Towards an Autonomic Auto-Scaling Prediction System for Cloud Resource Provisioning

Ali Nikravesh; Samuel A. Ajila; Chung-Horng Lung
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
Agenda

• Introduction
  • Problem Statement
  • Research Scope
  • Research Goal

• Our Approach
  • Workload Patterns
  • Time-series Prediction Algorithms

• Experiment and Results
• Self-adaptive Prediction Suite
• Conclusions & Future Work
• Two important characteristics of cloud computing:
  • Elasticity: Users can acquire and release resources on demand
  • Pay as you go pricing model
• Elasticity can lead to cost/performance trade-off
  • Over-provisioning
    • Cost
  • Under-provisioning
    • SLA breach
• Cost/performance trade-off
  • Solution → Auto-scaling systems: automatically adjusts resources based on the incoming requests
• **Auto-Scaling**
  • **Reactive**
    • Advantages: simple, easy to use
    • Disadvantage: slow, neglects virtual machine (VM) boot-up time. (between 5 to 15 minutes!)
  • **Proactive**
    • Advantage: considers overhead in advance, VM boot-up time
    • Disadvantage: suitable for environments with predictable load characteristics
  • **Predictive**
    • Advantage: can predict unplanned load spikes
    • Disadvantage: accuracy is a challenge
• Auto-Scaling
  • Reactive
    • Advantages: simple, easy to use
    • Disadvantage: neglects virtual machine (VM) boot-up time. (between 5 to 15 minutes!)
  • Proactive
    • Advantage: considers VM boot-up time
    • Disadvantage: suitable for environments with predictable load characteristics
  • Predictive
    • Advantage: can predict unplanned load spikes
    • Disadvantage: poor accuracy
• Cloud computing layers
  • Infrastructure as a Service (IaaS)
  • Platform as a Service (PaaS)
  • Software as a Service (SaaS)

• Cloud types
  • Public Cloud: accessible to public
  • Private Cloud: restricted for private use
  • Hybrid Cloud: combination of both public and private
• Cloud computing layers
  • Infrastructure as a Service (IaaS)
  • Platform as a Service (PaaS)
  • Software as a Service (SaaS)

• Cloud types
  • Public Cloud: public accessible
  • Private Cloud: restricted for private use
  • Hybrid Cloud: combination of both public and private
• Predictive Auto-Scaling system architectural overview

• Research goal: improve Predictor’s accuracy

• Hypothesis:

  Prediction **accuracy** of predictive auto-scaling systems can be increased by choosing an appropriate **time-series prediction algorithm** based on the incoming **workload pattern**
Objective:
Investigate the impact of different workload patterns on the prediction accuracy of time-series prediction algorithms

• Steps:
  1. Investigate workload patterns
  2. Explore time-series prediction algorithms
  3. Conduct experiments to compare prediction algorithms and validate the hypothesis
Workload refers to a number of user requests, together with the arrival times (trend)

Workload patterns in cloud computing IaaS environment:
- **Growing** pattern: represents workloads with increasing trend
- **Periodic** pattern: represents workloads with seasonal changes.
- **Unpredicted** pattern: represents fluctuating workloads.
• Time-series algorithms used in auto-scaling environments:
  
  • **Moving Average**
    • Poor prediction results
    • Usually used only for noise-removal purposes
  
  • **Auto-Regression**
    • Largely used for prediction purposes in auto-scaling
    • Performance highly depends on the monitoring interval, size of the history window, and size of the adaptation window
  
  • **ARMA** (autoregressive–moving-average)
    • Combination of “Moving Average” and “Auto-Regression”
  
  • **Machine Learning** The best prediction approach
• Support Vector Machine (SVM) and Neural Networks (NN) are the most accurate machine learning algorithms in the cloud auto-scaling field.

• **Support Vector Regression** (SVR) is the methodology by which a function is estimated using observed data, which in turn “trains” the SVM.

• **Neural Network** is a two-stage regression or classification model, typically represented by a network diagram.
Hypothesis

Prediction accuracy of predictive auto-scaling systems can be increased by choosing an appropriate time-series prediction algorithm based on the incoming workload pattern.

Objective: To explore relations between different workload patterns and prediction accuracy of SVM and NN.
• **Benchmark:** TPC-W benchmark (3 tier online bookstore website)

• **Infrastructure:** Amazon EC2
1. Select and generate a pattern using TPC-W workload generator (experiment duration is **300 minutes**)

2. On the webservice machine, count total number of user requests per minute and store results in a trace file

3. Divide the time-series into “training” and “testing” datasets:
   - **Training (60%)**: train SVM and NN using the “training” dataset (using “sliding window” and “cross-validation” techniques to create prediction models)
   - **Testing (40%)**: Generate workload predictions using SVM and NN prediction models

4. Compare SVM and NN prediction results using
   - **RMSE** (Root Mean Square Error)
   - **MAPE** (Mean Absolute Percentage Error)
TPC-W workload generator used along with customized scripts to produce “growing”, “periodic”, and “unpredicted” patterns.
Experiment Results

Periodic

Growing

Unpredicted
• Periodic pattern:
  • SVM outperforms NN
  • Increasing window size increases error for NN (upward trend), but does not affect SVM prediction accuracy
• Growing pattern:
  • SVM outperforms NN
  • Increasing window size increases error for NN (up & down), but does not affect SVM prediction accuracy
• Unpredicted pattern:
  • NN outperforms SVM
  • Increasing window size increases prediction accuracy for both
• Lesson:
  • Prediction accuracy can be improved by using a self-adaptive prediction suite that chooses the most suitable prediction algorithm based on the incoming workload pattern
Design of the Self-Adaptive Prediction Suite

- **Pattern-1**: CInterface()
- **Pattern-2**: C-Interface()
- **Pattern-3**: C-Interface()

**WorkLoadPattern**: ContextInterface()

**Concrete-A-1**: A-Interface()
**Concrete-A-2**: A-Interface()
**Concrete-A-n**: A-Interface()

**CloudWorkLoad**

**Predictor**

**Decision Maker**

**Monitor**

**Algorithms**: AlgorithmInterface()
We investigated machine learning techniques for auto-scaling prediction.

- Experimental results:
  - For “growing” or “periodic” workload patterns SVM outperforms NN
  - For “unpredicted” workload patterns NN outperforms SVM
  - Increasing the sliding window size
    - Positive impact on SVM and NN for “unpredicted” workload pattern
  - Ineffective to use only one particular prediction technique for all environments

Proposed self-adaptive prediction suite – multi-tier adaptation “strategy” and “template” design patterns
Future Work

- Detailed design of **self-adaptive prediction suite**
  - Multi-tier adaptation
  - Performance knowledge base – inference

- Investigate the impact of increasing **prediction accuracy** on the final **scaling decision**

- Study the impact of the **database layer** and latency on
  - Multi-tier adaptation
  - Prediction and decision making accuracy
  - Workload patterns & window sizes
  - Pricing models and SLAs
Thanks

Questions?