연산 스토리지 기반 전처리 오프로딩을 적용한 AI 컨테이너 오케스트레이션 아키텍처 설계에 관한 연구

길주현, 안재훈*, 김영환

한국전자기술연구원 지능형IDC 사업단

wngus0331@keti.re.kr, *corehun@keti.re.kr, yhkim@keti.re.kr

A Study on the Design of an AI Container Orchestration Architecture with Computational Storage-based Preprocessing Offloading

Juhyun Kil, Jea-Hoon An*, Younghwan Kim

Intelligent IDC Project Office, Korea Electronics Technology Institute

요 약

AI 학습 및 추론 워크로드의 데이터셋 규모 증가로 GPU 메모리 부족, 스토리지 I/O 병목, 운용 비용 증가 문제가 심화되고 있다. 특히 기존 Kubermetes 및 Kubeflow 기반 오케스트레이션은 시간에 따라 변하는 자원 사용량과 대규모 전처리의 네트워크 부하를 충분히 고려하지 못한다. 본 연구에서는 Computational Storage Device(CSD) 기반 인스토리지 전처리 오프로딩을 통합한 AI 컨테이너 오케스트레이션 아키텍처를 제안한다. 제안 구조는 자원 예측, 분석 기반 스케줄링과 CSD 연동 전처리 실행을 통해 AI 파이프라인의 효율성을 높인다. 이를 통해 자원 활용 최적화, 전처리 지연 감소, 비용 절감이 가능하며 대규모 데이터 처리화 학습 작업에서의 성능 향상을 기대할 수 있다.

Ⅰ. 서 론

최근 인공지능(AI) 학습 및 추론 파이프라인은 데이터셋 규모와 복잡성의 급격한 증가로 인해 막대한 컴퓨팅 자원과 효율적인 데이터 전처리 기술을 필요로 한다. 특히, GPU 메모리 부족과 스토리지 입출력(I/O) 병목 현상, 그리고 대규모 데이터 전송에 따른 높은 네트워크 부하는 AI 워크로드의 처리 속도와 비용 효율성에 심각한 제약으로 작용한다. 예를 들어, 시뮬레이션 및 이미지 처리와 같은 순수·응용과학 및 생명공학 분야, 대형 AI 모델 학습을 수행하는 AI 서비스 기업 등의 환경에서는 이러한 성능 저하와 스케일링 한계가 연구 및 서비스 품질 저하로 직결되고 있다. [1]

현재 널리 사용되는 Kubernetes나 Kubeflow 같은 컨테이너 오케스트레이션 플랫폼은 확장성과 자동화된 자원 관리를 지원하지만, 동적인 자원 수요 변동성과 GPU 공유 스케줄링, 대규모 전처리 과정에서의 네트워크 및 스토리지 I/O 병목 문제를 충분히 반영하지 못한다. [2] 이로 인해 데이터 전처리 단계에서 시스템 병목 현상이 발생하여, 전체 AI 파이프라인의 처리 지연과 자원 낭비가 발생하고 있다. 또한 기존 스케줄러들은 자원 사용 예측 및 정책 기반 최적화 기능이 미흡하여, AI 컨테이너의 효율적인 생명주기 관리에 한계를 보이고 있다. [3]

이러한 한계를 극복하고자 본 연구는 Computational Storage Device(CSD) 기반 인스토리지 전처리 오프로딩을 통합한 AI 컨테이너 오케스트레이션 아키텍처를 제안한다. 아키텍처는 마스터 노드의 자원분석 엔진, 자원 사용량 예측 엔진, 오케스트레이션 정책 최적화 엔진, 클러스터 오케스트레이터로 구성된다. 전처리 작업은 Kubernetes의 CRD(Custom Resource Definition)을 통해 정의되며, 워커 노드의 전처리 매니저가 이를 CSD 내부 도커 런타임 기반 전처리 모듈을 통해 실행된다. 이 방식은 전처리를 스토리지 내에서 수행해 네트워크 부하와 I/O 지연을 줄이며, 산업·연구 환경 전반에서의 비용 절감을 실현할 수 있다. [4]

Ⅱ. 본론

1. 제안하는 오케스트레이션 아키텍처

기존 시스템은 동적인 자원 수요 변동, GPU 메모리 공유 한계, 대규모 전처리 시 발생하는 네트워크·스토리지 병목으로 인한 AI 워크로드 처리 효율이 저하되는 한계가 있다. 이에 자원 분석·예측 기반 스케줄링과 CSD 기반 인스토리지 전처리 오프로딩을 통합한 AI 컨테이너 오케스트레이션 아키텍처를 정의하였다.

