Empirical Evaluation of Workload Forecasting Techniques for Predictive Cloud Resource Scaling

In Kee Kim, Wei Wang, Yanjun (Jane) Qi, and Marty Humphrey
Computer Science @ University of Virginia
Motivation - Cloud Resource Scaling Approach

Reactive Auto Scaling
[AWS, Google, Azure, etc.]

Autoscaling based on Resource Utilization:
CPU, Memory, Network-I/O...
Motivation - Cloud Resource Scaling Approach

Reactive Auto Scaling
[AWS, Google, Azure, etc.]

Autoscaling based on Resource Utilization:
CPU, Memory, Network-I/O...

![Resource Demand vs. Time Graph](image-url)
Motivation - Cloud Resource Scaling Approach

Reactive Auto Scaling
[AWS, Google, Azure, etc.]

Autoscaling based on Resource Utilization:
CPU, Memory, Network-I/O...

![Graph showing resource demand and number of instances over time](image-url)

- Time
- Resource Demand
- Number of Instances
Motivation - Cloud Resource Scaling Approach

Reactive Auto Scaling
[AWS, Google, Azure, etc.]

Autoscaling based on Resource Utilization:
CPU, Memory, Network-I/O...

Diagram showing resource demand over time with scaling delays and scaling actions.
Motivation - Cloud Resource Scaling Approach

**Reactive Auto Scaling**

[AWS, Google, Azure, etc.]

Autoscaling based on Resource Utilization:
- CPU, Memory, Network-I/O...

**Predictive Resource Scaling**

Resource Scaling based on forecasting:
1. Future Resource Usage
2. Workload Arrival Pattern

---

Diagram showing scaling delays over time and resource demand vs. number of instances.
Motivation - Cloud Resource Scaling Approach

Reactive Auto Scaling
[AWS, Google, Azure, etc.]
Autoscaling based on Resource Utilization:
CPU, Memory, Network-I/O...

Predictive Resource Scaling
Resource Scaling based on forecasting:
1. Future Resource Usage
2. Workload Arrival Pattern
Predictive Resource Scaling

1. Workload Predictor
   - Detects cloud workload pattern.
   - Predicts job arrival pattern in near future.

2. Resource Scaling
   - allocates/deallocates cloud resources based on the prediction.
Predictive Resource Management Engine

Workload Predictor ➔ Resource Scaling

Workload

Regression?

Time Series?

Machine Learning?

Cloud Infrastructure

Predictive Resource Scaling
Research Questions

• **Question #1**: Which workload predictor has the highest accuracy for job arrival time prediction?

• **Question #2**: Which exiting workload predictor has the best cost efficiency and performance benefits?

• **Question #3**: Which styles of predictive scaling achieves the best cost efficiency and performance benefits?
Research Big Picture

• **4K** cases are very challenging via actual deployment on IaaS clouds.
  - Use **PICS** (Public IaaS Cloud Simulator) - KWH - CLOUD’15
Experiment Design

- Collection of Workload Predictors.
- Simulation Workloads.
- Design of Resource Management System.
- Implementation and Performance Tuning.
Collection of (Existing) Workload Predictors

• We collect all 21 workload predictors:

1) Naïve Models
   - Mean-based
   - Recent-mean (kNN)

2) Regression Models
   - Global Model (Linear, Quad, Cubic)
   - Local Model (Linear, Quad, Cubic)

3) Time Series Models
   - Smoothing (WMA, EMA, DES)
   - Box-Jenkins (AR, ARMA, ARIMA)

4) Non-Temporal (ML) Models
   - SVMs (Linear, Gaussian)
   - Decision Tree
   - Ensemble (RF, GBM, Exts)
Simulation Workload Patterns

• We generate 24 workload patterns based on:

- On and Off (Batch/Scientific)
- Random/Unpredictable (Media)
- Growing (Emerging Service)
- Cyclic Bursting (E-Commerce)
Design of Resource Management System

Cloud Resource Management System

Workload

Job Portal

Job Portal

Job

Job

Resource Management Module
(e.g. job scheduling, VM scaling, and management)

Predictive Scaler

Predictive Scaling Module

Cloud Infrastructure
(e.g. AWS, Azure)

Workload Repository

Predictor for Scaling-Out

Predictor for Scaling-In

Samples for Prediction

Prediction Result

Job (Duration, Deadline)

Job Arrival Info

 +/- VMs, Job Assign.

