AI 기반의 HPC 환경을 위한 I/O 분석 모니터링 기법 연구

윤준원[1], 홍태영[1]
[1]한국과학기술정보연구원
jwyoon@kisti.re.kr, tyhong@kisti.re.kr

A Study on I/O Monitoring Methods for AI-Based HPC Systems

Junweon Yoon[1], Taeyeong Hong[1]
[1]Div. of Supercomputing Infrastructure Center, KISTI

요 약

고성능컴퓨팅(HPC) 시스템은 전통적인 과학·공학 시뮬레이션뿐 아니라, GPU 가속기를 활용한 대규모 인공지능(AI) 학습 및 추론 워크로드까지 지원하는 다목적 플랫폼으로 진화하고 있다. 이러한 워크로드는 방대한 양의 데이터셋과 빈번한 입출력(I/O) 연산을 수반하기 때문에, 병렬파일시스템과 스토리지 계층은 심각한 병목현상이 발생한다. 특히 AI 워크로드는 작은 파일 접근, 높은 메타데이터 접근 빈도, 반복적인 데이터 적재 패턴을 보이는 반면, 기존 HPC 워크로드는 대규모 순차 I/O가 중심이므로, 공존 환경에서는 I/O 자원 간섭과 성능 저하가 빈번하게 발생한다. 따라서 작업 레벨, 노드 레벨, GPU 활용도, 애플리케이션 계층별 I/O 패턴을 통합적으로 분석할 수 있는 모니터링 기법의 필요성이 증대되고 있다.

본 연구는 Darshan과 같은 애플리케이션 I/O 프로파일링 도구, Slurm 기반 job-level 로깅, GPU 사용량 모니터링, 노드 레벨 I/O 측정 등 기존 접근법을 비교 분석하며, AI 기반 워크로드에서의 I/O 모니터링 요구사항과 이를 충족하기 위한 연구 방향을 제시한다. 또한 실제 병렬연산에 대한 I/O 프로파일링을 수행하여 그 특성을 분석한다.

1. 서론

고성능컴퓨팅(HPC) 환경을 대표하는 슈퍼컴퓨터는 수천에서 수만대의 계산노드, 병렬파일시스템 그리고 이를 연결하는 고속의 인터커넥트로 구성되며 대규모 과학적 시뮬레이션부터 최근의 인공지능(AI) 학습·추론에 이르기까지 다양한 워크로드를 처리한다[1]. AI 워크로드는 GPU 가속기, 대규모 데이터셋, 반복 학습(epoch)을 필요로 하며 데이터 로딩 단계에서 스토리지 I/O 성능이 전체 처리율을 제한하는 주요 병목이 된다[2][3]. 반면 기존 HPC 워크로드는 주로 대규모 순차 입출력에 특징을 갖는다. 서로 다른 패턴의 두 워크로드가 동일한 스토리지 계층을 공유할 때, I/O 혼잡과 QoS 저하 문제가 주요 이슈로 보고되고 있다[4]. 따라서, 효율적인 HPC 운영을 위해서는 정밀한 I/O 모니터링 및 분석이 필수다. 이를 위해서는 (1)애플리케이션 수준의 I/O 패턴 이해, (2)작업 단위의 리소스 사용량 기록, (3)노드·GPU 단위의 정밀 리소스 모니터링, (4)시스템 전반의 혼잡 탐지 및 예측이 필요하다. 그러나 기존의 모니터링 체계는 전통적인 HPC 애플리케이션에 최적화되어 AI 워크로드의 특수성을 반영하는 데에는 한계가 있다. 따라서 AI 기반 HPC 환경에 적합한 새로운 I/O 분석 및 모니터링 기법에 대한 연구가 필요하다[5][6].

본 연구는 AI 기반 HPC 환경에서 I/O 분석 및 모니터링 기법을 검토하고, I/O가 발생하는 여러 단계(Application, Job, Node, GPU)에서 통합하여 분석하는 환경을 제시하고자 한다. 또한 HPC 환경에 I/O 프로파일링 도구를 설치하여 모니터링 환경을 구축하고 병렬연산 어플리케이션을 수행하여 I/O특성을 분석했다.

