SpotOn:
A Batch Computing Service for the Spot Market

Supreeth Subramanya, Tian Guo, Prateek Sharma,
David Irwin, Prashant Shenoy

University of Massachusetts Amherst
Cost vs. Availability Tradeoff in IaaS Clouds

Reserved
Guaranteed, Non-revocable

On-demand
Not guaranteed, Non-revocable

Spot
Not guaranteed, Revocable

Cost (per hour)
Cheap
Expensive

Availability

Infrastructure Cloud

Spot Instances
On-demand
Preemptible VM
**Spot Markets**

Amazon EC2

- *Bid* in a 2nd price auction
- *Acquire* when bid > spot price
- *Terminate* when spot price > bid

How do we mitigate the impact of revocation?

1. Raise the bid
2. **Employ fault-tolerance mechanisms**
Amazon EC2 operates ~4000 spot markets

Selecting an instance that yields lowest cost per unit of computation while also considering the probability of revocation is complex.

Scatterplot of ranks for EC2 spot markets
CHALLENGES — APPLICATION COMPLEXITY

Resource Vector

Spot VM

**Cost**
- 20% On-demand

**Revocation Rate**
- 2.4 per day

**Fault-tolerance Mechanism**
- Checkpoint (every 900s)

**CPU : IO**
- 1:1

**Working Set**
- 8GB

**Disk Type**
- Remote

**Running Time**
- 1 hour

**Working Set**
- 8GB

**Disk Type**
- Remote

**Running Time**
- 1 hour

**Resource Vector**

**Cost**
- 20% On-demand

**Revocation Rate**
- 2.4 per day

**Fault-tolerance Mechanism**
- Checkpoint (every 900s)

**CPU : IO**
- 1:1

**Working Set**
- 8GB

**Disk Type**
- Remote

**Running Time**
- 1 hour

**Working Set**
- 8GB

**Disk Type**
- Remote

**Running Time**
- 1 hour

**Completion Time (s)**
- 0
- 2000
- 4000
- 6000
- 8000

**Cost ($)**
- 0
- 25
- 50
- 75
- 100

**On-demand**

**Checkpoint**

Running Time (s) vs. Cost ($): On-demand tasks cost significantly more than checkpoint operations, with completion times varying from 0 to 8000 seconds. The chart illustrates the trade-off between cost and execution time, highlighting the efficiency of checkpoint mechanisms.
CHALLENGES — APPLICATION COMPLEXITY

Resource Vector

Spot VM

CPU : IO
1:1

Working Set
8GB

Disk Type
Local

Running Time
1 hour

Cost
20% On-demand

Revocation Rate
2.4 per day

Fault-tolerance Mechanism
Checkpoint (every 900s)

Spot VM

Challenges — application complexity

- Application complexity
- Local

Graph:
- X-axis: On-demand, Checkpoint, Replicate
- Y-axis: Cost ($), Completion time (s)
- Cost: On-demand 50, Checkpoint 75, Replicate 25
- Completion time: On-demand 0, Checkpoint 8000, Replicate 0
CHALLENGES — APPLICATION COMPLEXITY

CPU : IO | Working Set | Disk Type | Running Time
---------|-------------|-----------|---------------
1:1       | 8GB         | Local     | 1 hour

Resource Vector

Cost | Revocation Rate | Fault-tolerance Mechanism
-----|-----------------|--------------------------
20% On-demand | >24 per day | Checkpoint (every 900s)

Spot VM

---

Completion time (s) | Cost (¢)
-------------------|---------
On-demand | 25 | 75 |
Checkpoint | 50 | 50 |
Replicate | 25 | 25 |
Replicate (Revoked) | 50 | 50 |
**SpotOn: a batch computing service**

Service that accepts batch jobs from users and runs them on spot instances

Manages application and spot market complexity transparently

Run batch jobs at *on-demand performance* but paying *spot market price*
SELECTING THE BEST SPOT MARKET AND FAULT-TOLERANCE MECHANISM

Minimum cost of running \( J_b \) using \( F_t \) on \( S_i \)

Acquire \( S_i \)

Repeat on spot revocation (until \( J_b \) finishes)
Fault Tolerance (1/3)

Reactive Migration

$Z_k \rightarrow$ Random variable measuring time to revocation

$P_k \rightarrow$ Probability job gets revoked before completion

$T_m \propto \frac{\text{size of memory} + \text{local disk}}{\text{remote disk bandwidth}}$

Cost = $\frac{E[\text{Price}_k]}{E[\text{Time}_k]} = \frac{[(1 - P_k) * T + P_k * (E(Z_k) + T_M)] * \text{spot-price}}{(1 - P_k) * T + P_k * E(Z_k)}$
Proactive Checkpoint

Total Overhead = $\frac{T}{\tau} \times T_c + T_L$

Cost overhead is primarily a function of job’s resource usage
Spot Replication

\[ t_0 \]

\[ T_L \sim \text{market volatility} \]

Total Overhead \( \propto \) (Replication factor, \( T_L \))

Cost overhead is primarily a function of market characteristics
SpotOn Prototype

- Built on *Linux Containers* for efficient checkpointing / migration
- *App Emulator* to create synthetic jobs with varying resource usage
Effect of Application and Spot Market Characteristics

Best choice of fault-tolerance mechanism is a function of the spot market and job characteristics.
On Google cluster trace, Cost-aware selection achieved **91.9%** savings with little impact on performance.
Spot markets offer arbitrage opportunities

SpotOn manages application and market complexities

We model fault tolerance and propose a selection algorithm

Prototype on Amazon EC2

Achieves $\sim90\%$ cost savings

Thank you!
BACKUP SLIDES
Bidding does affect volatility but *not drastically*.

SpotOn always bids at the on-demand price.
If spot price goes above on-demand, SpotOn would choose on-demand.

**Spot price distribution has long tail**

**Spot price changes are peaky**