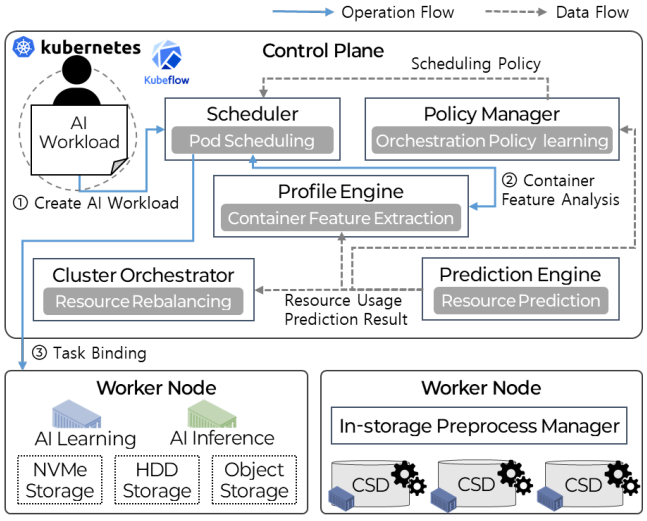

그림 1 AI 컨테이너 오케스트레이션 아키텍처

제안 아키텍처는 Kubernetes 컨테이너 오케스트레이터 위에 Kubeflow 환경이 구축된 클러스터 상에 구현되며, 클러스터는 컨트롤 플레인과 다수의 워커 노드로 구성된다. 컨트롤 플레인에는 클러스터 관리

* 교신저자

와 컨테이너 오케스트레이션을 담당하는 핵심 컴포넌트들이 위치하며, 워커노드에서는 전처리·학습·추론 등 다양한 AI 워크로드가 실행된다.

AI 컨테이너 스케줄링 과정은 다음과 같다. 사용자가 AI 작업을 수행하는 컨테이너를 정의해 요청하면 Custom Scheduler가 스케줄링이 필요한 작업을 감지한다. Custom Scheduler는 GPU 메모리 공유 스케줄링을 지원하여 자원을 효율적으로 활용하며, Profile Engine이 AI 작업 특성과 입력 데이터 특성을 분석하여 자원 요구량을 산출하고 이를 기반으로 배치 결정을 수행한다. 스케줄링 정책은 Policy Manager가 클러스터 상태에 맞추어 학습하고 최적화하며, 워커 노드에 배치된 AI 작업은 실행 중에도 시간에 따른 자원 사용량 변화가 모니터링된다.

Cluster Orchestrator는 클러스터 내 전반적인 리소스를 조정하며, 실행 중인 AI 워크로드를 포함해 노드 간 자원 활용 균형을 유지한다. 또한 Prediction Engine은 클러스터 내 모든 작업의 향후 자원 사용량을 예측하여 Cluster Orcestrator·Policy Engine·Policy Manager 등에 결과를 제공함으로써 자원 최적화를 지원한다.

2. CSD(Computational Storage Device)

CSD는 저장장치에 내장된 프로세서를 통해 데이터를 저장하는 동시에, 저장장치 내부에서 직접 데이터 처리가 가능한 차세대 스토리지 기술이다. 기존 NVMe SSD와 유사하게 서버에 장착되지만, 자체 CPU 및 운영체제가 내장되어 있어 사용자가 데이터 이동 없이 저장장치 내에서 AI 전처리, 머신러닝 인퍼런스 등의 작업을 직접 실행할 수 있다. [5] 본 논문에서는 이러한 CSD 내부에 도커 컨테이너 런타임 환경을 구비하여, 데이터 전처리 작업을 컨테이너화된 형태로 저장장치 안에서 직접 수행하도록 설계하였다. 이는 전처리 단계에서 네트워크 부하와 데이터 이동 지연을 최소화함으로써 전체 AI 워크로드의 성능을 향상하고, 자원 활용의 효율성을 극대화하는 데 기여한다.

