Bidding for **Highly Available Services with Low Price** in Spot Instance Market

Weichao Guo, Kang Chen, Yongwei Wu, and Weimin Zheng

Tsinghua University
On-Demand Instance vs. Spot Instance

• Price Model
  – On-Demand: pre-define
  – Spot: fluctuate based on supply & demand

• Failure Model
  – On-Demand: SLAs
  – Spot: terminated when spot price exceeds bid
Cost-Efficient Computing with Spot Instances

- Optional accelerators for MapReduce jobs
  
  [HotCloud ’10]
Cost-Efficient Computing with Spot Instances

• Adapting FT techniques for divisible parallel jobs [CLOUD ‘10, HotCloud ’11, etc.]
  • Re-execution
  • Checkpointing
  • Task migration

Fixed Bidding

Time Flexible
Cost-Efficient Computing with Spot Instances

• Profit aware dynamic bidding from a cloud service broker’s perspective [INFOCOM ‘12]
Cost-Efficient Computing with Spot Instances

High available services with low prices?
Distributed Service Basics

- State Machine Replication

Quorum-based Protocols
Distributed Service Basics

• Acceptance Set $\mathcal{A}$
  - Intersection $\forall S, T \in \mathcal{A}, S \cap T \neq \emptyset$
  - Monotonicity $\forall T \supseteq S, S \in \mathcal{A} \implies T \in \mathcal{A}$

• Availability

\[ A_{\mathcal{A}} = \sum_{S \in \mathcal{A}} \left( \prod_{i \in S} (1 - p_i) \prod_{j \in \overline{S}} p_j \right) \]
Distributed Service with Spot Instances

RSM  O  0.044 $/H  O  0.044 $/H  O  0.047 $/H  O  0.044 $/H  O  0.044 $/H

~25.5 s

RSM  s  0.008 $/H  s  0.008 $/H  s  0.009 $/H  s  0.008 $/H  s  0.009 $/H

>1500 s
Distributed Service with Spot Instances

To Improve Availability:

• Higher Bids
• More nodes
Contributions

• Spot Instance Failure Model
  – Availability analysis
  – Failure probability estimation

• Bidding Framework
  – Cost minimization problem
  – Online bidding strategy
Outline

• Problem Formalization
  – Spot Instance Failure Model
  – Cost Minimization Problem

• Bidding Framework
  – Failure Probability Estimation
  – Online Bidding

• Experiment
Spot Instance Failure Model

On-Demand Instance Failures

Out-of-bid Failure
Spot Instance Failure Model

$\Pr(p(t) > b)$

<table>
<thead>
<tr>
<th>Time (9:00 AM - 11:00 AM in June 24th, 2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:38</td>
</tr>
<tr>
<td>0.002</td>
</tr>
</tbody>
</table>

Spot Price (USD)

$\$0.99$
Spot Instance Failure Model

Spot Price

\[ \tau_1 \]

\[ \tau_2 \]

\[ \tau_3 \]

\[ (s_i, i = 1, 2, \ldots, n) \]

\[ (\tau_i, i = 1, 2, \ldots, n) \]
Spot Instance Failure Model

• Semi-Markovian Chain’s Stochastic Kernel

\[ Q(i, j, k) = (q_{i,j,k}; s_i, s_j \in S, k \in T) \]

where

\[ q_{i,j,k} = Pr(S_{n+1} = s_j, S_n = s_i, \tau_n = k) \]
Spot Instance Failure Model

• Failure probability at time $t$

$$FP(t) = 1 - (1 - FP') \cdot (1 - Pr(p(t) > b))$$

• Failure probability in time duration $d$

$$\int_0^d FP(t) \, dt$$
Cost Minimization Problem

Availability Zones & i.i.d.
Cost Minimization Problem

- Bidding Interval & Objective
Cost Minimization Problem

\[
\min_{i=1}^{n} \sum b_i
\]

s.t.
\[
\sum_{i=1}^{n} \epsilon (b_i - p_i) \geq m
\]

and
\[
\AA_o(S_o, FP') - \AA_o(S_s, FP(b)) < \varepsilon
\]
Outline

• Problem Formalization
  – Spot Instance Failure Model
  – Cost Minimization Problem

• Bidding Framework
  – Failure Probability Estimation
  – Online Bidding

• Experiment
Bidding Framework

Distributed System  \(\rightarrow\) Constraints  \(\rightarrow\) Online Bidding  \(\rightarrow\) Bids  \(\rightarrow\) Cloud Provider

Spot Instance Failure Model
Bidding Framework

• Keeping Safety at Reconfiguration

Interval A

Interval B

Members Change

Members Change
Failure Probability Estimation

• Maximum Likelihood Estimator (MLE)

\[ \hat{q}_{i,j,k} = \frac{N_{i,j}^k}{N_i}, \text{ if } N_i \neq 0, \text{ otherwise } \hat{q}_{i,j,k} = 0 \]

• Failure Probability (in a time unit) (FP)

\[ 1 - (1 - FP') \cdot \sum_{j=p}^{b} q_{p,j,k}, p < b < o \]
Online Bidding

• Constraint Without Analytic Expression

[Diagram]

• Exhaustive search?
  – Traverse space $m^n$

• Keeping FPs same & bidding greedily
For each possible \( n \), get \( \text{FP} \) with given \( \text{Availability} \).

For each AZ, get min \( \text{bid} \) with given \( \text{FP} \).

Select \textbf{bids} of AZs in a greedy way.

Choose the lowest bidding CONFIG.
Outline

• Problem Formalization
  – Spot Instance Failure Model
  – Cost Minimization Problem
• Bidding Framework
  – Failure Probability Estimation
  – Online Bidding
• Experiment
Experiment

• Whether cost has been reduced?

• What about the availability achieved?
Experiment Setup

• Experimental Systems
  – Distributed Lock Service (‘linux.m1.small’)
  – Erasure Code Based Distributed Storage Service (linux.m3.large)

• Estimator Training
  – ~ 3 months spot price data

• Baseline
  – 5 On-Demand Instances

• Straw-man Scheme
  – $Extra(m, p)$: Adding $m$ extra nodes & setting bids as spot price + extra portion $p$
Experiment Setup

• Test Cases

- Micro-Benchmark
  - 1-month-long test
- FP Estimator Bidding Schemes
  - 1-week-long running on EC2
  - Out-of-bid Failure Only
  - 11-week-long spot prices replay
**Experiment Results**

- **Feasibility**

<table>
<thead>
<tr>
<th>COST</th>
<th>AVAILABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jupiter</td>
<td><img src="images/smiley.png" alt="Smiley" /></td>
</tr>
<tr>
<td>Extra</td>
<td><img src="images/smiley.png" alt="Smiley" /></td>
</tr>
</tbody>
</table>

- **Micro-Benchmark**

  ![Graph showing estimating deviation for 'linux.m1.small' and 'linux.m3.large'.]
Experiment Results

- Cost under different bidding intervals

Jupiter costs only ~1/5 and ~1/6 of the baseline
Experiment Results

• Availability under different bidding intervals

Jupiter kept the service availability level close to the baseline
Summary

• Market pricing has bring a new vision of Cloud Computing
• Spot instance failure model challenges the reliability of quorum-based system
• The problem is formalized by Spot Instance Failure Model and Non-linear Programming
• Our bidding framework can obtain cost efficiency while still keeping high availability
THANKS FOR YOUR ATTENTION!

Questions?