동적 배칭과 동시성 제어를 통한 서버리스 환경 AI 추론 최적화 연구

김태림[1], 김예진[2], 김건민[3], 김경백[3]

전남대학교 인공지능학부[1], 전남대학교 소프트웨어공학과[2], 전남대학교 인공지능융합학과[3]

ktr0706@jnu.ac.kr, ye031010@jnu.ac.kr, geonminkim@jnu.ac.kr, kyungbaekkim@jnu.ac.kr

AI Inference Optimization in Serverless Environments through Dynamic Batching and Concurrency Control

Taerim Kim[1], Yejin Kim[2], Geonmin Kim[3], Kyungbaek Kim[3]

[1]Dept. of Artificial Intelligence, Chonnam National Univ.,
[2]Dept. of Software Engineering, Chonnam National Univ.,
[3]Dept. of AI Convergence, Chonnam National Univ.

요 약

서버리스 환경은 불규칙한 요청 패턴과 짧은 응답 시간이 중요한 AI 추론 환경에서 주목받고 있으나 자원 경쟁과 콜드스타트로 인한 지연 증가 문제가 발생한다. 따라서 본 연구는 서버리스 환경에서 동적 배칭 기법과 프로비저닝 동시성, 웜업 전략이 AI 추론 지연과 처리량에 미치는 영향을 분석하였다. AWS Lambda 기반 MNIST 모델을 대상으로 버스티 및 균일 워크로드 환경에서 평균, p50, p99 지연과 처리량을 측정한 결과, 적절한 동시성과 웜업은 지연을 최소화하면서 처리량을 동시에 향상시킬 수 있음이 확인되었다.

Ⅰ. 서 론

서버리스 컴퓨팅은 개발자가 인프라를 직접 관리하지 않고 애플리케이션을 실행할 수 있도록 지원하는 클라우드 네이티브 모델로 불규칙한 요청 패턴과 짧은 응답 시간이 중요한 AI 추론 환경에서 주목받고 있다. 그러나 대규모 모델 추론이나 버스티 워크로드에서는 콜드스타트와 자원 경쟁으로 인해 지연이 크게 증가할 수 있다. 본 연구는 서버리스 환경에서 동적 배칭과 프로비저닝 동시성이 AI 추론 지연에 미치는 영향을 체계적으로 분석하고, 다양한 워크로드 조건에서 평균 지연(ms), p50 / p99 지연(ms)을 최소화할 수 있는 최적 구성 탐색을 목표로 한다. 특히 균일 및 버스티 요청 패턴 하에서 배치 큐 임계값, 큐 타임아웃, 배치 크기, 동시성 수준 등의 조합이 지연에 미치는 영향을 실험적으로 검증함으로써 서버리스 AI 추론 서비스에서 지연 최적화를 위한 설계 지침을 제공한다. 이를 통해 단순 처리량 중심이 아닌 지연 중심 성능 평가와 최적화 방안을 제시하며 실험적으로 검증한다.

Ⅱ. 관련 연구

기존 연구에서는 동적 배칭 기법을 활용하여 처리량과 비용 효율을 개선하기 위해 동적 요청 관리와 배칭 기법이 활용된 사례가 보고되었다[1]. BATCH[2]와 같은 서버리스 ML 추론 플랫폼에서는 적응형 동적 배칭을 통해 처리량을 향상시키고 지연을 줄이는 방법이 제시되었다. ServerlessLoRA[3] 연구에서는 서버리스 환경에서 LLM 추론 시 동적 배칭과 함수 컨테이너 재사용을 통해 비용과 지연을 최소화하는 접근법을 다루었으나 일반적인 서버리스 환경에서 동적 배칭과 동시성 제어가 지연에 미치는 영향을 통합적으로 분석한 연구는 여전히 부족하다.

Ⅲ. 실험 방법

본 연구에서는 서버리스 환경에서 동적 배칭과 프로비저닝 동시성이 AI 추론 지연에 미치는 영향을 분석하였다. 균일 및 버스티 요청 패턴을 대상으로 배치 큐 임계값, 큐 타임아웃, 배치 크기, 동시성 수준 등의 다양한 파라미터 조합을 실험하였다. 각 설정 별로 평균 지연(ms), p50 / p99 지연(ms), 처리량을 측정하고, 워크로드 유형별 최적 설정 범위를 도출하였다. 그리드 서치 기반 파라미터 튜닝을 통해 배치 및 큐 관련 설정을 탐색하고, 프로비저닝된 동시성과 웜업 전략을 함께 적용하여 동적 배칭과 요청 급증 상황에서 지연 최소화 처리량을 동시에 달성할 수 있는 설정을 체계적으로 검증하였다.