Samples for Prediction

Prediction Result

Job Exe

JobQueue

Job Exe

Job Exe
Implementations and Performance Tuning

• Workload Predictor Implementation.
  - All predictors are written in Python.
    - numpy and Pandas.
    - statsmodels for time-series model implementation.
    - scikit-learn machine learning lib for non temporal models.

• Predictor Performance Tuning.
  - (Training) Sample Size Decision:
    - a tradeoff between prediction performance and overhead.
    - Most predictors use 50 -- 100 of most recent job arrival samples.
  - Parameter Selection:
    - a grid search algorithm with prediction accuracy.
Performance Evaluation

- Experiment #1 - Statistical Predictor Performance.
- Experiment #2 - Predictive Scaling Performance.
Experiment #1 - (Statistical) Predictor Performance

• Purpose: Measuring **Statistical Predictor Accuracy** and **Overhead**.
  - Accuracy: MAPE - Mean Absolute Percentage Error.
  - Overhead: Sum of All Prediction Time.

• Overall Results:

![Overall Prediction Accuracy](image1)

![Overall Prediction Overhead (10K Jobs)](image2)
Experiment #1 - (Statistical) Predictor Performance

• Purpose: Measuring **Statistical Predictor Accuracy** and **Overhead**.
  - Accuracy: MAPE - Mean Absolute Percentage Error.
  - Overhead: Sum of All Prediction Time.

• Overall Results:

![Overall Prediction Accuracy](image1)

- **SVMs**: 0.37 -- 0.4 (42% less than average)

- **Average**: 0.6360

![Overall Prediction Overhead (10K Jobs)](image2)
Experiment #1 - (Statistical) Predictor Performance

- **Purpose:** Measuring **Statistical Predictor Accuracy** and **Overhead**.
  - Accuracy: **MAPE** - Mean Absolute Percentage Error.
  - Overhead: Sum of All Prediction Time.

- **Overall Results:**

![Overall Prediction Accuracy](image1)

![Overall Prediction Overhead (10K Jobs)](image2)

- **kNN:** 0.5s for 10K Jobs
- **ARMA:** 6032s
## Experiment #1 - (Statistical) Predictor Performance

- Accuracy of Workload Predictor per Pattern.

<table>
<thead>
<tr>
<th>Workload</th>
<th>Rank</th>
<th>Predictor</th>
<th>MAPE</th>
<th>Workload</th>
<th>Rank</th>
<th>Predictor</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Growing</td>
<td>1</td>
<td>Lin. SVM</td>
<td>0.28</td>
<td>On/Off</td>
<td>1</td>
<td>Gau. SVM</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>AR</td>
<td>0.29</td>
<td></td>
<td>2</td>
<td>ARMA</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>ARMA</td>
<td>0.30</td>
<td></td>
<td>3</td>
<td>Lin. SVM</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
<td>--</td>
<td>0.51</td>
<td></td>
<td>Avg.</td>
<td>--</td>
<td>0.69</td>
</tr>
<tr>
<td>Bursty</td>
<td>1</td>
<td>ARIMA</td>
<td>0.38</td>
<td>Random</td>
<td>1</td>
<td>Gau. SVM</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Brown’s DES</td>
<td>0.41</td>
<td></td>
<td>2</td>
<td>Lin. Reg.</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Lin. SVM</td>
<td>0.43</td>
<td></td>
<td>3</td>
<td>Lin. SVM</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
<td>--</td>
<td>0.75</td>
<td></td>
<td>Avg.</td>
<td>--</td>
<td>0.52</td>
</tr>
</tbody>
</table>
Experiment #2 - Predictive Scaling Performance

• Purpose: **How much benefits** RM can achieve by applying
  1. “Good Predictor”
  2. Different Styles of Predictive Scaling.

