6J-05

マルチインスタンスGPUを用いた推論ワークロードのクラスタスケジューリング

三井 郁央 †　　杉木 章義 ‡
北海道大学 †　　北海道大学 ‡

概要

近年では科学技術計算などにおいて，GPU などのアクセラレータを使用することが一般的となっている．一方，推論ジョブが GPU のコアを使用しきれない問題が指摘されている．

マルチインスタンス GPU(MIG) は GPU を複数のインスタンスに分割する技術であり，この技術により，複数のジョブを単一の GPU 上で並列に実行することが可能である．

しかし，MIG には分割不可能なインスタンスサイズの組み合わせが存在する．そこで本研究では，複数の推論ジョブの SLO を満たしつつクラスタの使用 GPU 数を最小化する MIG の制約を考慮したスケジューラを作成した．

1　はじめに

マルチインスタンス GPU(MIG)[1] は，2020 年に NVIDIA によって発表された GPU を最大 7 つのインスタンスに分割する新たな物理分割機構である．この技術によって，障害分離性を保ったまま複数のジョブを同一 GPU 内で同時に実行でき，1 つのジョブで 1 つの GPU を使用する場合よりもコストを減らすことが可能である．

しかし，マルチインスタンス GPU においては，分割不可能なインスタンスサイズの組み合わせが存在する. 図 1 は NVIDIA A100 GPU の場合の一例を示したものである．A100 には 7 つの Graphic Processing Clouster(GPC) が存在し，これを分割の最小単位としている．例えば，{1, 2, 4} の組み合わせは許可されているが，{2, 5} や {1, 3, 3} の組み合わせは認められていない．

Cluster scheduling of inference workloads using multi-instance GPUs
† Ayahiro Mitsui, Hokkaido University
‡ Akiyoshi Sugiki, Hokkaido University

分割の可否	GPC0	GPC1	GPC2	GPC3	GPC4	GPC5	GPC6
○	1	1	1	1	1	1	1
○	1	2			4		
×	2		5				
×	1	3			3		

図 1　GPU インスタンスの組み合わせの分割可否の一例 (NVIDIA A100 GPU)

2　提案手法

2.1　前提条件

投入されるジョブは全て推論ジョブであるとする．すべてのジョブにおいて，予め GPU の各インスタンスサイズごとにスループットはすべて測定されているものとする．クラスタは, 使用可能な GPU の個数に関して制限がないものとする．また，クラスタの性能は均一とし，アクセラレータとして NVIDIA A100 GPU のみを持つものとした．

2.2　入力と出力

スケジューラは，(1) 各ジョブの各 GPU インスタンスサイズにおける推論スループットと，(2) 各ジョブの推論スループットに関する SLO を入力とする．

また，出力は GPU インスタンスの分割とジョブの配置とした．

2.3　アルゴリズム

スケジュール完了までの手続きを以下に示す．

Step1 入力に基づき，各ジョブが使用する GPU のインスタンスサイズを決定する．

Step2 ジョブの全組み合わせのうち，以下の条件を全て満たす組み合わせを全て列挙する．

- 使用する各 GPU インスタンス数の合計が GPU の許容量内である．
- MIG の仕様上許される GPU インスタンスサイズの分割パターンである．

Step3 線形ソルバにより GPU インスタンスの分割とジョブの配置を出力する．

2.4 線形計画問題への定式化 (Step3)

全 GPU に配置するすべてのジョブの集合を J とし，各 GPU に配置可能なジョブの組み合わせを列挙した集合を C とする．前節のアルゴリズムの Step2 で列挙したジョブの組み合わせを利用し，新たに行列 $A = \{a_{ij}\}$ を定義する．式 (1) はその定義で意義である． a_{ij} は,i 番目のジョブが j 番目の配置可能なジョブの組み合わせに含まれているとき 1 となり，そうでない時に 0 となるように定める．

$$a_{ij} = \begin{cases} 1 & (\text{if combination}_j \text{ includes job}_i) \\ 0 & (\text{otherwise}) \end{cases} \quad (1)$$

目的関数及び制約について以下に示す．

$$\textbf{minimize} \sum_{j=0}^{M-1} x_j \quad (2)$$

$$\textbf{subject to} \sum_{j=0}^{M-1} a_{ij} x_j \geq 1 \quad (i = 0, 1, \ldots, N-1) \quad (3)$$

$$x_j \in \{0, 1\} \quad (j = 0, 1, \ldots, M-1) \quad (4)$$

式 (2) は使用 GPU 数の最小化を目指すことを示し，式（3）はすべての Job が 1 つ以上配置されることを示す．式 (4) の x_j はジョブを GPU に配置する際の j 番目のジョブの組み合わせを採用する場合に 1，そうでない場合に 0 となる変数である．

3 実験

3.1 実験目標

MIG の制約を考慮したスケジューリングを行うことで，MIG を使用しない場合と比較してクラスタの使用 GPU 数を削減できることを検証する．

3.2 実験条件

各ジョブの SLO は測定したスループットの範囲から正規乱数を用いてランダムに設定した．また，ジョブ数を 100 とし，クラスタで使用する GPU の最大数はジョブ数と等しいものとした．以上の条件のもと，シミュレーションを実施した．

3.3 結果と考察

図 2 は，クラスタが使用した GPU 数についての結果を示している．図中の A100-1/7*7 は GPU を全て最小のインスタンスサイズに分割したクラスタであり，A100-7/7 は GPU の分割を行わなかったクラスタである．また，A100-MIX は各 GPU のインスタンスサイズを 1+2+4 に固定的に分割，または分割を行わなかったクラスタである．

実験の結果，提案手法では分割を行わなかった場合よりも使用 GPU 数を約 53 %削減した．また，全てを最小のインスタンスサイズに分割した場合よりも約 13 %削減した．これは，インスタンスサイズの増加量に対して推論スループットの増加量が大きいジョブが大きく寄与していると考える．このようなジョブでは，最小のインスタンスサイズを複数個占有するよりもより大きいサイズのインスタンスを利用する方が高いスループットになることから，さらに削減できたと考える．

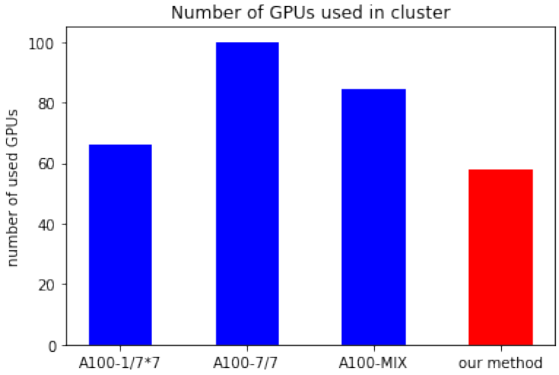

図 2 クラスタにおける GPU 使用数

4 まとめと今後の展望

本研究では，MIG の制約を考慮した推論ジョブのスケジューリングを提案した．A100GPU を均一に分割した場合などと比較して GPU の使用数を削減した．

今後は，ジョブにモデルの更新があった場合の再スケジューリングや，実機での実装を考えたい．

参考文献

[1] NVIDIA Multi-Instance GPU https://docs.nvidia.com/datacenter/tesla/mig-user-guide/

[2] Baolin Li et al., Characterizing multi-instance gpu for machine learning workloads. In IEEE IPDPSW'22, pages 724–731, 2022.

[3] Deepak Narayanan et al., Heterogeneity-Aware cluster scheduling policies for deep learning workloads. In OSDI 20, pages 481–498, 2020.