3.1. 그리드 서치 기반 파라미터 튜닝

그리드 서치를 통해 균일 및 버스티 워크로드에 대한 동적 배칭 파라미터를 탐색하였다. 균일 워크로드에서는 큐 임계값과 타임아웃을 후보로 설정하고 모든 조합을 테스트하였으며 버스티 워크로드에서는 여기에 버스트 크기와 버스트 간 휴식 시간을 추가하여 총 네 가지 파라미터 조합을 탐색하였다. 실험 결과, 균일 워크로드에서는 threshold=2와 timeout=0.3 조합이, 버스티 워크로드에서는 threshold=2, timeout=0.5, burst_size=10, burst_pause=2조합이 각각 평균 지연과 처리량 면에서 가장 효율적인 것으로 나타났다.

3.2. 동시성 설정 및 웜업 전략

본 연구에서는 프로비저닝된 동시성과 웜업 전략을 함께 적용하였다. 프로비저닝된 동시성은 특정수의 인스턴스를 항상 사전에 기동시켜 두는 방식으로 워크로드 변화에 따라 즉각적으로 요청을 처리할 수 있도록 한다. 반면, 동시성을 과도하게 설정하면 불필요한 자원 상주로 인해 비용 효율성이 저하될 수 있어 성능과 비용 간 균형이 필요하다.

웜업 전략은 일정 수의 인스턴스를 주기적으로 활성화하여 갑작스러운 요청 급증 시에도 콜드스타트가 최소화되도록 하는 방법이다. 그러나 웜업 빈도와 동시성 조합이 적절하지 않을 경우 오히려 자원 경합을 유발할

배치 방식	워크로드	평균 / p50 / p99지연 (ms)	처리량 (images/sec)
동적 배칭	균일	126.9 / 122.5 / 185.2	6.11
동적 배칭	버스티	125.1 / 121.1 / 167.5	6.15
동적 배칭(프로비저닝동시성(3)+웜업(3))	균일	129.7 / 119.5 / 281.9	6.06
동적 배칭(프로비저닝동시성(4)+웜업(3))	균일	280.8 / 121.8 / 311.8	4.15
동적 배칭(프로비저닝동시성(3)+웜업(4))	균일	137.3 / 124.1 / 347.8	5.92
동적 배칭(프로비저닝동시성(4)+웜업(4))	균일	126.7 / 121.8 / 179.9	6.11
동적 배칭(프로비저닝동시성(5)+웜업(5))	버스티	140.5 / 134.5 / 264.1	5.87
동적 배칭(프로비저닝동시성(6)+웜업(5))	버스티	**121.3 / 118.5 / 164.1**	**6.22**
동적 배칭(프로비저닝동시성(7)+웜업(5))	버스티	123.4 / 119.6 / 166.6	6.18
동적 배칭(프로비저닝동시성(5)+웜업(6))	버스티	278.3 / 267.1 / 917.0	4.18
동적 배칭(프로비저닝동시성(6)+웜업(6))	버스티	124.7 / 120.2 / 169.5	6.15
동적 배칭(프로비저닝동시성(7)+웜업(6))	버스티	274.0 / 232.1 / 2254.7	4.22

(표 1) 프로비저닝 동시성 및 웜업을 적용한 동적 배칭의 워크로드별 지연 시간 및 처리량 비교

수 있음을 시사한다. 이러한 결과는 동시성과 지연 시간 간의 밀접한 상관관계를 보여준다. 즉, 동시성이 부족하면 요청이 대기 큐에 쌓여 p99 지연이 증가하고, 동시성이 충분히 확보되면 전반적인 지연이 감소한다.

