

Amazon EC2 スポットインスタンスにおける入札アルゴリズムの検討

片山太輔¹ 磯部元輝² 小板隆浩²

概要 : Amazon Web Services (AWS) では余剰の計算資源をスポットインスタンスとして安価で提供している。スポット価格は需要の長期的なトレンドに基づいて動的に変化する。したがってスポットインスタンスの継続的な利用は信頼性が低い。本研究ではスポットインスタンスの中断のリスクを低減し、コストを削減する入札アルゴリズムを提案する。

キーワード : Amazon EC2, スポットインスタンス, 入札アルゴリズム

1. はじめに

クラウドを利用するメリットの一つにコストの削減がある。AWS が提供する Amazon Elastic Compute Cloud (Amazon EC2)はクラウド上にサーバを構築するサービスである。Amazon EC2 では余剰の計算資源をスポットインスタンスとして安価で提供している。スポットインスタンスは需要の長期的なトレンドに基づいて動的に変化する。利用者は支払い可能な最大価格である入札価格を設定する。入札価格が現在のスポット価格を上回ればスポットインスタンスが利用できる。需要が高まりスポット価格が入札価格を上回ると利用が中断される。したがって、単純な入札を行った場合、スポットインスタンスの継続的な利用は不安定となる可能性がある。一方、インスタンス間のホッピングを行えば中断率は軽減できることが明らかにされている[1]。しかし、ホッピングでは、データ転送によるオーバーヘッドやコストが大きくなる。またオンデマンド価格を基準にした入札アルゴリズム[2]が提案されているが、スポットインスタンスの価格変動は多様であり、単純に適用することは難しい。本研究では、まずスポットインスタンスを価格変動の特徴によって分類する。分類されたスポットインスタンスについて利用中断のリスクを軽減し、コストを削減する入札アルゴリズムを提案する。提案するアルゴリズムは実際の価格データを用いて評価する。

2. インスタンス分類と提案アルゴリズム

2.1 分類

スポットインスタンスを価格変動の特徴ごとに分類する。スポット価格はオンデマンド価格からの割引率によって設定されるので、本研究ではスポット価格をオンデマンド価格で正規化して評価する。正規化された価格を正規価格と呼び、以下のように計算する。

$$\text{normalized price} = \frac{\text{Spot price}}{\text{On - Demand price}}$$

正規価格の最大値、最小値をそれぞれ normalizedMax , normalizedMin として、変動幅は以下のように計算する。

$$\text{Fluctuation range} = \text{normalizedMax} - \text{normalizedMin}$$

次に、変動率の指標としてヒストリカルボラティリティ (HV) を用いる。HV は過去の一定期間における価格変動率である。サンプル数を n , 1 日の終値を $V_{(t)}, 1 \leq t \leq n-1$, ジョブサイズを s , 1 日の変動率 x_i とすると HV は以下のように定義される。

$$HV = 100 \sqrt{\frac{s}{n-1} \sum_{i=1}^{n-1} (x_i - \bar{x})^2}$$

$$\text{where } x_i = \log \left| \frac{V_{(i)}}{V_{(i-1)}} \right| \text{ and } \bar{x} = \frac{1}{n} \sum_{i=1}^{n-1} x_i$$

求めた HV に対し、指数平滑移動平均(EWMA)を用いる。EWMA は計算時から近い時間の値に重み付けをして計算するため、直近の価格変動を重視した値を得ることができる。 $\lambda \in [0,1]$ を確率変数である忘却因子とすると、EWMA は以下のように計算する。

$$\bar{x}_{n,\lambda} = \frac{1}{\omega_{n,\lambda}} \sum_{i=1}^n \lambda^{n-i} x_i$$

$$\text{where } \omega_{n,\lambda} = \sum_{i=1}^n \lambda^{n-i}$$

先行研究[3]に従い、 λ を 0.5 に設定し EWMA を算出した。

表 1 にこれらの指標からインスタンスを分類した結果を示す。変動幅と変動率の EWMA から 5 つのグループに分類した。図 1 に表 1 の価格変動例の一部を示す。

2.2 入札アルゴリズム

分類したグループごとに入札価格を設定するアルゴリズムを提案する。表 2 に各グループの入札価格を示す。グループ 1 の入札価格は現在のスポット価格であり、グループ 2 の入札価格はオンデマンド価格の 50% である。入札価

¹ 同志社大学理工学部
Faculty of Engineering, Doshisha University
² 同志社大学大学院理工学研究科
Graduate School of Engineering, Doshisha University