3. 전처리 오프로딩 아키텍처

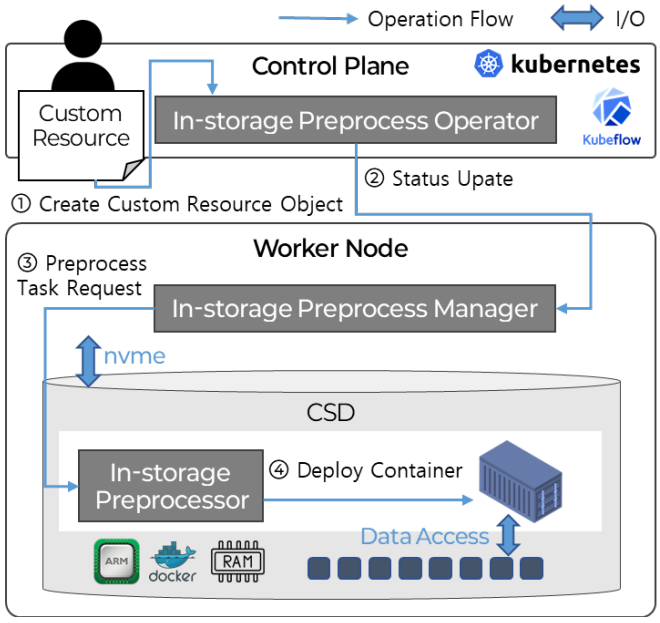

그림 2 전처리 컨테이너 오프로딩 아키텍처

AI 컨테이너의 전처리 오프로딩 작업은 Kubernetes의 CRD(Custom Resource Definition)로 정의 및 등록된다. 이 CRD는 In-stroage Preprocess Operator가 관리하며, CSD 환경에 맞춘 스케줄링 및 배치를 지원한다. CRD에는 작업 메타정보, 컨테이너 이미지, 데이터 위치, 스케줄링 힌트 등이 포함된다.

워크플로우는 먼저 사용자가 오프로딩할 전처리 작업을 CRD 오브젝트로 정의하는 것으로 시작된다. Control Plane의 In-storage Preprocess Operator는 이를 감지하고, CSD가 탑재된 워커 노드 중 작업 수행에 적합한 노드를 선정하여 작업 요청을 전달한다. 워커 노드 내 In-storage Preprocess Manager는 컨테이너 실행 정보를 수신하고 이를 CSD로 전달한다. CSD 내부의 In-storage Preprocessor는 Docker 컨테이너 런타임을 기반으로 전처리 컨테이너를 실행한다.

이 과정은 저장장치 내부 연산을 활용해 데이터 이동 및 네트워크 부하를 최소화하며, 입출력 지연을 줄여 자원 활용도와 전체 AI 워크로드 처리 성능을 최적화한다.

Ⅲ. 결론

본 논문에서는 기존 AI 컨테이너 오케스트레이션 환경에서 나타나는 동적 자원 수요 변동, GPU 메모리 공유의 한계, 대규모 전처리 단계의 네트워크 및 스토리지 병목 문제를 해결하기 위한 자원 분석·예측 기반 스케줄링과 CSD 기반 인스토리지 전처리 오프로딩을 통합한 아키텍처를 제안하였다. 제안 아키텍처는 Kubernets/Kubeflow 기반 클러스터에서 Profile Engine, Prediction Engine, Policy Manager, Custom Scheduler 등을 통해 최적의 자원 배치 결정을 수행하고, 전처리 작업을 CRD를 통해 CSD 내부에서 실행함으로써 데이터 이동을 최소화하였다.

이를 통해 네트워크 부하 및 스토리지 I/O 지연 감소, GPU 및 CPU 자원의 효율적 활용, AI 파이프라인 전반의 처리 지연 최소화, 운영 비용 절감 등의 효과를 기대할 수 있다. 또한 제안 구조는 기상 예측, 바이오 데이터 분석, 스마트 팩토리, 클라우드 AI 서비스, 자율주행 등 다양한 산업 및 연구 분야에 적용 가능하다.

향후 연구에서는 실제 시스템 구현 및 대규모 워크로드에 대한 실험을 통해 제안 아키텍처의 성능을 정량적으로 검증하고, 자원 예측 모델의 정확도 향상과 오케스트레이션 정책 최적화를 위한 강화학습 기반 기법을 추가로 적용할 계획이다.

ACKNOWLEDGMENT

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 결과임 (No.RS-2024-00461572, AI 연산 가속기 최적화 고효율 병렬 스토리지 SW 기술 개발)

참 고 문 헌

[1] Rodriguez, R. & Buyya, R., Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions, arXiv, 2021.

[2] Kaul, D., AI-Driven Self-Healing Container Orchestration Framework for Energy-Efficient Kubernetes Clusters, Emerging Science Research, 2024.

[3] Rothman, A. et al., Reinforcement Learning Based Kubernetes Scheduler for Efficient Resource Utilization, arXiv, 2025.

[4] Smith, J. et al., Resource-aware multi-task offloading architectures for edge AI, ScienceDirect, 2023.

[5] NGD Systems, Newport In-Situ Development Platform – Hardware Product Brief, Apr. 2019.