Topology Aware Analytics for Elastic Cloud Services

Foudoulis Athanasios
athafoud@cs.ucy.ac.cy

Master Thesis Presentation
May 28th 2015, Department of Computer Science, University of Cyprus
In Brief..

a Tool providing Performance and Cost Analytics for Elastic Cloud Services using Monitoring and Cost Data and utilizing the Cloud Service Topology
In Brief..

Load Balancer

Application Server
Apache Tomcat

Database Server
NoSQL
Cassandra DB

Graphs showing cost and standdev over time, and write latency over time.
Presentation Outline

• Cloud Computing and Cloud Analytics
• The gap in Cloud Analytics Tools
• Our Approach
  • Design / Implementation
  • Evaluation
• Conclusions & Future Work
Cloud Computing

- Provides a shared pool of physical resources
  - data storage space, networks and computer processing power
- Utilized to run virtualized resources
Cloud Computing Economics

• Biggest advantages of cloud computing
  • Rent on demand resources
    • reduce bare metal machines maintenance costs
  • Pay as you go
    • initial cost is minimal
    • over-cost ”easily” manageable
Cloud computing is responsible for providing the facilities that host nowadays the larger percentage of computing and storage services of the Internet.

Cloud Maturity of Responders

Source: Right 2015 Scale State of the Cloud Report

Enterprise workloads by type

Source: Right 2015 Scale State of the Cloud Report
Cloud Service Development Lifecycle

- Idea
- Build
- Code
- Measure
- Data
- Learn

Cloud Usage and Cost Analytics Tools
What is Analytics?

The process of applying **mathematics** and **statistics** over a set(s) of data and the **visual representation** of the extracted information.

- **Web analytics (e.g. Google)**
- **Business Analytics (e.g. SAP)**

**Website Traffic and Visit**

**Sales of a product**
Cloud Analytics

- Insights into application behaviour
  - assist users to improve application performance, resource utilization and thereby reduce cost

- Increase of demand from cloud analytics
  - More than 1,000 customers in RightScale Cloud Analytics platform after a month announced
Analytics Insights

• Utilization / Performance
  • Usage
    • e.g. ram or disk used in GB, cpu load in percentage
  • State of a resource over time
    • Short and long term trends
  • Visualizations intercepting the data input rate (e.g. requests/sec) and the resource\ usage (CPU, memory usage)

• Cost
  • Cost per hour/month/year
  • Cost change over the time
But Still – Lack of Visibility

“87% of deployments found underutilized in terms of CPU, Memory, I/O and Network”

“Instances are underutilized with average utilization rates between 8-9%”

“Difficult for user to digest analytics data, as cloud deployments scale out to massive sizes”
Different Users – Different Needs
IT Staff – Maintainers

• Forensic analysis of an incident, primarily uses monitoring data

• Analytics for anomaly detection e.g. in request rate to identify dos attacks
Different Users – Different Needs

• Application Developers
  • Optimize / Fine-tune cloud Services
  • Configure Cloud Service Elastic Behavior (e.g. set rules / thresholds)
Different Users – Different Needs

- Department Manager
  - Expenses management for the departments Products / Services
  - Single / Overall deployment(s) cost
Develop and Deploy

![Diagram showing OpenStack instances and a load balancer.](image-url)
Develop and Deploy
Cloud Service Analytics

- Analytics Per Instance
  - CPU utilization
- Possible Aggregation
  - Per Tenant
  - Per Cloud Project
  - Per Region
Cloud Service Analytics

• Find which instance under-performs
• Identify the most costly project
• Pinpoint which tenant cost less

Helpful for IT Staff and Managers

Insufficient for Developers

• Can not easily focus on single deployment
• Can not optimize a single application tier (e.g. database)
Visualizations Based on Service Topology

Designing User-oriented or User-centric Interfaces can increase the user experience

*From Human Computer Interaction (HCI) literature
Advance over State-of-the-Art

- Focus is driven on
  - Resource performance visualization
  - Total application cost visualization

- State of the Art not aware of application topology
  - Insights limited to application level
  - Multi-grain performance evaluation is not available
    - (e.g. how scaling one tier affects the cost of another tier)
Our Proposal

• An analytics tool
  • Aim helping Cloud Service Developers
  • Provide an additional view of the existing information.

• Cloud Service Topology - The keystone of our tool
  • a way to graphically represent the blueprint of a cloud service.
Our Proposal

• Extract insights, such as the trend of the data

• Apply statistical calculations,
  • In the time-series formatted data
Leverage Service Topology

• Enhance metrics data with service topology
• Provide fine-grained analytics.
  • Per tier Analytics (useful to identify bottlenecks or possible bottlenecks)
  • How one tier can affect others
Support elastic applications

• Exploit cloud elasticity

• More detailed analytics for the cloud application lifecycle.
  • Help understand what it may seem abnormal behavior
  • Use elasticity actions as time bookmarks to navigate through the deployment
Functionality / Requirements

• Calculate analytics for a cloud service deployment
• Visualize and Map analytics on the service topology
• Integrate different data sources (e.g. usage and cost metering data)
• Understand and expose the elastic nature of a service
• Respond in a timely manner
Implementation

• Two components
  • Back-end (Computation Layer)
    • Retrieves data from the configured data-sources
    • Combines and Analyzes the data
    • Exports the enriched data via a REST API.
  • Front-end (Presentation Layer)
    • Helps the user interact with the Backend Service
    • Presents the information in a unified and descriptive manner
The Analytics Tool

• Runs as a Cloud Service
Architecture
Data Collection

• Provides
  • A set of interfaces in order to ensure the needed abstraction
  • A set of data objects to help communication of the “internal” modules

• Interacts with “external” data sources to get data
  • Monitoring System
  • Cost calculation Module
  • File System