2. AI 기반의 HPC 환경 I/O 분석

전통적 HPC 환경과 AI 학습, 추론 워크로드의 상반된 I/O 패턴이 동일한 병렬 파일시스템 자원에서 공존할 경우, 자원 간섭(interference)이 심화되며 시스템 전반의 성능 저하로 이어진다[7]. 따라서 GPU 활용률을 나타내는 usage 지표와 I/O 성능 계측 결

과를 결합하여, 연산 병목과 데이터 병목을 구분하고 성능 저하의 원인을 정확히 진단하는 것이 필수적이다[8].

이러한 복합적 환경에서는 계측 계층의 다양화가 필요하다. 애플리케이션 계층에서는 Darshan과 같은 경량 I/O 프로파일러를 통해 API 호출 기반의 I/O 추적이 가능하며[9], Job-level 계층에서는 Slurm job accounting과 Lustre jobid_var, LMT 등을 활용하여 작업별 read/write throughput 및 metadata ops를 수집할 수 있다[10]. 노드 및 시스템 계층에서는 /proc, cgroup I/O 통계, Lustre 클라이언트 및 OST RPC 카운터 등을 통해 자원 사용량과 병목 현상을 관찰할 수 있다. 또한 GPU 계층에서는 NVML 라이브러리, eBPF 기반 계측, CUDA trace 도구를 활용하여 GPU 내부 및 스토리지 간 데이터 경로를 정밀하게 추적할 수 있다.

AI 기반 HPC 환경에서 I/O 분석 및 모니터링은 다양한 접근을 통해 발전해왔다. 본 절에서는 애플리케이션, 작업, 노드(CPU, GPU) 계층에 따라 다양한 분석 방법을 제시한다.

1) Darshan 애플리케이션 I/O 프로파일링

Argonne National Lab(ANL)에서 개발한 HPC 표준 I/O 프로파일러로 POSIX, MPI-IO, HDF5 호출을 샘플링하여 애플리케이션 실행 동안의 I/O 패턴을 수집할 수 있으며 AI 데이터 로더의 I/O 비효율성 진단이 가능하다 또한 경량 오버헤드로 대규모 시스템에서도 활용 가능, 연구·운영 모두에서 사실상 표준 도구로 활용되고 있다.

2) Job-level I/O 모니터링

SLURM JobID를 Lustre I/O RPC에 태깅하여 job별 I/O를 구분할 수 있다. Lustre Monitoring Tool (LMT): OSS/MDT 계층에서 job별 read/write throughput, metadata IOPS 추출하여 job-level I/O 계측 기반의 QoS 분석이 가능하며 이를 통해Job-aware scheduling 및 I/O throttling 연구의 기반 데이터로 활용 가능하다.

3) Node-level 및 시스템 계측

eBPF 기반 모니터링은 커널 tracepoint를 활용해 lustre_*, vfs_read/write 호출을 잡 단위로 분류할 수 있으며 노드 자원(CPU, 메모리, I/O, 네트워크)을 동시에 수집하여 잡별 I/O 패턴과 시스템 혼잡 상황의 연계를 분석 가능하다.

4) GPU usage 및 AI 워크로드 모니터링

NVIDIA에서 제공하는 NVML 기반 GPU usage를 수집할 수 있으며 GPU utilization, memory throughput, PCIe 사용량 측정이 가능하다 또한 CUDA/Nsight 계측: GPU 메모리 복사 지연, 데이터 로딩 overhead 분석할 수 있다.

5) 통합 모니터링 및 성능 예측

DLIO (Deep Learning I/O Benchmark)는 ANL에서 2021년부터 개발해 공개한 AI 워크로드 특화 I/O 벤치마크 도구로 HPC 벤치마크(IOzone, IOR, mdtest 등)가 순차적 대용량 I/O 패턴을 측정하는 데 적합했다면, DLIO는 딥러닝 학습/추론 워크로드의 데이터 로딩 패턴을 모사하는 데 초점을 둔다. 데이터 포맷(TFRecord, WebDataset 등), prefetch 설정, batch size 변화에 따른 I/O 성능 영향 측정 가능하도록 설계되었다.

