HCLOUD: RESOURCE-EFFICIENT PROVISIONING IN SHARED CLOUD SYSTEMS

Christina Delimitrou\textsuperscript{1} and Christos Kozyrakis\textsuperscript{2}

\textsuperscript{1}Stanford/Cornell University, \textsuperscript{2}Stanford University/EPFL

http://mast.stanford.edu

ASPLOS – April 5\textsuperscript{th} 2016
Executive Summary

- Problem: cloud provisioning is difficult
  - Many resource offerings ⇒ resource/cost inefficiencies
  - Interference from other users ⇒ performance jitter

- HCloud: resource-efficient public cloud provisioning
  - User provides resource reservations performance goals
  - Automatic selection of instance type and configuration
  - Adjust allocations between reserved and on-demand
  - High utilization and low performance jitter
Motivation
<table>
<thead>
<tr>
<th>AWS Instance</th>
<th>Google Instance</th>
<th>Azure Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>n1-standard-1</td>
<td>A0</td>
</tr>
<tr>
<td>t2.micro</td>
<td>n1-standard-2</td>
<td>A1</td>
</tr>
<tr>
<td>t2.small</td>
<td>n1-standard-4</td>
<td>A2</td>
</tr>
<tr>
<td>t2.medium</td>
<td>n1-standard-8</td>
<td>A3</td>
</tr>
<tr>
<td>t2.large</td>
<td>n1-standard-16</td>
<td>A4</td>
</tr>
<tr>
<td>m4.large</td>
<td>n1-standard-32</td>
<td>A5</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>n1-highmem-2</td>
<td>A6</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>n1-highmem-4</td>
<td>A7</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>n1-highmem-8</td>
<td>A8</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>n1-highmem-16</td>
<td>A9</td>
</tr>
<tr>
<td>m3.medium</td>
<td>n1-highmem-32</td>
<td>A10</td>
</tr>
<tr>
<td>m3.large</td>
<td>n1-highcpu-2</td>
<td>A11</td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>n1-highcpu-4</td>
<td>D1</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>n1-highcpu-8</td>
<td>D2</td>
</tr>
<tr>
<td>c4.large</td>
<td>n1-highcpu-16</td>
<td>D3</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>n1-highcpu-32</td>
<td>D4</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>f1-micro</td>
<td>D11</td>
</tr>
<tr>
<td>c4.4xlarge</td>
<td>a1-small</td>
<td>D12</td>
</tr>
<tr>
<td>Amazon Web Services</td>
<td>Google Cloud Platform</td>
<td>Microsoft Azure</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>t2.nano</td>
<td>t2.nano</td>
<td>A0</td>
</tr>
<tr>
<td>t2.micro</td>
<td>t2.micro</td>
<td>A1</td>
</tr>
<tr>
<td>t2.small</td>
<td>t2.small</td>
<td>A2</td>
</tr>
<tr>
<td>t2.medium</td>
<td>t2.medium</td>
<td>A3</td>
</tr>
<tr>
<td>t2.large</td>
<td>t2.large</td>
<td>A4</td>
</tr>
<tr>
<td>m4.large</td>
<td>m4.large</td>
<td>A5</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>A6</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>m4.2xlarge</td>
<td>A7</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>m4.4xlarge</td>
<td>A8</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>m4.10xlarge</td>
<td>A9</td>
</tr>
<tr>
<td>m3.medium</td>
<td>m3.medium</td>
<td>A0</td>
</tr>
<tr>
<td>m3.large</td>
<td>m3.large</td>
<td>A1</td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>m3.xlarge</td>
<td>A2</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>m3.2xlarge</td>
<td>A3</td>
</tr>
<tr>
<td>c4.large</td>
<td>c4.large</td>
<td>A4</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>c4.xlarge</td>
<td>A5</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>c4.2xlarge</td>
<td>A6</td>
</tr>
<tr>
<td>c4.4xlarge</td>
<td>c4.4xlarge</td>
<td>A7</td>
</tr>
<tr>
<td>c4.8xlarge</td>
<td>c4.8xlarge</td>
<td>A8</td>
</tr>
<tr>
<td>c3.large</td>
<td>c3.large</td>
<td>A9</td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>c3.xlarge</td>
<td>A10</td>
</tr>
<tr>
<td>c3.2xlarge</td>
<td>c3.2xlarge</td>
<td>A11</td>
</tr>
<tr>
<td>c3.4xlarge</td>
<td>c3.4xlarge</td>
<td>A12</td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>c3.8xlarge</td>
<td>A13</td>
</tr>
<tr>
<td>r3.large</td>
<td>r3.large</td>
<td>A14</td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>r3.xlarge</td>
<td>A15</td>
</tr>
<tr>
<td>g1.medium</td>
<td>g1-medium</td>
<td>A16</td>
</tr>
<tr>
<td>g1.small</td>
<td>g1-small</td>
<td>A17</td>
</tr>
</tbody>
</table>

