멀티 클러스터 환경에서 AI 워크로드 GPU/NPU 자원 할당과 운용 비용 최적화

조성민, 안재훈*, 김영환

한국전자기술연구원

{another0306, corehun, yhkim}@keti.re.kr

GPU/NPU Resource Allocation and Operational Cost Optimization for AI Workloads in Multi-Cluster Environments

Sungmin Cho, Jaehoon An*, Younghwan Kim

Intelligent IDC Project Office, Korea Electronics Technology Institute

요 약

최근 초거대 인공지능 모델의 등장과 함께 클라우드 인프라 비용이 폭증하고 있다. 실제로 크라우드 AI 워크로드에 대한 지출은 2025년까지 연 평균 35%를 증가하여 360억 달러에 이를 것으로 전망된다. 이에 따른 학습, 추론, 서빙 등 다양한 AI 워크로드의 특성에 맞추어 GPU/NPU 등 가속기 자원을 효율적으로 배치함으로써 클라우드 운용비용의 최적화하려는 요구가 높아지고 있다. 본 논문에서는 쿠버네티스(Kubernetes) 기반 멀티 클러스터 환경에서 AI 워크로드의 비용 효율적인 자원 할당 전략을 제안한다. 제안하는 시스템 구조는 중앙 Control Plane을 통해 이기종 GPI/NPU 자원을 통합 관리하며, 워크로드별 스케줄링 정책을 적용하여 AI 가속기 자원 활용도를 극대화하고 유휴 자원을 최소화한다. 이를 통해 대규모 AI 모델 서비스의 비용 부담을 완화하고, 클라우드 환경에서 AI 모델을 효율적으로 운영이 가능할 것으로 기대된다.

Ⅰ. 서론

인공지능 모델의 규모와 활용이 폭발적으로 증가하면서 클라우드 인프라 비용이 기업과 연구기관에 큰 부담이 되고 있다. 특히 최신 딥러닝 모델의 학습과 추론에는 대용량 GPU 등 전용 가속기 자원이 필수적이며, 이러한 자원은 매우 고가로서 클라우드 비용의 상당 부분을 차지한다.

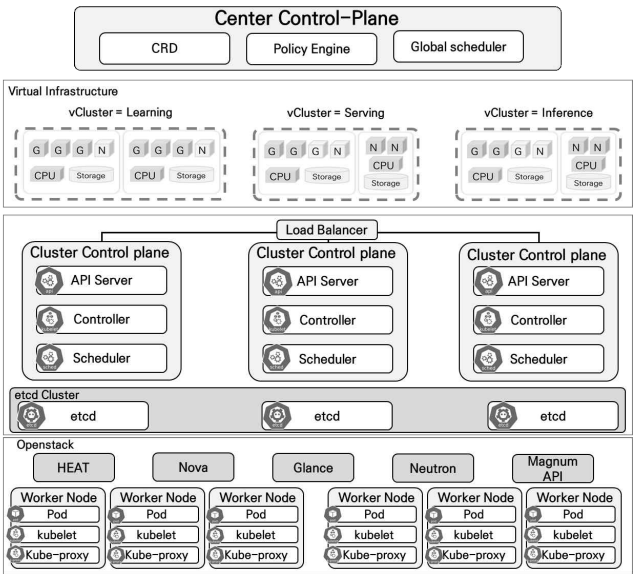
다만 대다수 조직은 가속기 집약 워크로드의 시간적·공간적 변동성을 정확히 예측하기 어렵고, 이를 보수적으로 처리하기 위해 과잉 할당(over-provisioning)과 단편화(fragmentation)를 초래한다[1]. 단일 클러스터 내부에서도 모델 병렬화 제약, 노드 간 상호연결(PCIe/NVLink) 토폴로지, 데이터 지역성 등으로 인해 사용가능한 자원과 실제로 배치 가능한 자원 사이에 괴리가 발생하며, 멀티 클러스터로 확장될수록 가용성·비용·지연시간의 상충 관계가 더 커진다[2].

본 논문에서는 이러한 문제를 해결하기 위해 멀티 클러스터 환경에서의 가상 클러스터·가상 인프라 구성과 C3-ATS(Cost-Carbon-Constraint Accelerator-aware Two-level Scheduler) 알고리즘을 제안한다. 제안 시스템은 다양한 클라우드 제공자가 이기종 가속가 자원을 통합된 풀로 관리하면서, 테넌트별 격리와 민첩성을 보장한다. C3-ATS 알고리즘은 비용·탄소·SLA 제약을 동시에 고려하는 이중 단계 스케줄링을 통해, 전역 클러스터 간 최적화와 로컬(노드) 최적화를 연계하여 운용 비용을 최소화한다[3].

