Cost and Carbon Reduction for Microsoft Azure Virtual Machines Using Workload Analysis

Daisy Wong, Oliver Zhang, Jacky Huang
Introduction

● Cloud computing is a hot market
  ○ The VM market is expected to reach US $26,042.8 million by 2033.
  ○ Thus users and organizations are relying on cloud based infrastructure.

● Enterprises wasting over 30% of their cloud spending, with wasted spending totaling $14.1 billion annually
  ○ reducing cloud expenditure has become a top priority for organizations using the cloud

● A solution is needed to investigate cost reduction and the implicated carbon emissions.
Related Work

Cortez et al. - proposed a workload prediction model to enhance resource allocation

Hadary et al. - developed a system employing ML algorithms to automate VM allocation

BUT! These studies did not address cost reduction from the user's standpoint.

Giacobbe et al. - explored migration algorithms to reduce carbon reduction

Khosravi et al. - investigated distributed data centers with varying carbon footprints and power usage effectiveness

Techniques may be difficult for users to understand and mitigate their carbon emissions.

Our study:
✓ presents a comprehensive and practical solution tailored to user needs
✓ program automatically leverages factors to recommend the most cost-effective VM for a given workload.
✓ achieves substantial cost reductions without compromising performance or quality
Solution

- Our goal: help reduce cost
  - Targeted towards individual inexperienced users who generate the most waste
- Downgrading: reducing its core size to the smallest possible size
- Two approaches
  - Passive - downgrade based on core and highest memory
  - Aggressive - downgrade based on core and lowest memory
- Thresholds: only downgrade VMs with utilizations 0 - 33% (aggressive) or 0 - 50% (passive) and waste

A table of VM core/memory combinations

<table>
<thead>
<tr>
<th>Core</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>24</td>
<td>64</td>
</tr>
<tr>
<td>30</td>
<td>70</td>
</tr>
</tbody>
</table>
Solution

- Marginal Operating Emissions Rate (MOER): emission rate of electricity generators on a certain grid at a certain time
  - Low MOER reflects more sustainable electricity and vice versa.
  - Due to electricity demand trends, MOER is often high during the day but low in the night
- Forecasting MOER allows workloads to be delayed until times of low MOER to reduce carbon emissions*
  - Workloads rescheduled at the interval with the lowest MOER level averages over the next 24 hours
- ran “Delay-insensitive” and “Unknown” type workloads as they could tolerate delays.

*operates under the assumption that the MOER levels observed today will persist unchanged into tomorrow.
Algorithm 1 Aggressive Downgrading

Require: \( \text{percentWasted} \geq 10 \)

\[
\begin{align*}
\text{while} \quad \text{canDowngrade} \quad \text{and} \quad \text{core} \geq 2 \\
\text{if} \quad \text{core} = 2 \\
\phantom{\text{while}} \quad \text{newMaxUtil} = \frac{2}{2} \times \text{maxUtil} \\
\phantom{\text{while}} \quad \text{newUtil} = \frac{2}{2} \times \text{util} \\
\phantom{\text{while}} \quad \text{core} = 2 \\
\phantom{\text{while}} \quad \text{mem} = 2 \\
\text{else} \\
\phantom{\text{while}} \quad \text{newMaxUtil} = \frac{\text{oldCore}}{\text{newCore}} \times \text{maxUtil} \\
\phantom{\text{while}} \quad \text{newUtil} = \frac{\text{oldCore}}{\text{newCore}} \times \text{util} \\
\phantom{\text{while}} \quad \text{if} \quad \text{newMaxUtil} < 100 \\
\phantom{\text{while}} \\
\phantom{\text{while}} \quad \text{maxUtil} = \text{newMaxUtil} \\
\phantom{\text{while}} \quad \text{util} = \text{newUtil} \\
\phantom{\text{while}} \quad \text{else} \\
\phantom{\text{while}} \\
\phantom{\text{while}} \quad \text{canDowngrade} = \text{false} \\
\text{end if} \\
\text{end if} \\
\text{if} \quad \text{newMaxUtil} > 100 \text{ or } \text{newUtil} > 100 \\
\phantom{\text{while}} \quad \text{core} = \text{coreLevel} \\
\phantom{\text{while}} \quad \text{mem} = \text{memLevel} \\
\text{end if} \\
\text{end while}
\]