- **Reserved**: Fixed commitment for optimal performance.
- **On-demand**: Pay as you go with flexible capacity.
- **Spot**: Reserve capacity and save up to 90% compared to on-demand pricing.
- **Discounted**: Pay less for the same performance.
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>t2.nano</td>
<td>t2.nano</td>
<td>n1-standard-1</td>
<td>n1-standard-1</td>
</tr>
<tr>
<td>t2.micro</td>
<td>t2.micro</td>
<td>t2.micro</td>
<td>n1-standard-2</td>
<td>n1-standard-2</td>
</tr>
<tr>
<td>t2.small</td>
<td>t2.small</td>
<td>t2.small</td>
<td>n1-standard-4</td>
<td>n1-standard-4</td>
</tr>
<tr>
<td>t2.medium</td>
<td>t2.medium</td>
<td>t2.medium</td>
<td>n1-standard-8</td>
<td>n1-standard-8</td>
</tr>
<tr>
<td>t2.large</td>
<td>t2.large</td>
<td>t2.large</td>
<td>n1-standard-16</td>
<td>n1-standard-16</td>
</tr>
<tr>
<td>m4.large</td>
<td>m4.large</td>
<td>m4.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>n1-highmem-2</td>
<td>n1-highmem-2</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>m4.2xlarge</td>
<td>m4.2xlarge</td>
<td>n1-highmem-4</td>
<td>n1-highmem-4</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>n1-highmem-2</td>
<td>n1-highmem-2</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>m4.2xlarge</td>
<td>m4.2xlarge</td>
<td>n1-highmem-4</td>
<td>n1-highmem-4</td>
</tr>
<tr>
<td>m4.large</td>
<td>m4.large</td>
<td>m4.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>m3.large</td>
<td>m3.large</td>
<td>m3.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>m4.xlarge</td>
<td>n1-highmem-2</td>
<td>n1-highmem-2</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>m3.2xlarge</td>
<td>m3.2xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c4.large</td>
<td>c4.large</td>
<td>c4.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>c4.xlarge</td>
<td>c4.xlarge</td>
<td>n1-highcpu-32</td>
<td>n1-highcpu-32</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>c4.2xlarge</td>
<td>c4.2xlarge</td>
<td>f1-micro</td>
<td>f1-micro</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>c4.xlarge</td>
<td>c4.xlarge</td>
<td>g1-small</td>
<td>g1-small</td>
</tr>
<tr>
<td>c4.8xlarge</td>
<td>c4.8xlarge</td>
<td>c4.8xlarge</td>
<td>g1-small</td>
<td>g1-small</td>
</tr>
<tr>
<td>c3.large</td>
<td>c3.large</td>
<td>c3.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>c3.xlarge</td>
<td>c3.xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.2xlarge</td>
<td>c3.2xlarge</td>
<td>c3.2xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.4xlarge</td>
<td>c3.4xlarge</td>
<td>c3.4xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>c3.8xlarge</td>
<td>c3.8xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>r3.large</td>
<td>r3.large</td>
<td>r3.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>c3.xlarge</td>
<td>c3.xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.2xlarge</td>
<td>c3.2xlarge</td>
<td>c3.2xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.4xlarge</td>
<td>c3.4xlarge</td>
<td>c3.4xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>c3.8xlarge</td>
<td>c3.8xlarge</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
<tr>
<td>r3.large</td>
<td>r3.large</td>
<td>r3.large</td>
<td>n1-standard-32</td>
<td>n1-standard-32</td>
</tr>
</tbody>
</table>

- Cost, predictability, availability, flexibility, spin-up overheads, retention time, load fluctuation, external load, sensitivity to interference, ...
Overprovisioning
High cost &
Unpredictable performance
Cloud Provisioning Goals

✗ Determine appropriate instance size/type

✗ Determine appropriate instance configuration

✗ Dynamically adjust allocation decisions at runtime
Cloud Provisioning Scenarios

- Batch analytics, latency-critical applications, and scientific workloads in all scenarios
  - Batch: 50%
  - Latency: 40%
  - Scientific: 10%
Provisioning Baselines

- **All reserved (SR):**
  - 16vCPU instances on GCE, no external load
  - Predictable performance, high cost, low flexibility
  - Applications provisioned for peak requirements

- **All on-demand:**
  - Largest instances (**OdF**): only 16vCPU instances
  - Mixed instances (**OdM**): mix of large and smaller instances
  - Interference, low cost, high flexibility

- **Resource management:**
  - Least-loaded scheduler
Extracting Resource Preferences

- Exhaustive characterization is infeasible

- **Heterogeneity**
- **Interference**
- **Resources per server**
- **Resource ratio**
- **Number of servers**
- **Application params**

Combinations

- 10 servers
  - 40 apps
- 100 servers
  - 300 apps
- 1000 servers
  - 1200 apps

Systems

1,000,000,000

1,000,000

1,000

0
Quasar Overview

[ASPLOS’14]