3. I/O 프로파일링 도구 적용 및 분석

대규모 HPC 환경에서 I/O 분석을 위해 KISTI 슈퍼컴퓨터 5호기 누리온에 Darshan 프로파일러를 설치하고 I/O를 분석했다. 누리온 8,400여대의 계산노드는 구성되어 초병렬시스템으로 분산메모리 환경에서 MPI를 통해 병렬 연산과 I/O가 수행된다. 또한 Lustre 기반의 스토리지가 적용되어 있어 Darshan, Cerebro, Lustre Monitoring Tool(LMT), DB를 동시에 구성하여 분석환경을 구축하였다. LMT는 Lustre 서버(MDS, OSS)에서 I/O 트래픽, RPC 수, bandwidth 등을 수집하며 jobid_var=PBS_JOB_ID 옵션을 Lustre 클라이언트 마운트 시 활성화하여 각 job의 I/O가 식별 가능하도록 구성했다. Cerebro는 Darshan, LMT 데이터를 통합 분석하는 I/O telemetry 프레임워크이다.

<표 1>과 같이 Darshan = LD_PRELOAD로 실행하여 mpi 병렬작업이 수행되면 I/O 매트릭들을 실시간으로 수집하고 .darshan 로그 생성한다. 이후 darshan에서 제공되는 parser/summary 분석을 통해 access size, throughput, 메타데이터, rank 균등성, 시간 분포를 분석할수 있다.

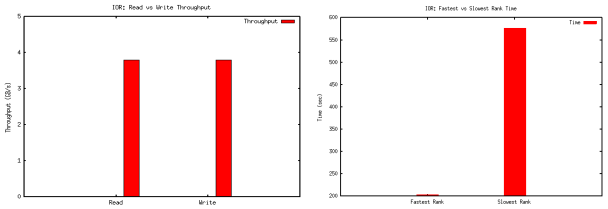
<표 1> Darshan 프로파일링 수행 절차

```
# 실행 환경에 라이브러리 preload:
export LD_PRELOAD=/opt/darshan/lib/libdarshan.so
# 애플리케이션 실행 시 계측
mpirun -n 256 ./mpi_app
# 로그 파싱 & 리포트 생성
darshan-parser jobid.darshan > jobid_17959204.txt
# 요약 리포트:
darshan-job-summary.pl jobid.darshan
```

<표 2>는 실제 어플리케이션 수행을 통해 I/O 프로파일링 결과를 얻을 수 있다. 누리온에 적용되어 있는 PBS 스케줄러에 Lustre 클라이언트를 연동하여 작업별 I/O를 수집할 수 있음을 보여준다. 수행 작업은 MPI-IO를 통한 대규모 순차 스트리밍 I/O 패턴(2MB 블록, 총 2.75TB R/W, 256 프로세스 균등 분산)을 분석하는 어플리케이션으로 (1)throughput 산출, (2)균등성 및 rank 편차 확인, (3)쓰기/읽기 성능 차이, (4)메타데이터 시간 영향, (5)OST-level 부하 분산 등을 확인할 수 있다.

<표 2> Darshan을 통한 I/O 프로파일링

```
# darshan log version: 3.21
# compression method: ZLIB
# exe: /lustre/IOR_mbcm/ior_4.1.0/bin/ior -a MPIIO -b 10g -o /lustre/IOR_mbcm/ior_4.1.0/IOR_data/iorData_M17959204.pbs -t 2m -d 5 -w -r -i 1 -C -Q 25 -k
# uid: 1016
# jobid: 17959204
# start_time: 1747242052
# start_time_asci: Thu May 15 02:00:52 2025
# end_time: 1747242777
# end_time_asci: Thu May 15 02:12:57 2025
# nprocs: 256
# run time: 726
# metadata: lib_ver = 3.2.1
# metadata: h = romio_no_indep_rw=true;cb_nodes=4
```



<그림 1> Darshan 프로파일링 (Throughput, Rank I/O Time)

수행된 어플리케이션은 약 2.75 TB 쓰기와 읽기를 726초 동안 수행(약 3.8 GB/s)되었으며 <그림 1>에서 보는것과 같이 성능 편차는 fastest/slowest rank의 시간 차이(F_FASTEST_RANK_TIME 204s vs F_SLOWEST_RANK_TIME 576s)로 일부 노드에서 I/O 지연 발생됨을 볼수 있다. GPU I/O가 수행되는 어플리케이션에서는 GPU/AI 워크로드 연계(현재는 IOR synthetic test이지만) AI workload라면, 이 throughput이 GPU feeding 속도를 충족하는지 검증할 수 있다.

