Cloud Index Tracking:
Enabling Predictable Costs in Cloud Spot Markets

Supreeth Shastri and David Irwin
University of Massachusetts Amherst
Spot Servers are gaining significance in the cloud

Servers that may terminate anytime after an advance warning period
Spot Servers are gaining significance in the cloud

Servers that may terminate anytime after an advance warning period
Spot Servers are gaining significance in the cloud

Servers that may **terminate anytime** after an **advance warning** period

---

**Cost**

- **Reserved**
- **On-demand**
- **Spot**

**Availability**

- **Guaranteed, Non-revocable**
- **Not guaranteed, Non-revocable**
- **Not guaranteed, Revocable**

---

*Note: The image features a cost vs. availability diagram with the following key points:*

- **Reserved** services are expensive but have guaranteed availability.
- **On-demand** services are generally cheaper but may have non-revocable availability.
- **Spot** services are the cheapest but have revocable availability, potentially terminating anytime after an advance warning period.*
Spot Servers are gaining significance in the cloud

Servers that may **terminate anytime** after an **advance warning** period

Spot instances helped scale our clusters up by 4X during the discovery of the **Higgs Boson**

Researchers built the **largest** HPC cluster in the cloud with **1.1 million** vCPUs on EC2 spot

---

**Cost**

- **Reserved**
- **On-demand**
- **Spot**

**Availability**

- **Guaranteed, Non-revocable**
- **Not guaranteed, Non-revocable**
- **Not guaranteed, Revocable**

**Expensive**

**Cheap**
Spot server pricing

while low on average, it is characterized by *variability* and *deliberate revocations*
Spot server pricing

while low on average, it is characterized by

variability and deliberate revocations

Predicting Spot Prices is an Active Area of Research

Ability to compare servers, plan IT budgets, and avoid disruptive revocations
Spot server pricing

while low on average, it is characterized by **variability** and **deliberate revocations**

Predicting Spot Prices is an Active Area of Research

Ability to compare servers, plan IT budgets, and avoid disruptive revocations

**2015**
- Bid [SIGCOMM]
- SpotOn [SoCC]
- Cumulon [VLDB]

**2016**
- No-bid [HotCloud]
- Flint [Eurosys]
- BOSS [Infocom]

**2017**
- Prob-Guarantee [SC]
- Proteus [EuroSys]
- Exosphere [SIGMETRICS]

**2018**
- LSTM [HPDC]
- Tributary [ATC]
Predicting Spot Prices is Important

Prior work models individual spot server prices based on their historical spot price data.
Accurately Predicting Spot Prices is Important. Difficult

Prior work models individual spot server prices based on their historical spot price data.
Accurately Predicting Spot Prices is Important

Prior work models individual spot server prices based on their historical spot price data.
Accurately Predicting Spot Prices is Difficult Important

Prior work models individual spot server prices based on their historical spot price data
Accurately Predicting Spot Prices is Important

Prior work models individual spot server prices based on their historical spot price data.

\[
68 \times 2 \times 2-5 \times 14 \times 2 = 7600^+ 
\]

- Hardware config
- OS types
- Zones (datacenters)
- Regions (country, state)
- Time commitments

worldwide markets

One size fits all model is unlikely

No visibility into market internals

Limited correlation with external variables
U.S. FUTURES

DOW JONES INDUS FUT 3/17 (D/JH7)
19,904.00 ▲ 60.00 (0.30%)

S&P 500 3/17 (ES/H7)
2,280.75 ▲ 6.25 (0.27%)

NASDAQ 100 3/17 (NQ/H7)
5,116.50 ▲ 20.20 (0.40%)
Key Insight: A Market-based Index for CLOUD
Key Insight: A Market-based Index for CLOUD

Rather than focusing exclusively on predicting individual servers, cloud users should make decisions based on broader market indices.
Cloud Index

- intuition for our hypothesis
- index construction methodology
- validation on Amazon EC2

Index-tracking

- techniques for predictability
- design of index-tracking by server hopping
- performance evaluation
Underlying Characteristics of Large Cloud Platforms
Underlying Characteristics of Large Cloud Platforms