Scheduler

Resource preferences

Cluster

User

App

Data mining

Profiling

QoS

App

App

App

App

App

App

Cluster
Quasar Overview

Scheduler

User

Profiling

Data mining

Resource preferences

App

Greedy scheduler

Resource selection

Cluster

[ASPLOS’14]
Quasar Overview

[ASPLOS’14]

Scheduler

Resource preferences

Greedy scheduler

App

User

Profiling

Data mining

Resource selection

Cluster
Quasar Overview

[ASPLOS’14]

Scheduler

User → Cluster


Resource selection [50msec-2sec]
Evaluation

**Static Scenario**
- **Execution Time (min)**
- **Request Latency (usec)**

**High Variability Scenario**
- **Execution Time (min)**
- **Request Latency (usec)**

**Legend:**
- **with profiling info**
- **without profiling info**
Evaluation

[Graphs showing performance metrics (Execution Time, Request Latency, Cost) for different scenarios (Static and High Variability) with and without profiling information.]

- **Static Scenario**: Comparison of Execution Time, Request Latency, and Cost for SR, OdF, and OdM with and without profiling info.
- **High Variability Scenario**: Similar comparisons with additional data points highlighting variability.

Legend: 
- Solid line: Static
- Dotted line: Low Variability
- Dashed line: High Variability
Cloud Provisioning Goals

✔ Determine appropriate instance size/type

✗ Determine appropriate instance configuration

✗ Dynamically adjust allocation decisions at runtime
Hybrid Cloud Resource Allocation

- **Insight:**
  - Combine reserved and on-demand resources

- **Challenge:**
  - Separate applications between reserved (~private) and on-demand (public) resources
Hybrid Allocation Policies

✗ Account for application interference sensitivity
Hybrid Allocation Policies

✓ Account for application interference sensitivity

✗ Do not overload reserved resources
Hybrid Allocation Policies

✓ Account for application interference sensitivity

✓ Do not overload reserved resources

✗ Dynamic decisions (e.g., utilization limits)
HCloud

Insights:
- Account for interference sensitivity
- Set load limits dynamically
HCloud

- **Insights:**
  - Account for interference sensitivity
  - Set load limits dynamically
Insights:

- Account for interference sensitivity
- Set load limits dynamically
Insights:
- Account for interference sensitivity
- Set load limits dynamically (feedback loop on queue length)
HCloud

- **Insights:**
  - Account for interference sensitivity
  - Set load limits dynamically (feedback loop on queue length)

- What signals a sensitive job?
HCloud

- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically
- What signals a sensitive job?
Hybrid Allocation Policies

✓ Account for application interference sensitivity
✓ Do not overload reserved resources
✓ Dynamic decisions (e.g., utilization limits)
Evaluation

Static Scenario

High Variability Scenario

Execution Time (min)

Request Latency (usec)

Cost

with profiling info  
without profiling info

Reserved Cost  
On Demand Cost

SR  HF  HM

SR  HF  HM

SR  HF  HM

SR  HF  HM

SR  HF  HM

SR  HF  HM
Cloud Provisioning Goals

✔ Determine appropriate instance size/type

✔ Determine appropriate instance configuration

✗ Dynamically adjust allocation decisions at runtime
Adjusting Allocations at Runtime

- **Runtime**: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

---

**Reserved Resources**

- Allocated
- Hadoop
- Spark
- memcached

**On-Demand Resources**
Adjusting Allocations at Runtime

- Runtime: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

![Reserved Resources](image1)
![On-Demand Resources](image2)
Adjusting Allocations at Runtime

- **Runtime**: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

![Graph showing resource usage over time](image)

- **Reserved**: resources allocated on demand
- **On-demand**: resources allocated on demand
Adjusting Allocations at Runtime

- **Runtime**: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

![Reserved Resources](image1)

![On-Demand Resources](image2)
Conclusions

- **HCloud**: hybrid cloud provisioning (reserved & on-demand)
  - Account for interference sensitivity
  - Account for performance-cost tradeoffs
  - Adjust to load fluctuations
  - 2.1x better performance than on-demand, 50% lower cost than reserved

- See paper for:
  - Performance unpredictability analysis on EC2 and GCE
  - Sensitivity studies on system & app parameters
    - Spin-up overheads, cost, retention time, load, app characteristics, external load, ...
  - Further workload scenarios
Questions??

- HCloud: hybrid cloud provisioning (reserved & on-demand)
  - Account for interference sensitivity
  - Account for performance-cost tradeoffs
  - Adjust to load fluctuations
  - 2.1x better performance than on-demand, 50% lower cost than reserved

- See paper for:
  - Performance unpredictability analysis on EC2 and GCE
  - Sensitivity studies on system & app parameters
    - Spin-up overheads, cost, retention time, load, app characteristics, external load, …
  - Further workload scenarios
Questions??

Thank you
Resource Efficiency

- **Configuration: SR**
- **Configuration: OdF**
- **Configuration: OdM**
- **Configuration: HF**
- **Configuration: HM**

The diagrams illustrate the resource efficiency over time with different configurations.