분산 딥러닝 학습을 위한 GPU 클러스터 스케줄러 연구 동향 분석

고영훈, 신창용, 양경식, 유혁
고려대학교 컴퓨터학과

{yhgo, cyshin, ksyang, chuckyoo}@os.korea.ac.kr

Analysis of GPU Cluster Schedulers for Distributed Training of Deep Learning Models

Younghun Go, Changyong Shin, Gyeongsik Yang, Chuck Yoo
Department of Computer Science and Engineering, Korea University

요 약

최근 다양한 분야에 딥러닝 모델이 적용되면서 GPU 를 활용한 딥러닝 모델의 학습이 활발히 이뤄지고 있으며, 딥러닝 모델 학습을 가속하기 위해 분산 딥러닝 학습이 필수적으로 요구된다. 특히, GPU 자원을 효율적으로 사용하기 위해 데이터센터 내에 멀티-테넌트 GPU 클러스터를 구성하여 사용하며, 이 때 분산 딥러닝 학습의 특성에 최적화된 스케줄러 연구가 활발히 진행되고 있다. 본 논문은 최근까지 제안된 분산 딥러닝 학습을 위한 GPU 클러스터 스케줄러 연구를 상세히 분석하여, 1) 달성 목표, 2) 스케줄러의 특징, 3) 알고리즘, 4) 선점형 유무 등을 비교 분석한다. 더 나아가, 기존 연구가 아직 해결하지 못한 문제인 분산 딥러닝 학습의 선점을 전제로 하는 스케줄링 기법의 병목을 지적하고, 향후 연구 방향을 제안한다.

1. 서 론

최근 GPU 를 활용하여 이미지 분류, 기계번역, 추천 시스템, 자율주행 등 다양한 분야의 딥러닝 모델을 학습시키고 있다. 또한, 학습에 사용하는 GPU 활용률을 개선하기 위해, 데이터센터를 기반으로 하는 멀티-테넌트(multi-tenant) GPU 클러스터가 사용된다. 클러스터에서 구동되는 워크로드들에 대해, 기존에는 빅데이터 처리를 위한 워크로드 스케줄링 기법이 연구되었다.[1] 그러나 딥러닝 학습은 그 고유한 특성(§2)으로, 기존 스케줄링 그대로 적용하기에 적합하지 않다[5]. 또한 다수의 GPU 를 활용하여 학습을 가속화하는 분산 학습(Distributed Training)이 널리 활용되고 있지만[2], 기존 연구[1]는 분산학습에 필요한 GPU 개수 등 분산 딥러닝 학습을 고려하고 있지 않다.

상기 한계를 극복하기 위해, 분산 딥러닝 모델 학습에 최적화된 GPU 클러스터 스케줄링 연구[3-7]가 활발히 진행되고 있다. 그러나 다양한 지표 및 목표를 위해 제안되는 스케줄러 간의 비교는 부재하다. 이에, 본 논문은 딥러닝 학습을 위한 클러스터 스케줄러의 연구 동향을 상세하게 비교 및 분석하고, 이를 기반으로 향후 연구방안을 도출한다.

2. 분산 딥러닝 학습의 특성

일반적인 딥러닝 학습은 데이터셋을 미니배치(mini-batch) 단위로 분할한 뒤, 미니배치를 반복하여 학습한다. 각 학습 과정은 1) 각 미니배치와 모델의 가중치를 연산하여 모델의 예측 결과를 계산하는 순전파(forward propagation), 2) 목적함수를 통해 오차를 계산하고, 그래디언트(gradient)를 산출하기 위해 역방향으로 오차를 전파하는 역전파(backward propagation), 3) 학습률(learning rate)과 그래디언트에 따라 모델의 가중치를 갱신하는 과정으로 구성된다. 상기 과정을 반복하고, 딥러닝 모델의 손실 값이 목표 임계값에 도달하면 학습이 완료된다.

분산 딥러닝은 모델 학습에 다수의 GPU 를 사용하는 학습 방식으로, 학습 과정에서 어떤 컴퓨팅 자원이 주로 사용되는지를 기준으로 크게 네 단계로 구분할 수 있다[5]. 구체적으로, 1) 학습 데이터셋의 로드(스토리지 I/O), 2) 학습을 수행하기 위한 전처리(CPU), 3) 반복적으로 미니배치를 학습(GPU), 4) 다수의 GPU 를 사용하여 학습하는 경우 그래디언트의 동기화 과정(네트워크 I/O)으로 구분된다.