1. Dependence of VMs

Spot markets originating from the same physical machine family are not free from mutual interference.
Underlying Characteristics of Large Cloud Platforms

1. Dependence of VMs
   Spot markets originating from the same physical machine family are not free from mutual interference

   Not all spot markets could be individually modeled and predicted
Underlying Characteristics of Large Cloud Platforms

1. Dependence of VMs
Spot markets originating from the same physical machine family are not free from mutual interference

Not all spot markets could be individually modeled and predicted

2. Stability of Idle Capacity
Aggregate idle VM capacity in public cloud datacenters tends to be stable
[SoCC 2014, SOSP 2017]
Underlying Characteristics of Large Cloud Platforms

1. Dependence of VMs
Spot markets originating from the same physical machine family are not free from mutual interference

Not all spot markets could be individually modeled and predicted

2. Stability of Idle Capacity
Aggregate idle VM capacity in public cloud datacenters tends to be stable [SoCC 2014, SOSP 2017]

If idle capacity were priced like commodity, its clearing price will be stable and predictable
Underlying Characteristics of Large Cloud Platforms

1. Dependence of VMs
   Spot markets originating from the same physical machine family are not free from mutual interference
   
   Not all spot markets could be individually modeled and predicted

2. Stability of Idle Capacity
   Aggregate idle VM capacity in public cloud datacenters tends to be stable
   [SoCC 2014, SOSP 2017]
   
   If idle capacity were priced like commodity, its clearing price will be stable and predictable

We hypothesize that observing spot markets at aggregate levels (say, server family or datacenter levels) should lead to stable prices
Constructing a Market Index for CLOUD
Characterizing an individual server $i$

Price = $P_i$, Memory = $M_i$ GB

Compute = $C_i$ ECUs

$$P_i^{\text{norm}} = \frac{P_i}{\sqrt{(C_i \cdot M_i)}}$$
Constructing a Market Index for CLOUD

Characterizing an individual server $i$

Price $= P_i$, Memory $= M_i$ GB
Compute $= C_i$ ECUs

$$P_i^{\text{norm}} = \frac{P_i}{\sqrt{(C_i \cdot M_i)}}$$

Characterizing a group of servers

Average of normalized prices

$$\text{Index-level} = \frac{\sum_{i=1}^{N} P_i^{\text{norm}}}{N}$$
Constructing a Market Index for CLOUD

**Characterizing an individual server \( i \)**

- Price = \( P_i \)
- Memory = \( M_i \) GB
- Compute = \( C_i \) ECUs

\[
P_i^{\text{norm}} = \frac{P_i}{\sqrt{(C_i \cdot M_i)}}
\]

**Characterizing a group of servers**

Average of normalized prices

\[
\text{Index-level} = \frac{\sum_{i=1}^{N} P_i^{\text{norm}}}{N}
\]

**Cloud index** value represents the average price per unit of compute time for the selected group of servers
Individual Server Level

Price (cents/hr)

bid level

March 1  April 1  May 1  June 1  July 1
Individual Server Level

Datacenter Level (US-West-1a)
Individual Server Level

Datacenter Level (US-West-1a)

Server Family Level (US-West-1a)

bid level
Price prediction is more accurate and stable at datacenter- and server family level than individual level.
Cloud Index

Index-tracking

intuition for our hypothesis
index construction methodology
validation on Amazon EC2
techniques for predictability
design of index-tracking by server hopping
performance evaluation
Design elements
Design elements

Index-tracking in financial markets

Investments that match the returns of an index.

Construct a portfolio such that its constituent items are same as those present in the index.
Design elements

Index-tracking in financial markets

Investments that match the returns of an index.

Construct a portfolio such that its constituent items are same as those present in the index.

Server hopping in cloud markets

A container that automatically hops spot VMs as market conditions change [SoCC 2017].

