

SPA: Harnessing Availability in the AWS Spot Market

Walter Wong¹, Aleksandr Zavodovski¹, Lorenzo Corneo²,
Nitinder Mohan³, Jussi Kangasharju¹

¹University of Helsinki, Finland

²Uppsala University, Sweden

³Technical University Munich, Germany



UNIVERSITY OF HELSINKI



UPPSALA
UNIVERSITET

Motivation – AWS Cloud Computing

- Cloud computing eases the deployment of new services due to the elastic and on-demand resource provisioning
 - No need for upfront capital expenditure
 - Pay-as-you-go model
 - Adapts to seasonal peak demands, e.g., Black Friday, etc.
- Cloud providers such as AWS provide opportunistic spot instances
 - Up to 90% off compared to the on-demand pricing
 - Caveat: no availability guarantee (instances are shut down with a 2-minute warning → need to migrate before eviction)



Motivation – Research Questions

Can we use spot instances with (almost) the same availability as on-demand instances but at a fraction of the on-demand pricing?

Can we anticipate any possible spot instance eviction and migrate to new instances?

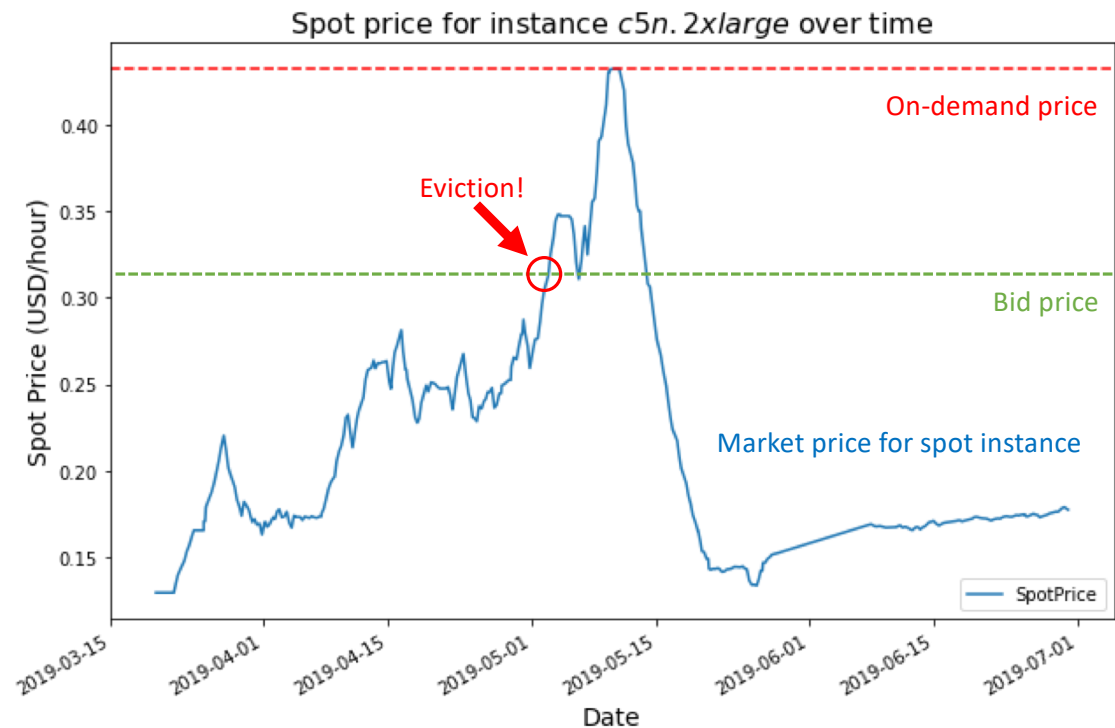
Outline

- AWS Spot Market Analysis
- SPA Proposal
- Evaluation and Results
- Conclusion

AWS Spot Market Analysis – Eviction Process

Main characteristics

- Market price
- Bid price
- On-demand price
- Eviction (market price > bid price)