• List of Predictors: **8 best predictors** from evaluation #1.
  • Linear Regression, WMA, BRDES, AR, ARMA, ARIMA, Linear SVM, Gaussian SVM

• Four Different Styles of Resource Scaling.
  - **RR** (Reactive Scaling-Out + Reactive Scaling-In) -- **Baseline**
  - **PR** (Predictive Scaling-Out + Reactive Scaling-In)
  - **RP** (Reactive Scaling-Out + Predictive Scaling-In)
  - **PP** (Predictive Scaling-Out + Predictive Scaling-In)

• Cloud Configurations: Two Pricing Models -- Hourly and Minutely.
• Metrics: Cloud Cost and Job Deadline Miss Rate.
Experiment #2 - Predictive Scaling Performance

• Overall Results:

(a) Hourly Pricing Model
(b) Minutely Pricing Model
Experiment #2 - Predictive Scaling Performance

• Overall Results:

(a) Hourly Pricing Model

(b) Minutely Pricing Model
Experiment #2 - Predictive Scaling Performance

• Overall Results:

(a) Hourly Pricing Model

(b) Minutely Pricing Model
Experiment #2 - Predictive Scaling Performance

- PP (Predictive Scaling-Out - Predictive Scaling-In) Details -- Deadline Miss Rate

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Growing</td>
<td>Linear SVM</td>
<td>AR</td>
<td>ARMA</td>
</tr>
<tr>
<td>On/Off</td>
<td>Gaussian SVM</td>
<td>ARMA</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Bursty</td>
<td>ARIMA</td>
<td>Brown’s DES</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Random</td>
<td>Gaussian SVM</td>
<td>Linear Regression</td>
<td>Linear SVM</td>
</tr>
</tbody>
</table>
Experiment #2 - Predictive Scaling Performance

- **PP (Predictive Scaling-Out - Predictive Scaling-In) Details -- Deadline Miss Rate**

### Hourly Pricing Model

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Growing</td>
<td>Linear SVM</td>
<td>AR</td>
<td>ARMA</td>
</tr>
<tr>
<td>On/Off</td>
<td>Gaussian SVM</td>
<td>ARMA</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Bursty</td>
<td>ARIMA</td>
<td>Brown’s DES</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Random</td>
<td>Gaussian SVM</td>
<td>Linear Regression</td>
<td>Linear SVM</td>
</tr>
</tbody>
</table>

### Minutely Pricing Model
Experiment #2 - Predictive Scaling Performance

- **PP (Predictive Scaling-Out - Predictive Scaling-In) Details -- Deadline Miss Rate**

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Growing</td>
<td>Linear SVM</td>
<td>AR</td>
<td>ARMA</td>
</tr>
<tr>
<td>On/Off</td>
<td>Gaussian SVM</td>
<td>ARMA</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Bursty</td>
<td>ARIMA</td>
<td>Brown’s DES</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Random</td>
<td>Gaussian SVM</td>
<td>Linear Regression</td>
<td>Linear SVM</td>
</tr>
</tbody>
</table>
Experiment #2 - Predictive Scaling Performance

- **PP** (Predictive Scaling-Out - Predictive Scaling-In) Details -- Deadline Miss Rate

### Workloads

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Growing</td>
<td>Linear SVM</td>
<td>AR</td>
<td>ARMA</td>
</tr>
<tr>
<td>On/Off</td>
<td>Gaussian SVM</td>
<td>ARMA</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Bursty</td>
<td>ARIMA</td>
<td>Brown’s DES</td>
<td>Linear SVM</td>
</tr>
<tr>
<td>Random</td>
<td>Gaussian SVM</td>
<td>Linear Regression</td>
<td>Linear SVM</td>
</tr>
</tbody>
</table>

### Hourly Pricing Model

- **Lin-SVM (Best)**
- **Gau-SVM (Best)**
- **BRDES (Best)**
- **Lin-SVM (Best)**

### Minutely Pricing Model

- **AR (Best)**
- **ARMA (Best)**
- **BRDES (Best)**
- **Gau-SVM (Best)**
Summary -- Revisit 3 Research Questions

• Q1: Which WL predictor has the highest accuracy?
  - No one predictor fits all workload patterns.

• Q2: Which WL predictor provides the best performance benefits?
  - Similar with Q1 – no universally best workload predictor exits.
    - Depends on workload patterns and cloud configurations (e.g. billing model)
    - In general, best workload predictor (cloud metric) is one of top three most (statistically) accurate predictors.

• Q3: Which styles of predictive scaling provides the best performance benefits?
  - PP (Predictive Scaling-Out/In) is the best style of predictive scaling.
    - “Predictive Scaling-Out” can improve cloud metrics.
Questions?

Thank you!

ik2sb@virgina.edu