A Deep Learning-based approach to VM behavior identification in cloud systems

Matteo Stefanini, Riccardo Lancellotti, Lorenzo Baraldi, Simone Calderara

University of Modena and Reggio Emilia
Department of engineering “Enzo Ferrari”
Cloud Computing Challenges

• Critical operations in Cloud data centers
  - Monitoring (overloaded/underutilized VMs and Hosts)
  - Management (huge bin packing problem)

• Main challenge: Scalability
  - Volume of data for monitoring
  - Size (and dimensionality) of optimization problem

• Current solution
  - Oversimplification of the problem
Identification of VMs

• Alternative approach:
  - Exploit similarity in VMs: (classes, not instances)
  - Reduced problem size (less data, less VMs)

• Problem: how to classify VMs?
  - Fast and accurate classification
State of the art

- Trade-off accuracy/speed
  - Fast classification is not accurate
  - Accurate classification takes time
  - Cannot be applied to on-demand VMs in public Cloud

- Adaptive Gray Area TErchnique (AGATE)
  - Add a confidence value to classification
  - Fast and accurate classification of some VMs
  - Still unsatisfactory → Proposal of a different approach
Deep Learning model

- Input: time series of W samples of several VMs metrics
- Output: class belonging probabilities
- Multiple layers (number depending on the input size)
- Two models:
  - DeepConv: based on convolutive networks
    Focus on patterns between samples
  - DeepFFT: based on Fast Fourier Transformation
    Focus on spectral domain (novel Deep Learning approach)
Deep Learning model

- General structure:
  - Input layer (pre-processing of samples)
  - Processing blocks (multiple layers)
  - Fully connected layer (and softmax classifier)

- DeepConv:
  - Standard model

- DeepFFT:
  - Performs FFT in input layer
Model representation

---

Input metrics (channels)

Block 1 → Block 2 → Block 3 → Block 4 → Fully Connected layer (data flattened) → Softmax → Class probabilities

Time OR Frequency

---

CLOSER 2019, May., 2-4, Heraklion, Greece
Processing block

- Each processing block contains
  - Activation function (ReLU)
  - Batch Normalization
  - 1-Dimensional convolution

- Each block:
  - Reduces by 2 the input size (stride=2)
  - Doubles the number of channels

- Number of blocks: \( N_b = \max(\log_2(W) - 1, 2) \)
Implementation details

- Implementation based on Pytorch
  - In-house implementation of FFT

- Source code available
  - Code in git repository
  - See paper for details

- Deployment on CINECA data center
Experimental setup

- Data from a real datacenter (e-health app)
- Two classes of VMs:
  - Web servers
  - DBMS
- Traces divided in chunks with different window
  - 1 sample every 5 min
  - 4 samples (20 mins) → 256 samples (21 hrs)
Experimental setup

- **16 metrics** (virtualized HW / guest OS)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SysCallRate</td>
<td>Rate of system calls [req/sec]</td>
</tr>
<tr>
<td>CPU</td>
<td>CPU utilization [%]</td>
</tr>
<tr>
<td>IdleCPU</td>
<td>Idle CPU fraction [%]</td>
</tr>
<tr>
<td>I/O buffer</td>
<td>Utilization of I/O buffer [%]</td>
</tr>
<tr>
<td>DiskAvl</td>
<td>Available disk space [%]</td>
</tr>
<tr>
<td>CacheMiss</td>
<td>Cache miss [%]</td>
</tr>
<tr>
<td>Memory</td>
<td>Physical memory utilization [%]</td>
</tr>
<tr>
<td>UserMem</td>
<td>User-space memory utilization [%]</td>
</tr>
<tr>
<td>PgOutRate</td>
<td>Rate of memory pages swap-out [pages/sec]</td>
</tr>
<tr>
<td>InPktRate</td>
<td>Rate of network incoming packets [pkts/sec]</td>
</tr>
<tr>
<td>OutPktRate</td>
<td>Rate of network outgoing packets [pkts/sec]</td>
</tr>
<tr>
<td>InByteRate</td>
<td>Rate of network incoming traffic [KB/sec]</td>
</tr>
<tr>
<td>OutByteRate</td>
<td>Rate of network outgoing traffic [KB/sec]</td>
</tr>
<tr>
<td>AliveProc</td>
<td>Number of processes in system</td>
</tr>
<tr>
<td>ActiveProc</td>
<td>Number of active processes in run queue</td>
</tr>
<tr>
<td>RunTime</td>
<td>Execution time</td>
</tr>
</tbody>
</table>
Deep Learning performance

Models Accuracy with respect to window length

- **DeepConv**
- **DeepFFT**

Accuracy

Window (Time-steps)

CLOSER 2019, May., 2-4, Heraklion, Greece
Comparison with AGATE

% Error Our models Vs. previous state-of-the-art

- DeepConv error (ours)
- DeepFFT error (ours)
- PCA-based error
- Agate error
- Agate upper gray area

VM Misplaced (%) vs. Time (hours)
Concluding remarks

• Challenge: **scalability** of monitoring/management in Cloud data centers → VMs identification

• Complex to achieve **fast and accurate** identification

• Proposal of a **Deep Learning**-based approach

• **Outperforms** state of the art (AGATE)

• Suitable also for **on-demand** VMs
Future research directions

- Thorough evaluation in cases with limited / low-quality data
- Identification of new classes:
  - Auto-encoders / triggers in NN
  - Integration with AGATE
- Generative Adversarial Network for workload generation
A Deep Learning-based approach to VM behavior identification in cloud systems

Matteo Stefanini, Riccardo Lancellotti, Lorenzo Baraldi, Simone Calderara

University of Modena and Reggio Emilia
Department of engineering “Enzo Ferrari”