AWS has 32 availability zones and 252 instance types resulting in ~8k individual spot markets

AWS Spot Market Analysis

Goal

- Find patterns in the AWS pricing data that explains the spot instance eviction process

Methodology

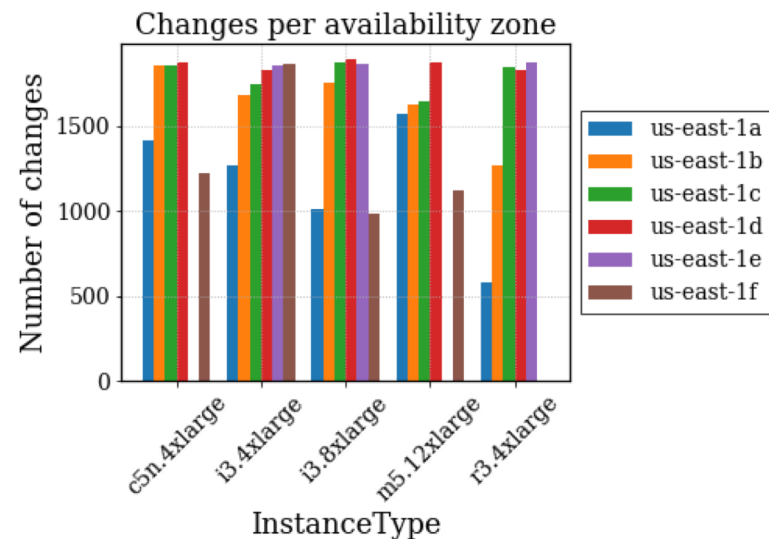
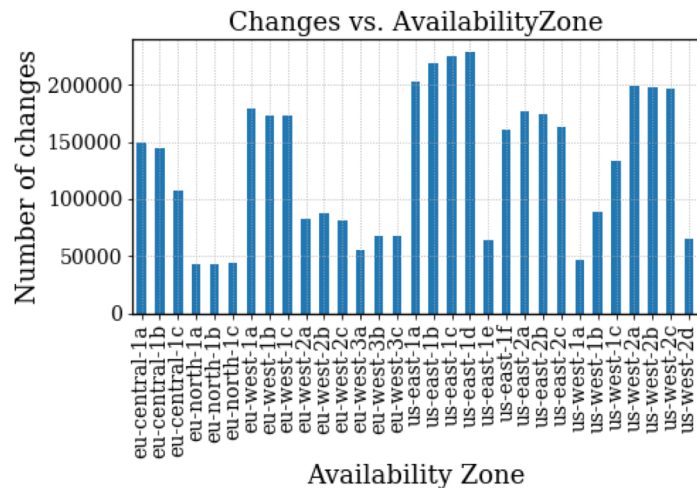
- Analysis of AWS Spot instance attributes (region, availability zone, and instance type)
- Analysis of the pricing data for instance volatility, average price update time and minimum price update interval

Expected results

- Find spot instances that are less volatile
- Find the minimum lifetime period for a spot instance

AWS Spot Market Analysis

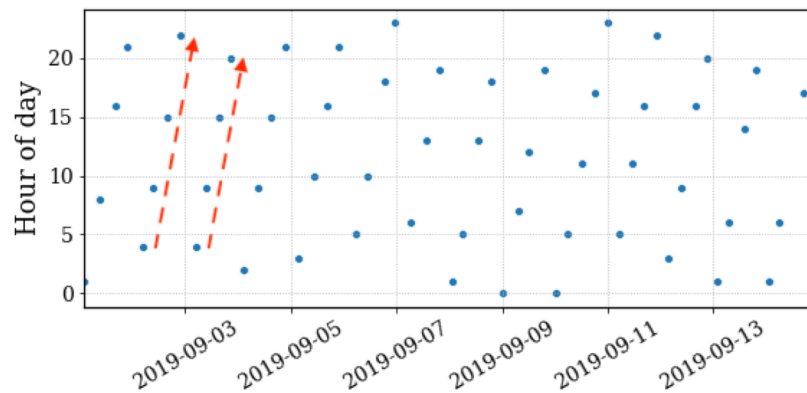
- AWS Spot Instance Attributes
 - Region: business requirements, e.g., pricing, lower latency to the customer, etc.
 - Availability Zone: redundant data-center in the same region
 - Instance Type: CPU, RAM, storage, etc.
 - Volatility: number of price changes within an interval