• Transforms the data to the “internal” structures
Analytics Calculation

• Statistical analysis over time-series data
• Common methods like MIN, MAX, MEDIAN, MEAN, SD and COUNT
  • Based on apache.common.math

• Additional Optimization for performance
  • Incremental MEAN calculation
Analytics Calculation

• Trend Calculation
  • Implements a Simple Moving Average (SMA)* function
• Process-level Parallelization
  • Break data to small chunks
  • Calculate each chunk Trend
  • Combine the result

* G. Sanchez, 2010, “Moving intervals for nonlinear time series forecasting,”
Visualizing Time-series

- Time-series data
  - "a sequence of data points, typically consisting of successive measurements made over a time interval"
- Frequently plotted via line charts
  - The most popular choice of visualization, among scatter plots and space filling approaches
Visualizing Time-series - Issues

• Not the whole data can fit to a screen
  • Plotting too much data* results
    • ‘messy’ Visualizations
    • Reduces the clarity and readability

• User can digest less data than the one actually visualized

* According to screen size in pixels
Visualizing Time-series - Solutions

• Reduce the amount of data (Data reduction)
  • Remove some strain from the rendering tool
  • Transfer less data over network
• Data reduction Techniques
  • Aggregation, Binning and Sampling
• Algorithms preserve
  • The visualization accuracy
  • Not the analysis process accuracy
Down-sampling

• Largest-Triangle-Three-Buckets algorithm
  • Separates the raw data in almost equal size buckets and
  • Selects from each one the most representative point based on the previously selected sample.
Additional Optimizations

• Visual Feedback
  • While user waits for analytics Calculations
• Partial Loading / Pre-Loading
  • Quickly create and present an ‘estimation’ of the visualization
    • e.g. Using High sampling rate
  • Add the details later, seamlessly
Does it meet our expectations?

- Expected Characteristics / Requirements
  - Calculate analytics for a cloud service deployment
  - Visualize and Map analytics on the service topology
  - Integrate different data sources (e.g. usage and cost metering data)
  - Understand and expose the elastic nature of a service
  - Respond in a timely manner
Evaluation - Functionality

- Define two realistic applications
- Configure to use the collected data
- Use UI to inspect analytics
Use Cases

Case 1

A three-tier service*

1. HAProxy load balancer
2. Application server tier
3. Cassandra NoSQL distributed data storage

*D. Trihinas, “Managing and monitoring elastic cloud applications”

**Service Topology As it was Designed through CAMF User Interface
Use Cases

Case 2

A data analysis and exploratory game application*
1. Load balancer
2. Application servers (GO)
3. Compute nodes for extracting and transforming of data
4. Redis cache
5. PostgreSQL database as the primary storage
6. Cassandra DB storing large binary blobs

*Dataplay was developed by Playgen for the purposes of CELAR (FP7 - 317790)
**Service Topology As it was Designed through CAMF User Interface
Evaluation - Results

View ‘auto-generated’ analytics reports, for cloud services components including both usage and cost metrics.
Evaluation - Results

Present analytics for various components mapped onto the cloud service topology
Evaluation - Response time

- Define What to measure
- Date sets
  - Up to 1-million random metric values
  - Data from running a 3-Tier Application
- Repeat each measurement several times, extract the mean
- Test if response time < 4s

D. Trihinas, “Managing and monitoring elastic cloud applications”
F. F.-H. Nah, “A study on tolerable waiting time: how long are web users willing to wait?”
Backend Service (B - C)

- Backend Service
- Parse data from data source
- Calculate trend
- Apply sampling
- Write the response
Frontend Application (E - F)

- Frontend application
  - Transform to Google Charts format
  - Render the Chart
End-to-End Time – Scenario 1

- Scenario - 1
  - 3 Day raw data (10sec interval)
  - 60% Down-sampling
  - ~18000 point for rendering
End-to-End Time – Scenario 2

- Scenario - 2
  - 2 Month raw data (10sec interval)
  - Variable Down-sampling
    - Target 9000 final points
Evaluation Results

• Successfully deliver analytics
• In less than 4-sec time

• Fetch data 20% - 30% of total time
• Abstraction(s) does not come for free
• Google Chart Limits
  • Unresponsive after ~9000 points
  • Significant Impact to total response time
Conclusion

• In this thesis
  • Identified the missing of a ‘middle’ abstraction layer

• To advance the state of the art
  • Utilize service topology
    • provide a more structured visualization approach for the cloud usage analytics.

• Considering elastic cloud services
  • Use ‘Resizing Action’ as Add-on information to analytics
Conclusion

• We implemented an analytics tools
  • Targets to help the developers.
  • Connects to different data sources to obtain the data for the analytics
  • Provide topology-aware analytics for elastic cloud services

• We evaluated our tool
  • Provides accurate analytics results using metering data in less than 4 seconds.
Future Work / Plans

• Increase the variety of offered analytics
  • In-depth analysis
  • Behavior forecasting
• Explore ‘ways’ to increase performance
  • Usage Analytics Frameworks (e.g. Storm)
  • Stream processes
  • But, not all time-series statistic can be “stream-lined”
Future Work / Plans

• In the current state, we only support accessing
  • JCatascopia monitoring system API
  • Various user or system exported files that follow a specific format

• Expand the supported Data-Sources
  • Build additional “connectors”
  • Based on our Abstraction Model
  • Target Cloud Monitoring Systems
    • or any service that can provide us with data to enhance the extracted insights.
Acknowledgements

www.celarcloud.eu

co-funded by the
European Commission

source code: https://github.com/CELAR/cloud-is
Thank you!

Laboratory for Internet Computing
Department of Computer Science
University of Cyprus

http://linc.ucy.ac.cy