Ⅱ. 본론

1. 운용 비용 최적화를 위한 멀티 클러스터 아키텍처 설계

본 연구에서 제안하는 멀티 클러스터 환경의 가상인프라는 네가지 핵심 설계 목표를 달성하기 위해 계층화된 아키텍처를 채택한다. 첫째 클러스터의 이기종 가속기를 포함한 모든 가용 자원을 통합 풀로 관리하여 워크로드 특성별 비용을 최소화한다. 둘째, 테넌트, 팀, 서비스별로 독립적인 제어면을 제공하면서도 물리 자원은 공유하여 유휴 자원을 최소화한다. 셋째, GPU, NPU 등 다양한 가속기의 특성 차이를 통합 API를 통해 인터페이스로 제공한다. 넷째, 전역 클러스터 간과 로컬(노드) 레벨에서 서로 다른 목적함수와 제약을 처리하는 이중 단계 스케줄링을 구현한다.



[그림 1] 멀티 클러스터 가상 인프라 아키텍처

그림1은 워커 노드와 컨트롤 플레인을 분리하고, API 서버는 L4/L7 로드 밸런서 뒤에서 노출한다. etcd는 전용 클러스터로 분리해 컨트롤 플레인을 보호하며, 과반수 합의로 일관성을 보장한다.

중앙 Control Plane은 멀티 클러스터 차원의 정책·카탈로그·스케줄링 결정을 담당하며, 각 지역에 배치된 고가용 컨트롤 플레인과 gRPC/REST/CRD 동기화로 연동된다. 중앙 평면은 상태 저장을 자체 스토리지(예: RDB/메시지 버스)로 관리하되, 워크로드의 상태는 각 클러스터별 외부 etcd

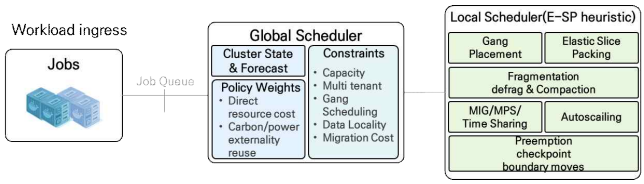
에 보관한다. 따라서 중앙 Control Plane 장애 시에도 각 클러스터는 독립적으로 안정 동작하며, 중앙 Control Plane이 복구되면 조정자(reconciler)가 최종 상태를 재수렴한다.

가상 인프라 계층은 노드와 가속기 가상화를 담당하는 핵심 구성요소다. vNode 풀(vNode Pool)은 실제 노드 풀을 가상 노드 클래스로 추상화한다. 가속기 가상화는 하드웨어 분할(MIG)과 소프트웨어 공유(MPS), 그리고 시분할 방식을 병행해 구현하며, 각 가상 가속기는 큐 기반 또는 슬라이스 기반의 단위로 제공된다.

자원 모델링 측면에서 각 가속기 프로파일은 타입, 메모리 용량(GB), 연산 성능(TOPS), 대역폭, 지원 데이터 타입, 공유 가능 여부를 포함한다. 비용 모델은 클러스터와 시간대별로 산정한 단위 자원 비용률(원화/시간)을 사용한다. 필요에 따라 클러스터별 탄소 강도와 전력 지표를 비용 항에 가중치로 포함해 환경 비용을 함께 고려한다[4].

2. C3-ATS: 비용-탄소-제약 인지형 이중 단계 가속기 스케줄러

C3-ATS(Cost-Carbon-Constraint Accelerator-aware Two-level Scheduler)는 멀티 클러스터 환경에서 비용, 탄소 배출, 그리고 서비스 수준 합의(SLA) 제약을 동시에 고려해 작업을 배치하는 이중 단계 스케줄링 알고리즘이다. 시간에 따라 도착하는 다양한 작업을 멤버 클러스터와 그 하위의 노드·가속기 자원에 적절히 배치하는 문제를 다룬다.



[그림 2] C3-ATS 스케줄러 동작방식

그림 2는 스케줄러의 전반적인 동작 방식을 나타낸다. 목표는 주어진 기간 동안의 총 운용비용을 최소화하는 것이다. 여기에는 자원 사용에 따른 직접 비용, SLA 위반으로 인한 패널티, 전력·탄소에 기인한 간접 비용이 포함되며, 반대로 모델 가중치의 재사용과 데이터 지역성으로 얻는 절감 효과는 비용에서 공제된다.

전역 스케줄링에서는 시간에 따른 가격과 수요의 변동은 자연스럽게 반영되며, 최적화는 다음 제약을 모두 만족하는 범위에서 수행된다.

(i) 노드와 가속기의 슬라이스·메모리·대역폭 등 용량 제약,

(ii) 테넌트 격리와 우선순위 기반의 공정성,

(iii) 학습 작업의 동시 배치(gang scheduling) 요구,

(iv) 모델·데이터의 지역성,

(v) 인스턴스 이동 시 발생하는 마이그레이션 비용.