Increasing cost-efficiency, lowers revocations
Index Tracking by Server Hopping
Achieving index-level cost-efficiency despite market volatility
Index Tracking by Server Hopping

Achieving index-level cost-efficiency despite market volatility

1. Determine a broad set of candidate markets, and then compute its market index.
Index Tracking by Server Hopping

Achieving index-level cost-efficiency despite market volatility

1. Determine a broad set of candidate markets, and then compute its market index.

2. Host the application on a server that meets the index-level cost-efficiency.
Index Tracking by Server Hopping

Achieving index-level cost-efficiency despite market volatility

1. Determine a broad set of candidate markets, and then compute its market index.

2. Host the application on a server that meets the index-level cost-efficiency.

3. If market conditions violate the index invariant, then transparently hop to a better server.
Achieving index-level cost-efficiency despite market volatility

Index Tracking by Server Hopping

1. Determine a broad set of candidate markets, and then compute its market index
2. Host the application on a server that meets the index-level cost-efficiency
3. If market conditions violate the index invariant, then transparently hop to a better server

Server Choice

Select a server that shows best balance between risk (price volatility) vs. reward (cost-efficiency)

Sharpe ratio = \[
\frac{\langle I - \hat{P}_i \rangle}{\text{std-dev} \langle I - \hat{P}_i \rangle}
\]

$I = \text{Index-level}$, and $\hat{P}_i = \text{Spot server's normalized efficiency}$
LXC based prototype for EC2 spot markets

https://umass-sustainablecomputinglab.github.io/cloudIndex/
Evaluation

- Does index-tracking achieve **predictable expenses**?
- How does cost-availability of index-tracking **compare to others**?

LXC based prototype for EC2 spot markets

https://umass-sustainablecomputinglab.github.io/cloudIndex/
Evaluation

- Does index-tracking achieve **predictable expenses**?
- How does cost-availability of index-tracking **compare to others**?

We compare **three systems** for running **two classes** of applications on EC2 spot markets:

- **Spot server with static prediction (SpotFleet)**
- **Spot server with cost-based hopping (HotSpot)**
- **Spot server with index-tracking**
Long-running Single-node App
E.g., IoT sinks, crypto miners, p2p file trackers

Bulk-synchronous Parallel Jobs
MapReduce type workload from Google traces
Long-running Single-node App
E.g., IoT sinks, crypto miners, p2p file trackers

Bulk-synchronous Parallel Jobs
MapReduce type workload from Google traces
Long-running Single-node App
E.g., IoT sinks, crypto miners, p2p file trackers

Bulk-synchronous Parallel Jobs
MapReduce type workload from Google traces

![Graph showing cost and availability comparisons between Spot-fleet, Index-tracking, and HotSpot for both Long-running Single-node App and Bulk-synchronous Parallel Jobs. The graphs display cost in % on-demand and availability (%).]
Long-running Single-node App

E.g., IoT sinks, crypto miners, p2p file trackers

Bulk-synchronous Parallel Jobs

MapReduce type workload from Google traces
Long-running Single-node App
E.g., IoT sinks, crypto miners, p2p file trackers

Bulk-synchronous Parallel Jobs
MapReduce type workload from Google traces
Index-Tracking not only meets the **predicted cost-efficiency** but also achieves the **best cost-availability tradeoff** compared to other approaches.
Conclusion

Spot server markets enable inexpensive computing at scale but expose users to cost uncertainty.
Conclusion
Spot server markets enable inexpensive computing at scale but expose users to cost uncertainty

Cost Uncertainty

Affects app performance and user’s budget planning

Prior work focuses on history-based prediction
Spot server markets enable inexpensive computing at scale but expose users to cost uncertainty

Conclusion

Cost Uncertainty
Affects app performance and user’s budget planning
Prior work focuses on history-based prediction

Cloud Index Tracking
Propose market-based indices for EC2 spot servers
Design technique for index tracking by server hopping
Conclusion
Spot server markets enable inexpensive computing at scale but expose users to cost uncertainty

Cost Uncertainty
Affects app performance and user’s budget planning
Prior work focuses on history-based prediction

Cloud Index Tracking
Propose market-based indices for EC2 spot servers
Design technique for index tracking by server hopping

Evaluations
Index-level cost-efficiency
vs. other approaches
Achieves predictable costs with higher availability across applications