Algorithm 2 Passive Downgrading

Require: 

\[
\begin{align*}
\text{percentWasted} \geq 10 \\
\text{core} \geq 2 \text{ and } \text{mem} \geq 8 \\
\text{while} \quad \text{canDowngrade} \quad \text{and} \quad \text{core} \geq 2 \\
\text{if} \quad \text{core} = 2 \\
\phantom{\text{while}} \quad \text{newMaxUtil} = \frac{2}{2} \times \text{maxUtil} \\
\phantom{\text{while}} \quad \text{newUtil} = \frac{2}{2} \times \text{util} \\
\phantom{\text{while}} \quad \text{core} = 2 \\
\phantom{\text{while}} \quad \text{mem} = 8 \\
\text{else} \\
\phantom{\text{while}} \quad \text{newMaxUtil} = \frac{\text{oldCore}}{\text{newCore}} \times \text{maxUtil} \\
\phantom{\text{while}} \quad \text{newUtil} = \frac{\text{oldCore}}{\text{newCore}} \times \text{util} \\
\phantom{\text{while}} \quad \text{if} \quad \text{newMaxUtil} < 100 \\
\phantom{\text{while}} \\
\phantom{\text{while}} \quad \text{maxUtil} = \text{newMaxUtil} \\
\phantom{\text{while}} \quad \text{util} = \text{newUtil} \\
\phantom{\text{while}} \quad \text{else} \\
\phantom{\text{while}} \\
\phantom{\text{while}} \quad \text{canDowngrade} = \text{false} \\
\text{end if} \\
\text{end if} \\
\text{if} \quad \text{newMaxUtil} > 100 \text{ or } \text{newUtil} > 100 \\
\phantom{\text{while}} \quad \text{core} = \text{coreLevel} \\
\phantom{\text{while}} \quad \text{mem} = \text{memLevel} \\
\text{end if} \\
\text{end if}
\]

6
Algorithm 3 Carbon Reduction

Require: $\text{RunTime} > 0$

$\text{AverageMOER} = \text{Average MOER in Window}$

$\text{Window} \leftarrow \text{Start Time to End Time}$

for \( \left( \frac{1\text{Day}}{5\text{minutes}} \right) \) do

\hspace{1em} \text{Shift Window By 5 Minutes}

\hspace{1em} \text{NewAverage} \leftarrow \text{Average of Modified Window}

\hspace{1em} \text{if} \text{ NewAverageMOER} < \text{AverageMOER} \text{ then}

\hspace{2em} \text{AverageMOER} \leftarrow \text{NewAverageMOER}

\hspace{2em} \text{BestWindow} \leftarrow \text{Window}

\hspace{1em} \text{end if}

\hspace{1em} \text{end for}

\hspace{1em} \text{Reschedule Task to BestWindow}$

$\text{Savings} = \text{DefaultEmission} - \text{AverageMOER}$

$\text{PercentSavings} = 100 \times \left( \frac{\text{Savings}}{\text{DefaultEmission}} \right)$
Results

- Using data from a 2019 Azure Trace:
  - Total cost of over 2.6 million VMs: $23 million before downgrading
  - 1,975,282 VMs aggressively downgraded saved almost $4 million or 17% in savings
  - 730,436 VMs passively downgraded saved users a total of about $950,000 or 4% in savings

- Carbon savings ranged from 0 to 8% in the Delay Insensitive group
- Average carbon savings was 55% in the Unknown group of workloads
Conclusion - Major Contributions

Real-World Work Analysis
Analyze the characteristics of approximately 2.7 million VMs in the Azure Public Dataset
Observe consistent patterns where as VM requests more resources, average utilization decreases

VM Cost Prediction Model
Utilize linear regression algorithm to find the cost of each VM using specific characteristics

Waste Quantification
Propose a metric to quantify the waste of VM by considering the total cost and resource utilization
Offers insights into users of the cloud
Cost Reduction Algorithm
Effective solution for reducing the cost of cloud computing
Downgrades VMs by reducing core size

Carbon Reduction Algorithm
Innovative solution to maximize cloud computing efficiency and minimize carbon emissions
Limitations

- Limited by the large number of VMs, VM availability and memory space
- Limited by the challenge of accurately predicting MOER
- Does not reflect real world scenarios as it may not be possible to change VM size and run time whenever needed.

Future Works

- Explore the relationship between cost saving and carbon emissions
- Evaluate a new algorithm incorporating external factors for precision in forecasting MOER levels.
Thank You!