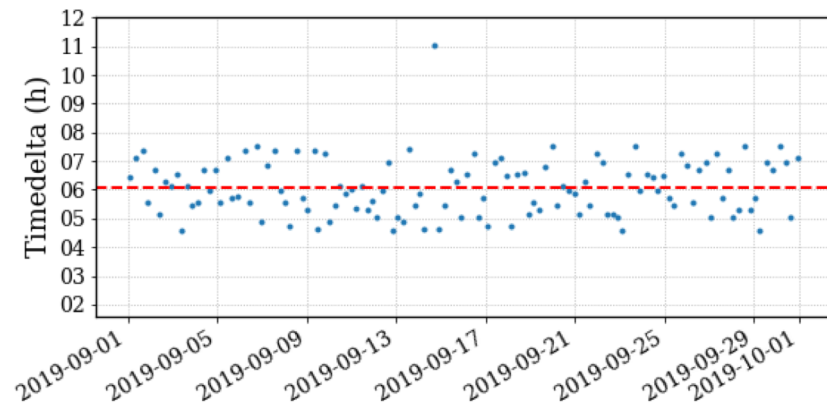
Key takeaway: a given spot instance has different pricing and volatility across different availability zone.

AWS Spot Market Analysis

- AWS Spot Instance Insights
 - Avg. Price Change Interval: almost constant price update time for spot instances
 - Minimum price update interval: minimum interval between any price change (4h32min)
 - Percentile of 1% price change interval: remove any outlier in the dataset



(a)



(b)

Key takeaway: AWS spot instances have a price update pattern along the day (Fig. a) and have a minimum price update time (Fig. b). By removing the outliers, the 1% percentile price update interval is constant across all availability zones.

SPot Availability (SPA) Goals

Goals

- Offer a software tool that provides (almost) the same availability as on-demand instances
- Leverage opportunities with AWS spot instances
 - Use the least volatile availability zones
 - Use instances that have better pricing among availability zones

Strategy

- Developed the *Spot Instance MarketPlace Exploitation (SIMPLE)* algorithm to migrate between locations
- Inspect the historical pricing data for patterns in the minimum price change interval