3. 딥러닝 학습을 위한 GPU 클러스터 스케줄러 연구 비교

딥러닝 학습을 위한 GPU 클러스터 스케줄러[3-7]는 각각 다양한 목표 및 핵심 아이디어를 지닌다. 이에, 본 논문은 기존 연구를 알고리즘 및 선점 유무 측면에서 비교 분석한다. 분석 결과는 표 1 에 요약되어 있고, 다음 단락부터 분석 기준별로 순차적으로 논하고자 한다.

3.1 달성 목표

본 논문에서 비교 분석한 연구 모두 학습 성능의 개선을 첫 번째 목표로 고려한다. 구체적으로는, 스케줄링의 효율성을 나타내는 평균 작업 완료시간(JCT)를 개선한다. 이에 더불어, Gandiva[4]와 Antman[6]은 GPU 활용률을 개선하고자 한다. 또한, Antman 은 자원-보장이 필요한 학습을 구분하고, 학습의 GPU 활용률을 일종의 클라우드의 성능 보장 지표(SLA)로 간주하고 달성하는 기법을 제안한다.

3.2 핵심 아이디어

Optimus[3]는 실시간 학습(online learning)을 기반으로 모델 수렴을 예측하여 작업 시간을 알아내고, 이 정보를 토대로 작업완료시간(JCT)을 최소화하는 최적화 문제를 해결한다. 또한 분산 학습의 파라미터 서버와 워커의 배치 방식에 따른 성능 모델을 제안했다. Gandiva[4]는 GPU 활용률을 높이기 위해 딥러닝 학습 작업을 시분할하는

✔:해당 ✘:해당하지 않음

	핵심 아이디어	알고리즘	선점	달성 목표
Optimus[3]	실시간 학습 기반의 모델 수렴 예측 및 파라미터 서버와 워커 수에 따른 성능 모델	✘	✔	평균 작업완료시간(JCT)
Gandiva[4]	스케줄러의 새로운 기초 요소 (suspend-resume, migration, grow-shrink)	✘	✔	GPU 활용률 증가 평균 작업완료시간(JCT) 감소
Tiresias[5]	분산 학습을 위해 GPU 개수를 고려하는 스케줄링 텐서 비대칭(skewness)을 고려한 배치	2D-Gittins Index 2D-LAS	✔	평균 작업완료시간(JCT) 감소
Antman[6]	세분화된 GPU 공유 자원-보장(resource-guarantee) 작업의 SLA 보장과 기회주의적(opportunistic) 작업을 통한 GPU 활용률 개선	✘	✘	GPU 활용률 증가 평균 작업완료시간(JCT) 감소 SLA 보장
Muri[7]	Storage I/O, CPU, GPU, Network I/O 모두 고려하여 인터리빙	SRSF 2D-LAS	✔	평균 작업완료시간(JCT) 감소

표 1. 딥러닝 학습 클러스터 스케줄러 연구 비교

suspend-resume, 학습 중에 GPU 를 교체하는 migration, 학습 중에 GPU 를 추가하고 줄일 수 있는 grow-shrink 기법을 제안했다. Tiresias[5]는 딥러닝 모델의 텐서 비대칭의 정도에 따라서 학습 성능이 다르다는 점을 고려하여 작업을 노드에 배치한다. 또한, 분산 학습에 필요한 GPU 개수와 학습이 진행된 시간을 함께 고려하여 스케줄링한다. Antman[6]은 하나의 GPU 에 다수의 딥러닝 학습 작업을 함께 위치시키는 GPU 공유 방식을 제안했다. 딥러닝 학습 작업을 1) 특정 GPU 활용률을 보장 해야하는 자원-보장 작업과 2) 그렇지 않아도 되는 기회주의 작업으로 구분하며, 자원-보장 작업의 SLA 를 보장함과 동시에 기회주의 작업을 동시 실행한다. 이를 통해, GPU 활용률을 개선했다. Muri[7]는 딥러닝 학습의 단계에 따라 집중적으로 필요한 자원을 스토리지 IO, 네트워크 IO, CPU, GPU 로 구분하고, 작업을 인터리빙(interleaving)한다.

3.3 알고리즘

스케줄러를 설계할 때 작업 실행 순서에 대해 FIFO, SRTF, Gittins index 등의 다양한 알고리즘을 고려할 수 있다. 특히, 분산 딥러닝은 동일한 과정을 반복하는 특성을 갖기 때문에 이를 고려하여 스케줄링 알고리즘을 선택해야 한다.