Ⅳ. 실험 환경

본 연구는 서버리스 환경에서 동적 배칭이 AI 추론 지연에 미치는 영향을 평가하기 위해 AWS Lambda와 Docker 이미지로 배포된 MNIST 분류 모델을 사용하였다. Lambda 함수는 1024MB 메모리와 60초 타임아웃으로 설정하였다. 동적 배칭은 큐 임계값과 큐 타임아웃을 기반으로 구현되었으며 균일 및 버스티 요청 워크로드에 대해 최적 파라미터를 탐색하였다. 실험에서는 프로비저닝 동시성 수준과 웜업 횟수를 조합하여 다양한 시나리오를 평가하였다. 각 설정에서 평균 지연(ms), p50 / p99 지연(ms), 처리량(images/sec)을 측정하여 기록하였다. 이를 통해 동적 배칭과 동시성/웜업 조합이 지연과 처리량에 미치는 영향을 분석하였다.

V. 실험 결과

(표 1)에 제시된 것과 같이 프로비저닝 동시성과 웜업 전략을 적용한 경우 동시성 수준과 웜업 횟수에 따라 지연과 처리량이 변화하는 양상이 나타났다. 버스티 워크로드에서 동시성 5와 웜업 5 조합 대비 동시성 6과 웜업 5 조합에서는 평균 지연이 19.2ms 감소했고, 처리량은 0.35 images/sec 증가하였다. 적절한 동시성 확보가 p99 지연과 처리량 개선에 효과적임을 확인할 수 있었다. 그러나 동시성과 웜업 횟수가 과도하게 증가한 동시성 7과 웜업 6 조합에서는 평균 지연이 274.0ms, p99 지연이 2254.7ms로 급격히 증가하고 처리량이 4.22 images/sec로 감소하였다. 이는 과도한 동시성과 잦은 웜업 호출이 Lambda 인스턴스 간 자원 경합을 유발하여 지연을 증가시키는 결과임을 시사한다. 반면, 균일 워크로드에서는 동시성 4와 웜업 4 조합에서도 평균 지연과 처리량이 각각 126.7ms와 6.11 images/sec로 큰 차이가 없었다. 이는 균일 요청 패턴에서는 과도한 동시성 확보가 필요하지 않으며 워크로드 유형에 따라 동시성과 웜업 전략을 적절히 조정해야 함을 보여준다.

VI. 결론

본 연구에서는 서버리스 환경에서 동적 배칭과 프로비저닝 동시성, 웜업 전략이 AI 추론 지연 및 처리량에 미치는 영향을 분석하였다. AWS Lambda와 Docker 기반 MNIST 분류 모델을 활용한 실험 결과, 프로비저닝 동시성과 웜업 전략을 적절히 조합할 경우 버스티 워크로드에서 평균 지연(ms), p50 / p99 지연(ms)을 효과적으로 줄이면서 처리량을 최적화할 수 있었다. 이는 서버리스 AI 추론 환경에서 동적 배칭과 프로비저닝 동시성, 웜업 전략을 적절히 조합하면 지연을 최소화하면서 처리량을 높일 수 있음을 보여준다. 향후 연구에서는 보다 대규모 모델과 다양한 AI 추론 워크로드로 실험을 확장하고, 실시간 변화에 대응하는 적응형 동적 배칭 알고리즘과 비용 효율을 고려한 동시성 조정 기법 연구가 요구된다.

ACKNOWLEDGMENT

본 연구는 2025년도 과학기술정보통신부 및 정보통신기획평가원의 소프트웨어중심대학사업의 연구결과로 수행되었습니다.(2021-0-01409)(34%).본 연구성과는 2025년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(RS-2025-25398164)(33%).본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2025-RS-2023-00256629)(33%).

참 고 문 헌

[1] N. Mahmoudi and H. Khazaei, "MLProxy: SLA-Aware Reverse Proxy for Machine Learning Inference Serving on Serverless Computing Platforms," Proc. 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2022), Pittsburgh, PA, USA, Feb. 2022.

[2] A. Ali, R. Pinciroli, F. Yan, and E. Smirni, "BATCH: Machine Learning Inference Serving on Serverless Platforms with Adaptive Batching," in Proc. Int. Conf. High Performance Comput., Networking, Storage and Analysis (SC), Nov. 2020.

[3] Y. Sui, H. Wang, H. Yu, Y. Hu, J. Li, and H. Wang, "ServerlessLoRA: Minimizing latency and cost in serverless inference for LoRA-based LLMs," Proc. ACM Symposium on Cloud Computing (SoCC 2024), Santa Cruz, CA, USA, May. 2025.