SPA – SIMPLE algorithm

SIMPLE algorithm workflow

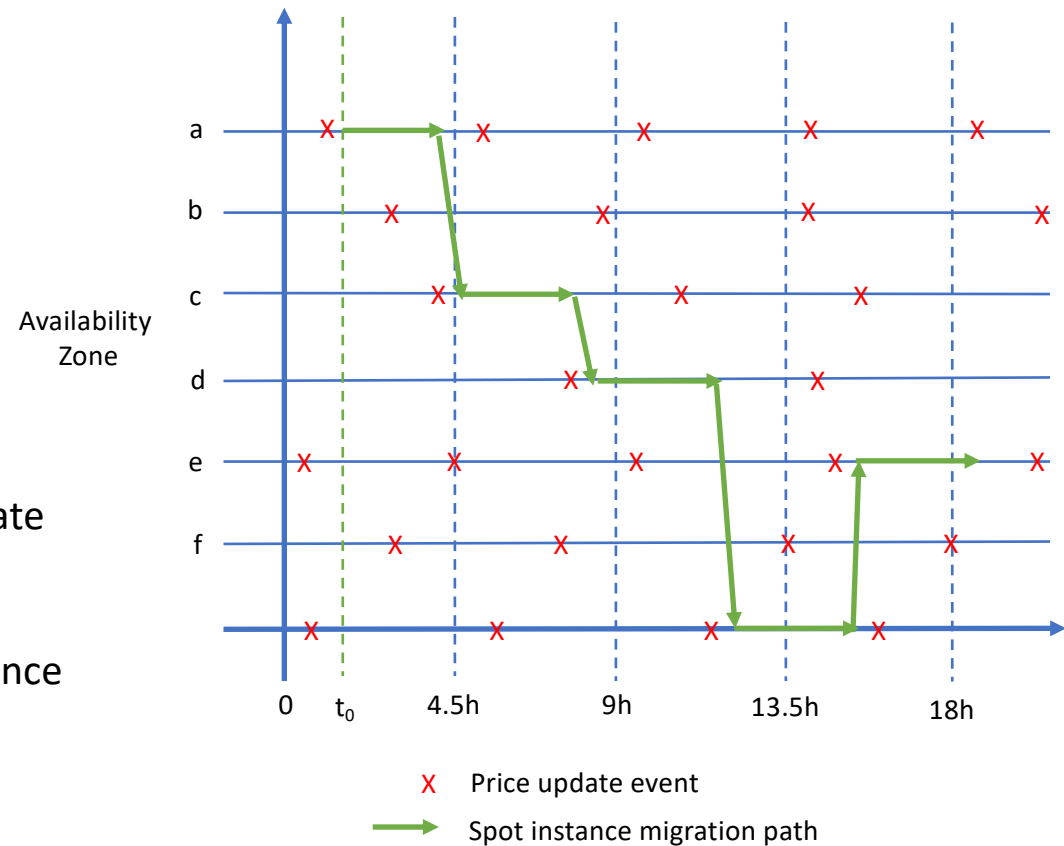
A. Bootstrapping

1. For a given instance, check the price update time among all AZs and pick the latest one
2. Calculate the remaining lifetime and boot the spot instance in that AZ

B. Migration

1. Prior to the end of the spot lifetime*, SIMPLE checks which AZ had the most recent price update for the selected spot instance
2. Starts a new spot instance in that AZ
3. Load balancer redirects requests to new instance
4. Tear down previous instance

* The usual buffer time is 10 minutes.