表 1 インスタンスの分類

# of group	変動幅	変動率の EWMA	インスタンス数	割合 [%]	例
1	0	N/A	556	56.16	r5dn.24xlarge
2	(0,0.2]	[0,3]	317	32.02	m5ad.24xlarge
3	(0,0.2]	(3,14]	23	2.32	c5a.24xlarge
4	(0.2,1]	[0,3]	27	2.73	p4d.24xlarge
5	(0.2,1]	(3,14]	62	6.26	infl.24xlarge

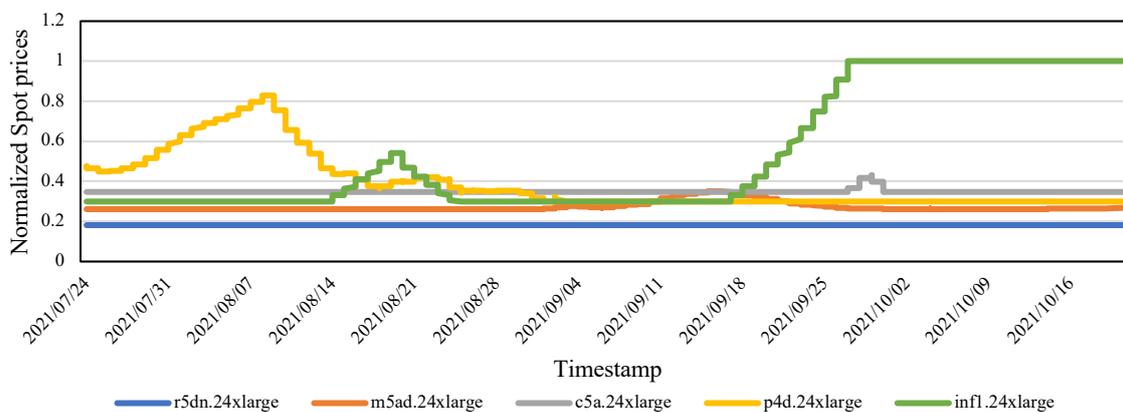


図 1 価格変動例の一部 (2021/7/24-10/21)

表 2 入札アルゴリズム

# of group	入札価格
group1	current Spot Price
group2	50% of On-Demand Price
group3	50% of On-Demand Price
group4	80% of On-Demand Price
group5	On-Demand Price

表 3 実験条件

期間	2021/07/24~2021/10/21
リージョン	ap-northeast-1
アベイラビリティゾーン	a, c, d
OS	Linux/UNIX

格は各グループの *normalizedMax* から算出した。

3. 評価実験

AWS が提供する AWS SDK for Python (Boto3) を用いて取得した実際の価格データで実験を行った。表 3 に実験条件を示す。価格履歴を取得する期間は 2021 年 7 月 24 日から 2021 年 10 月 21 日までの 90 日間、リージョンは東京リージョンである。提案した入札アルゴリズムを表 1 に示した各グループの例に適用して評価実験を行う。

4. 結果と考察

表 4 に実験結果を示す。グループ 1, 2, 3 では 90 日間インスタンスを獲得できた。グループ 4, 5 ではそれぞれ全体の 1.39%, 26.62% の中断が発生した。

グループ 5 のように最大価格がオンデマンド価格に達し、価格変動が激しいインスタンスを低コストで利用するには、ホッピングやオンデマンドインスタンスの利用が有効であると考えられる。

表 4 実験結果

# of group	実行時間 [h]	中断率 [%]
group1	2160	0.00
group2	2160	0.00
group3	2160	0.00
group4	2013	1.39
group5	1585	26.62

また、インスタンスの分類における各指標の閾値による影響も今後検討する必要がある。

参考文献

- [1] Ali Jassim Hasan and Mustafa Hammad, Spot Hopping: Increasing Reliability and Reducing Cost, International Journal of Computing and Digital Systems, Volume 09, Issue 06, pp.1237-1250, 2020.
- [2] V.Khandelwal, A.K.Chaturvedi, and C.P.Gupta, Bidding Strategies for Amazon EC2 Spot Instances – A Comprehensive Review, Proceedings of 4th International Conference on Computing Communication Control and Automation, 2018.
- [3] M.Lumpe, M.B.Chhetri, Q.B.Vo, and R.Kowalczyk, On Estimating Minimum Bids for Amazon EC2 Spot Instances, Proceedings of 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2017.