4. 결론

AI 기반 HPC 환경은 어플리케이션의 다양성과 워크로드의 이질성이 크게 증가하여 다계층 I/O 모니터링은 운영적·연구적 측면에서 의의가 크다. 본 연구는 이런 환경의 여러 계층(Darshan, job-level, node-level, GPU usage)에서 발생하는 I/O 분석 및 모니터링 기법을 검토하고 통합 분석할 수 있는 방향성을 제시하였다. 향후, 다양한 딥러닝 프레임워크(PyTorch, TensorFlow, JAX) 및 실제 과학 데이터셋을 반영한 I/O 벤치마킹을 수행해 모니터링 기법의 일반성을 검증해야 한다. 또한, GPUDirect Storage(GDS)와 같은 기술의 확산으로 GPU 메모리 접근 방식에 대한 워크로드 분석, GPU usage 로그와 Lustre RPC 로그를 연계하여 GPU idle 현상과 I/O 지연의 직접적 인과관계 추가 분석이 필요하다.

이런 연구를 통해 운영자는 JOB 스케줄링 정책을 개선하고 QoS 보장이나 I/O 스로틀링 기법을 적용하여 시스템 안정성을 높일 수 있다. 또한 연구자는 AI 워크로드 특성을 반영한 데이터 로딩 및 캐싱 전략을 설계할 수 있으며, 장기적으로는 수집된 데이터를 활용하여 AI 기반의 이상 탐지나 I/O 성능 예측 모델을 개발할 수 있다.

감사의 글(Acknowledgment)

이 논문은 2025년도 한국과학기술정보연구원의 기본사업(과제명:국가 플래그십 초고성능컴퓨팅 인프라 구축 및 서비스, 과제번호:K25L2M2C2)으로 수행된 연구입니다.

참고문헌

[1] Chien, Steven WD, et al. "Characterizing deep-learning I/O workloads in TensorFlow." 2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS). IEEE, 2018.

[2] Tessler, Chen, , et al. “Reinforcement Learning for Datacenter Congestion Control.” arXiv preprint arXiv:2102.09337, 2021.

[3] Du, Jiangsu, et al. “SAIH: A Scalable Evaluation Methodology for Understanding AI Performance Trend on HPC Systems.” arXiv preprint arXiv:2212.03410, 2022.

[4] Byna, S., Oral, S., Lockwood, G., et al. “Parallel I/O, HPC, and Data-Intensive Science: Five Research Challenges.” Future Generation Computer Systems, 2020.

[5] Cui, Shengkun, et al. “Characterizing GPU Resilience and Impact on AI/HPC Systems.” arXiv preprint arXiv:2503.11901, 2025.
[6] LBNL (Lawrence Berkeley National Laboratory). “AIIO: Using Artificial Intelligence for Job-Level and Automatic I/O Performance Bottleneck Diagnosis.” Presented at HPDC ’23, 2023.
[7] Yildiz, Orcun, et al. "On the root causes of cross-application I/O interference in HPC storage systems.", IPDPS. IEEE, 2016.
[8] Sencan, Efe, et al. "Analyzing GPU Utilization in HPC Workloads: Insights from Large-Scale Systems." Practice and Experience in Advanced Research Computing, The Power of Collaboration. pp. 1-8, 2025.
[9] P. Carns et al., “Understanding and Improving Computational Science I/O Through Darshan,” Cluster, 2011.
[10] Whamcloud. "Lustre Monitoring Tool(LMT) Documentation." Available: https://wiki.lustre.org, 2023.