대표적인 알고리즘인 최단 (잔여) 작업 스케줄링(SJF, SRTF)은 딥러닝 학습에 소요되는 시간을 고려하여 스케줄링 하여 JCT 와 활용률을 개선하지만, 딥러닝 학습의 소요시간을 예측하는 것은 쉽지 않다[5]. Tiresias[5]는 Gittins Index 와 LAS(Least Attained Service)를 변형하여 분산 학습에 필요한 GPU 개수와 현재까지의 작업시간을 고려하는 알고리즘인 2D-Gittins Index 와 2D-LAS 를 제안했다. 구체적으로, 이전 수행했던 학습들의 기록을 통해 작업 수행시간의 분포 정보가 주어진 경우 2D-Gittins index, 주어지지 않았을 경우 2D-Gittins Index 알고리즘을 사용하여 분산 딥러닝 학습을 스케줄링한다.

Muri[7]는 학습에 소요되는 작업시간을 별도 측정하여 소모시간을 계산하고, SRTF(Shortest Remaining Time First)를 변형한 SRSF(Shortest Remaining Service First) 알고리즘을 사용한다. Muri 또한 모델 학습 작업시간 정보를 알 수 없는 경우 2D-LAS 를 활용한다. 다른 알고리즘에 비해 SRSF 가 평균 JCT 성능을 향상시킨다는 점에서 우수하다[5]. 이 외에 Optimus[3], Gandiva[4], Antman[6]은 스케줄링 알고리즘 개발에 초점을 두지 않는다.

3.4 선점형 스케줄링

딥러닝 학습 작업의 선점 유무에 따라 선점형과 비선점형 스케줄링으로 구분할 수 있다. 비선점형 스케줄링의 경우 수행시간이 긴 작업의 HOL(Head-Of-Line) 블로킹으로 인해 수행시간이 짧은 작업이 우선 실행될 기회를 보장받지 못한다. 구체적으로, 비선점형 스케줄링을 사용한 환경에서 사용자는 평균 4102 초의 큐 대기 시간을 겪었음이 보고된 바 있다[3].

따라서 이러한 점을 고려하여 Antman[6]을 제외한 나머지 스케줄러들[3-5, 7]은 대부분 선점 기반의 스케줄링 방식을 채택하고 있다. 그러나 CPU 프로세스와 다르게, 딥러닝 학습 작업의 선점은 모델의 가중치를 저장하고 불러오는 과정이 포함되기 때문에 상당한 병목이 발생한다. 최악의 경우, CNN 기반 이미지분류 모델인 ResNet152 모델의 선점 및 재개의 총 소요 시간이 100 초가량 소요된다[5].

4. 결 론

본 논문에서는 딥러닝 학습을 위한 GPU 클러스터 스케줄러의 연구동향을 비교분석 했다. 딥러닝 학습 작업을 선점하는 경우 상당한 오버헤드가 발생하지만, 현재까지 제안된 기법들은 이를 고려하고 있지 않다. 향후 본 연구진은 선점형 스케줄링의 선점 오버헤드를 고려하여 효과적으로 선점 작업을 구동하는 스케줄링 연구를 수행하고자 한다.

ACKNOWLEDGMENT

이 논문은 2022 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No. NRF-2021R1A6A1A13044830)과 2022 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No. 2015-0-00280, (SW 스타랩) 성능 및 보안 SLA 보장이 가능한 차세대 클라우드 인프라 SW 개발)을 받아 수행된 연구임.

참 고 문 헌

[1] Vavilapalli, Vinod Kumar, et al. "Apache hadoop yarn: Yet another resource negotiator." *Proceedings of the 4th annual Symposium on Cloud Computing*. 2013.

[2] Yang, Gyeongsik, et al. " Prediction of the Resource consumption of Distributed Deep Learning Systems." Abstract Proceedings of the 2022 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, 2022.

[3] Peng, Yanghua, et al. "Optimus: an efficient dynamic resource scheduler for deep learning clusters." *Proceedings of the Thirteenth EuroSys Conference*. 2018.

[4] Xiao, Wencong, et al. "Gandiva: Introspective cluster scheduling for deep learning." *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*.

[5] Gu, Juncheng, et al. "Tiresias: A GPU cluster manager for distributed deep learning." *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*.

[6] Xiao, Wencong, et al. "AntMan: Dynamic Scaling on GPU Clusters for Deep Learning." *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*.

[7] Zhao, Yihao, et al. "Multi-resource interleaving for deep learning training." Proceedings of the ACM SIGCOMM 2022 Conference. 2022.