가중치는 온라인 관측에 기반해 점진적으로 조정된다. 혼잡이 발생한 클러스터에는 가격을 높여 실효 단가를 증가시키고, 그 결과 신규 작업이 다른 클러스터로 분산되도록 유도한다. SLO 위반이 관측되면 지연 민감도를 강화하고, 모델·데이터 재사용이 잦을수록 지역성의 중요도를 높이는 식으로 정책이 조정된다.

로컬 스케줄링 단계에서는 Elastic Slice Packing(E-SP) 휴리스틱을 사용해 노드·가속기 단에서의 실제 배치를 최적화한다. 학습 작업은 통신 오버헤드를 줄이기 위해 가능하면 동일 노드의 연속 자원에 우선 배치하고, 배치 추론 작업은 MIG, MPS, 시분할 등 공유 메커니즘을 적극 활용해 빈 슬롯을 촘촘히 채운다[5]. 온라인 서빙 작업은 목표 지연을 보장하기 위해, 큐 길이와 지연 신호에 기반한 자동 확장으로 워밍 인스턴스를 유지한다.

자원 모델 측면에서 각 가속기는 메모리, 연산, 대역폭의 다차원 용량을 가진다. 가상화된 슬라이스 단위는 이들 차원을 일정 비율 점유하며, 작업은 각 차원에 대한 요구량을 명시한다. 서빙 워크로드는 추가로 요청률과 목표 지연을 포함한다.

E-SP는 우선순위가 높은 작업과 적합한 슬라이스를 먼저 짝지어 파편화가 최소화되도록 배치한다. 일정 주기마다 조각모음(defragmentation)과 압축(compaction)을 수행해 잘게 나뉜 슬라이스를 병합하고, 필요시 가장 영향이 작은 인스턴스를 선별적으로 이동시켜 큰 연속 슬라이스를 확보한다. 지연이 목표치를 초과할 조짐이 관측되면 버스트를 흡수하도록 슬라이스를 일시 증설하고, 부하가 안정되면 단계적으로 축소한다.

마이그레이션은 다음 원칙을 따른다. 첫째, 기대 절감액이 사전에 정한 임계치를 넘을 때만 이동한다. 둘째, 서빙 워크로드는 대체 인스턴스를 먼저 웜업해 무중단 전환을 보장한다. 셋째, 학습 워크로드는 체크포인트 경계에서만 이동을 허용해 진행 중인 학습에 미치는 영향을 최소화한다.

Ⅲ. 결론

본 논문에서는 멀티 클러스터 환경에서 이기종 가속기 자원을 효율적으로 관리하기 위한 가상 인프라 아키텍처와 C3-ATS(Cost-Carbon-Constraint Accelerator-aware Two-level Scheduler) 알고리즘을 제안하였다. 제안된 시스템은 계층화된 아키텍처를 통해 GPU/NPU 등 다양한 자원을 통합 관리하면서도 테넌트별 격리와 독립성을 보장한다. C3-ATS는 전역 단계에서 비용, 탄소 배출, SLA를 종합적으로 고려한 클러스터 선택과 로컬 단계에서 워크로드의 최적화된 자원 할당을 수행하는 이중 단계 접근법을 채택하였다. 본 연구는 AI/ML 워크로드가 급증하는 현 시점에서 비용 효율성과 지속 가능성을 동시에 달성할 수 있는 실용적인 솔루션을 제시하며, 클라우드 환경에서의 자원 관리에 새로운 방향을 제시한다. 향후 강화학습 기반 적응형 최적화와 엣지-클라우드 연속체로의 확장을 통해 더욱 발전된 시스템으로 진화할 것으로 기대된다.

ACKNOWLEDGMENT

이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 결과임 (RS-2025-02293869, AI 반도체를 활용한 클라우드 플랫폼 구축 및 최적화 기술 개발)

참 고 문 헌

[1] Wencong Xiao, Romil Bhardwaj, et al. “Gandiva: Introspective Cluster Scheduling for Deep Learning.” OSDI 2018. USENIX Association, 2018.

[2] Juncheng Gu, Mosharaf Chowdhury, Kang G. Shin. “Tiresias: A GPU Cluster Manager for Distributed Deep Learning.” NSDI 2019. USENIX Association, 2019.

[3] Peifeng Yu, Mosharaf Chowdhury. “Salus: Fine-Grained GPU Sharing Primitives for Deep Learning Applications.” MLSys 2020. 2020.

[4] Jiaqi You, Yubo Li, et al. “Zeus: Understanding and Optimizing GPU Energy for DNN Training.” NSDI 2023. USENIX Association, 2023.

[5] Zhisheng Ye, Wei Gao, Qinghao Hu, Peng Sun, Xiaolin Wang, Yingwei Luo, Tianwei Zhang, and Yonggang Wen. “Deep Learning Workload Scheduling in GPU Datacenters: A Survey.” ACM Computing Surveys 56(6), Article 146, 2024. ACM, 2024.