SPA – Architecture

Parameter Input

- User’s specific criteria, e.g., region to be located

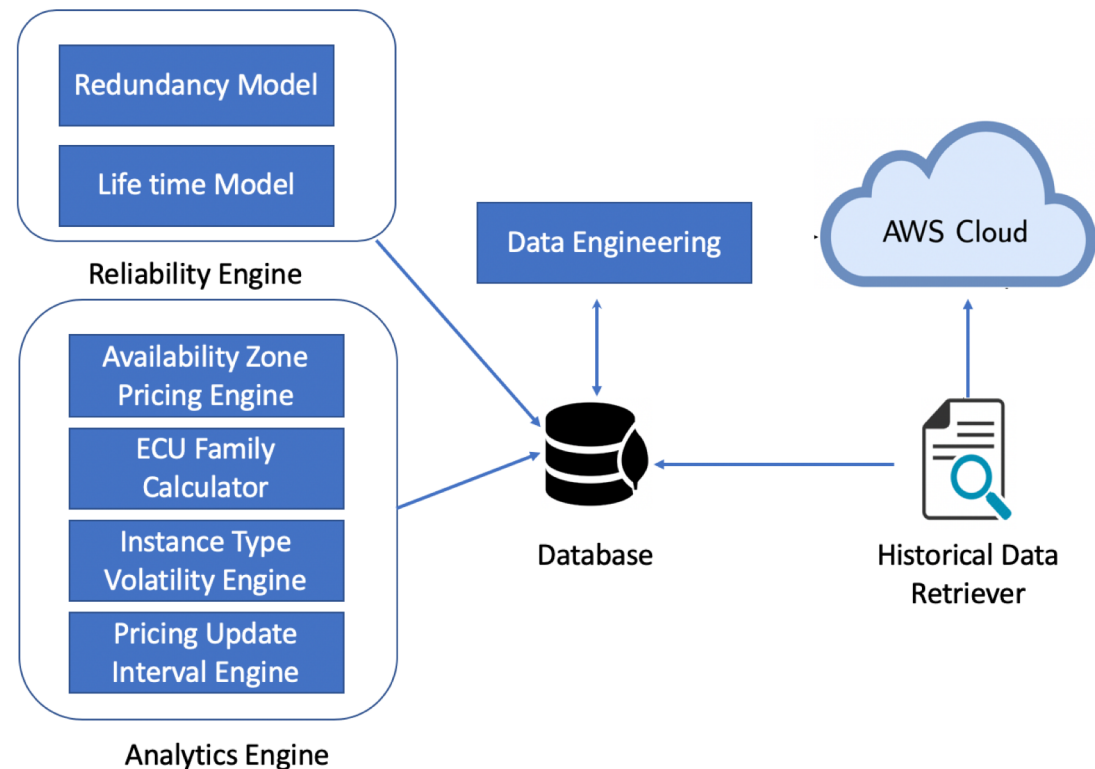
User Input →

Analytics Engine

- Calculates the minimum pricing update interval

Reliability Engine

- Creates a new spot instance, migrates the spot instance from one location to another, and tear down the previous one
- Migrates before any price update occurs, preventing any spot instance eviction



Evaluation Setup

Strategies

- Single spot instance
- Spot instance + SPA
- One and two additional redundant spot instance

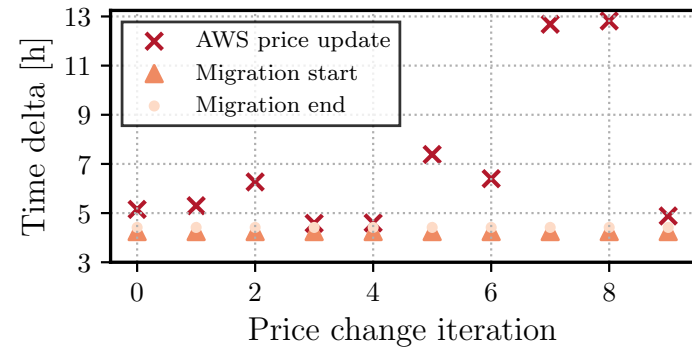
Dataset

- AWS spot pricing data (March to September, 2019)
- Contains 32 availability zones and 252 instance types (~8k combinations)

Evaluation Results

Instance Type	Spot Price	SIMPLE	$r = 1$	$r = 2$
m2.xlarge	10.00%	10.35%	20.00%	30.00%
r3.xlarge	19.49%	20.17%	38.98%	58.47%
m5dn.24xlarge	25.00%	25.88%	50.01%	75.01%
c3.4xlarge	32.36%	33.49%	64.71%	97.07%
a1.large	44.31%	45.86%	88.63%	132.94%
m5a.2xlarge	52.50%	54.34%	105.00%	157.50%
r5dn.xlarge	69.10%	71.52%	138.20%	207.31%

(a)



(b)

Table (a) shows the pricing of different strategies*. SIMPLE algorithm provides a lower cost overhead compared to any strategy that requires redundant spot instances while anticipating any possible spot instance eviction.

Fig. (b) shows the migration process and the price update event. In all cases, SPA was able to finish the migration before the spot instance might be evicted.

* The price shown in the table is the percentage of the on-demand instance, e.g., 10% of the on-demand instance pricing.

Conclusion

- AWS Spot Instances have a minimum price update interval
 - Within this interval, spot instances do not have any evictions, thus, it is safe to run within this time window
- SIMPLE algorithm migrates from one location to another based on least volatile availability zone
 - Anticipates any evictions by migrating every 4.5 hours
 - Overall cost overhead is cheaper than adding any redundant spot instance or on-demand instance
- Experimental results show that SPA can provide the same availability as on-demand instances at a fraction of a cost

Questions?

Walter Wong
University of Helsinki
walter.wong@helsinki.fi

