How to Squeeze a Cloud

Eliot Salant – salant@il.ibm.com
IBM Research – Haifa
Motivation
Current SotA
Strategies for improved performance
Literature
CloudWave
Future ideas
Everyone knows that virtualization takes better advantage of physical resources...
But in actuality, data centre server efficiency is still low...

And this is for a Twitter cluster managed by Mesos...

So what's going on here?
Understanding the challenges in Cloud resource scheduling

Estimating resource requirements

Determining workload placement
In IaaS systems (OpenStack, EC2...) typically VMs of a given **flavor** are requested

- Template values for RAM, number of cores, disk size, bandwidth cap, etc.

In Cluster Computer, **resource reservations** are (ultimately) given to the Cluster Framework Manager (e.g. Mesos, Google Borg...)
## Amazon EC2 – General Purpose VMs

<table>
<thead>
<tr>
<th>Model</th>
<th>vCPU</th>
<th>CPU Credits / hour</th>
<th>Mem (GiB)</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.micro</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>EBS-Only</td>
</tr>
<tr>
<td>t2.small</td>
<td>1</td>
<td>12</td>
<td>2</td>
<td>EBS-Only</td>
</tr>
<tr>
<td>t2.medium</td>
<td>2</td>
<td>24</td>
<td>4</td>
<td>EBS-Only</td>
</tr>
<tr>
<td>t2.large</td>
<td>2</td>
<td>36</td>
<td>8</td>
<td>EBS-Only</td>
</tr>
</tbody>
</table>
Estimating resource size is hard

Workload can change day-to-day or over the day

Analytics tasks depend on size of data set and complexity
Determining workload location is important..

• Co-located tasks can interfere with one another
  • Contention for CPU, cache, network, I/O, memory...

• Heterogeneity of servers in a data centre
  • Average life-cycle of a data centre is 15 years
  • Workloads that are sensitive to hardware configuration can run 2x’s as slow in worst case
Initial allocation and placement

• Resource allocation is typically done by:
  • Modelling – tightly coupled to a specific workload
  • Profiling – typically requires runs of several minutes
  • “Best guess” with future auto-scale anticipated
  • Machine learning/classification techniques for workloads and applications

• Resource placement for IaaS and PaaS is typically out of a user’s control
Getting the initial deployment right is more efficient than trying to fix it later on.
Machine Learning and Classification Techniques
• DejaVu (N. Vasic et al, 2012) – classification of workloads using workload signatures
  • Profiling done using low-level metrics from cloud benchmarks
  • Clustering techniques applied to collected behavior metrics
  • Performance degradation in production (vs. in isolation) is assumed to be due to interference
  • Database of previous resource allocations per classification

• Paragon (C. Delimitrou and C. Kozyrakis, 2013) – scalable classification of workloads with respect to heterogeneity and resource interference
  • Recommendation system based on the Netflix Challenge (Collaborative Filtering)
  • Micro-benching for 10 sources of interference
  • Offline profiling of a small number of applications on heterogeneous server configurations
Quasar (Delimitrou and Kozyrakis - 2014): Extension of Paragon ideas

- Handles both resource allocation and assignment
- Profiling for scale-up, scale-out, heterogeneity and interference
A closer look at Paragon/Quasar core ideas

• Recommendation system – finds similarities between workloads for incoming and historical based on server configurations and interference
  • Minimal *a priori* knowledge of incoming application
  • Fills in a sparse array for the incoming workload

• Runtime profiling runs can be done on only two different server configurations
  • Tens of server configurations possible
  • Extensive offline profiling of 20-30 workloads done on varying number of nodes Influencing parameters (e.g. interference, heterogeneity ...) considered independent
SVD overview in Paragon/Quasar

\[ A_{m,n} = U \cdot \Sigma \cdot V^T \]

Server configuration

- State per workload and per server is maintained
- Greedy allocation is used

C. Delimitrou and C. Kozyrakis
“Quasar: Resource-efficient and QoS-aware Cluster Management”
But bad things can happen to good initial deployments...

- Initial assessment wrong
- Workload changes phase
- Interference changes
Need to Monitor the Cloud Ecosystem

- Icinga
- Nagios
- Zabbix
- Observium
When changes need to be made
Most commercial PaaS systems support auto-scaling, but...

IBM Bluemix
Vmware CloudFoundry + RightScale
OpenShift (RedHat)
Google Autoscaler
Azure
The Cloud is a multi-layered entity

Adaptation actions exist for each layer

Changes to one layer may affect another

Clarissa Cassales Marquezan et al,
Towards Exploiting the Full Adaptation Potential of Cloud Applications,
PESOS ’14, 2014
So when adaptation is required, how can it be done?
How Paragon and Quasar handle adaptation

- Scale-up and scale-out behavior classified in advance.
- When a workload underperforms, reclassify and readjust resources.
- Proactively check for misclassification by injecting interference and observing behavior.
But, what they don’t handle...

• Migration not supported
  • Time to carry out migration
• Business policy and cost
The CloudWave Approach
The CloudWave approach

Concept:

**Enactment point** is any combination of observed metrics.

**Enactment point triggering** notifies the Adaptation Engine.

Adaptation Engine **mediates collaboration** between the infrastructure and the application.
Physical Monitoring levels

Virtual

Application

Probe

Logfiles

Monitoring bus

Trigger (enactment) point definition

CEP

Adaptation request

Persistent store

SLA
Main project concepts

- Infrastructure monitoring
- Application monitoring
- Execution
- Analytics
- Adaptation engine
- FDD
Building the Adaptation Engine

Monolithic engine

Consolidator
- Evaluator 1
- Evaluator n
- Target Engine 1
- Target Engine 2
- Engine 1
- Engine n

Constraint Engine

(Action, Target, Score)
Hierarchical arrangement of sub-engines

- Solution Aggregator
- Solution Evaluation Engines
- Target Recommendation Engines
- Action Recommendation Engines
- Constraint Engines

Potential adaptation actions

Recommended action

SLA

Potential adaptation actions
Types of Adaptation Engines

• SLA Constraints
• Application Model
• Hardware
• Analytics -> e.g. Quasar-like classification
• Network
• DevOps Recommendations
• Risk
Handling Application Adaptation
Defining what we mean

Web-based client

Application Business Layer Logic

Load balancer

Logical Application Architecture Layer

Client-side logic can grey out options

Logic can monitor timeouts, QoS etc

Application Server, Web server...
Coordinated Adaptation

In theory, application is written to negotiate behavior changes. Both infrastructure and application will optimally change.
What negotiations would require

- Language for expressing adaptation capabilities
- Protocol for communicating capabilities
- Method for invoking adaptation
In practice...

Throw it over the wall when there is no effective infrastructure adaptation action!
Research Challenges: To boldly go forth...
Can business level metrics be translated to infrastructure metrics?
A lot of research has gone into this in the past... CloudWave examining statistic-based ways to map the metrics - obtaining big data is hard though
Coordinated Adaptation with IoT Devices

Where is the computation done?
What is the energy tradeoff?
Security issues?
Latency issues?
Adaptations need to consider the semantics of the application, especially if profiling not used.
Adaptation of the Ecosystem
Example – Do you really want to migrate a database on a segment with low bandwidth?
Data on host may restrict relocation

Capture developer knowledge

Some strategies to improve Cloud performance

Initial allocation and placement

Performance monitoring
Some strategies to improve Cloud performance
[YUVAL’S SLIDES]