) Reconfigure VM memory settings
   a. Hot-add memory $s_{t+1} = a_{t+1}$, if $a_{t+1} > s_t$
   b. Set memory limit $l_{t+1} = a_{t+1}$, if $a_{t+1} < s_t$
3) Activate memory in OS
4) Update memory settings of application in VM
5) Start application (optional)
Forecast accuracy

- Real-world workload traces
  - FIFA’98 World Cup
  - Wikipedia
  - CICS transactions

- Evaluated forecasters
  - SplitTs
  - WCF (Herbst et al. 2014)
  - ARIMA (Box et al. 2008)
  - tBATS (Livera et al. 2011)
Forecast accuracy metric

Mean absolute scaled error (MASE) (see Hyndman et al. 2006)

\[ e_t = \text{forecastValue}_t - \text{observedValue}_t \]

\[ b_n = \frac{1}{n} \times \sum_{i=1}^{n} |\text{observedValue}_i - \text{observedValue}_{i-1}| \]

\[ \text{mase}(0; n) = \text{mean}_{t=1}^{n} \left( \frac{|e_t|}{b_n} \right) \]
Forecast accuracy

MASE (lower values are better):

<table>
<thead>
<tr>
<th></th>
<th>splitTs</th>
<th>WCF</th>
<th>ARIMA</th>
<th>tBATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFA’98</td>
<td>1.42</td>
<td>2.18</td>
<td>2.65</td>
<td>2.33</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>1.14</td>
<td>1.28</td>
<td>1.68</td>
<td>2.21</td>
</tr>
<tr>
<td>CICS</td>
<td>1.23</td>
<td>3.01</td>
<td>4.97</td>
<td>3.28</td>
</tr>
</tbody>
</table>

→ Improvement between 11% and 59% compared to WCF
Case study

- Zimbra collaboration server
- Not dynamically clusterable
- Architecture:

![Diagram showing the architecture of Zimbra collaboration server](image)
Observed response times

Proactive controller (uses workload forecasts)

- Reduced availability: → 4 min unavailable

Reactive controller (threshold based on monitored arrivals)

- Reduced availability: → 33 min unavailable

Graphs show:
- Arrival rate (req. per min)
- Response time (ms)

Legend:
- Orange: Arrival rate
- Blue: Response time
- Gray: Reduced availability
Related work

Proactive vertical CPU scaling of VMs
(Lorido-Botrán et al. 2012)
(Jennings and Stadler 2014)
(Galante and Bona 2012)

Dynamic adaptation of VM memory limits
(Shen et al. 2011)
(Lu et al. 2014)

Application-level memory management
(Salomie et al. 2013)
(Bobroff et al. 2014)
(Hines et al. 2011)
Conclusions

- Proactive approach for vertical memory scaling of virtualized applications
- SplitTs workload forecast method leverages calendar information
  - Improves forecast accuracy by up to 59% on considered traces
- Automatic control based on memory hot-add capabilities of hypervisor
  - Schedules reconfiguration during phases of low load
  - Includes adaptation of application configuration
  - Reduces unavailability time by more than 80% compared to reactive control
Future Work

- Extension to workload forecaster
  - Automatic break detection
  - Inclusion of additional user-provided meta-knowledge
- Dynamic memory demand estimation
S. Spinner  Intro          Approach       Forecast Accuracy             Case Study
References


References

