LLM 서빙 시스템에서 RWT Estimator 정확도 향상을 위한 온라인 파라미터 적응 알고리즘

황용수○, 장성, 김기현, 김영재†
서강대학교 컴퓨터공학과

Online Parameter Adaptation Algorithm for RWT Estimator Accuracy in LLM Serving Systems

Yongsoo Hwang, Seong Jang, Kihyun Kim, Youngjae Kim†
Department of Computer Science and Engineering, Sogang University

요 약

LLM 서빙 시스템에서 스케줄링 품질은 RWT Estimator가 제공하는 요청 그룹 완료시간 추정값의 정확도에 의존한다. 기존 QLM은 유효 디코드 시간(d_eff)과 평균 출력 토큰 수(μo)를 오프라인 프로파일링으로 측정한 고정값으로 사용하나, 실제 운용 환경에서는 연속 처리 과정에서 발생하는 런타임 상태 변화에 따른 디코드 속도 저하와 그룹별 출력 분포 편차로 인해 추정 정확도가 저하된다. 본 논문은 이러한 두 가지 오차 원인을 probe 실험으로 실증하고, 각 파라미터를 런타임에 동적으로 보정하는 온라인 파라미터 적응 알고리즘을 제안한다. 실험 결과(Qwen2.5-1.5B-Instruct, RTX 2080 SUPER), 정적 추정기(Static Estimator) 대비 hat_T MAPE가 28.2%에서 16.9%로 11.3pp 감소하였으며, d_eff 기여분 -2.5pp, μo 기여분 –8.8pp를 분리 정량화하였다. 추가 실험에서는 Dynamic Estimator 적용 시 SLO 달성률이 54.1%에서 71.8%로 17.7pp 향상됨을 실증하였다.

1. 서 론

OpenAI GPT-4, Google Gemini 등 대규모 언어 모델(LLM)은 챗봇, 코딩 보조, 문서 처리 등 다양한 분야에서 광범위하게 활용되고 있으며, 이에 따라 LLM 서빙 시스템의 효율적 설계가 핵심 과제로 부상하였다.

LLM 서빙 환경에서는 응답 지연 기한(SLO)이 엄격한 인터랙티브 요청과 여유가 큰 배치 요청이 혼합 처리되며, QLM[1]은 요청 대기시간 추정기(RWT Estimator)로 각 그룹의 완료 시점을 사전에 추정하고 이를 기반으로 스케줄링 결정을 내린다. 그러나 추정에 사용되는 파라미터를 오프라인 고정값으로 사용하기 때문에, 연속 처리 중 나타나는 디코드 속도 저하와 그룹별 출력 토큰 편차로 인해 추정 정확도가 저하된다.

본 논문은 probe 실험으로 두 가지 오차 원인을 실증하고, 각 파라미터를 런타임에 동적으로 보정하는 온라인 파라미터 적응 알고리즘을 제안한다. 실험 결과 완료시간 추정 오차가 28.2%에서 16.9%로 감소하였으며, 추가 실험에서 SLO 달성률이 54.1%에서 71.8%로 17.7pp 향상됨을 확인하였다.

2. 연구 배경

2.1 QLM 개요

QLM[1]은 인터랙티브 요청과 배치 요청이 혼합되는 LLM 서빙 환경에서 SLO 달성을 목표로 하는 큐 관리 시스템으로, 요청을 그룹화하고 RWT Estimator가 예측한 그룹 완료 시간을 바탕으로 글로벌 스케줄러가 Eviction, Load Balancing 등의 LSO(LLM Serving Operations)를 결정한다. 따라서 RWT Estimator의 정확도는 스케줄링 품질에 직접적인 영향을 미친다.

2.2 QLM 및 RWT Estimator 한계 분석

QLM의 RWT Estimator는 n개 요청으로 구성된 그룹의 완료 시간을 아래와 같이 추정한다[1].

$$T(n) = (n-1)\cdot\frac{\mu o}{\theta} + P + maxO\cdot\varepsilon\cdot d$$

여기서 d는 토큰당 디코드 시간, ε는 비효율 계수, $\theta=1/(d\times\varepsilon)$는 토큰 처리량, P는 prefill 시간, μo는 평균 출력 토큰 수, $maxO$는 최대 출력 토큰 수를 나타낸다. 본 논문은 QLM 전체 스케줄링 정책의 재현보다 RWT Estimator의 그룹 완료시간 추정 정확도에 초점을 두며, 실험의 SLO는 요청 그룹의 완료 deadline으로 정의한다. QLM은 vLLM[2]을 기반으로 구현되며, 위 파라미터들을 실험 전 오프라인 프로파일링으로 측정하여 고정값으로 사용한다. 이 중 μo는 요청 그룹별로 피팅되나 런타임에는 갱신되지 않는다. 이 추정기는 중심극한정리(CLT)에 근거해 대규모 큐 환경에서는 R^2=0.99의 높은 정확도를 보인다[1]. 그러나 소규모 워크로드에서는 CLT 적용 조건이 충족되지 않으며, 완료시간을 과대 추정하면 불필요한 Eviction이 발생할 수 있고, 과소 추정하면 SLO 위반 위험이 커질 수 있다. 따라서 아래에서는 d_eff의 런타임 변동성과 μo의 분포

본 연구는 2026년 과학기술정보통신부 및 정보통신기획평가원의 AI중심대학사업 지원을 받아 수행되었음(2026-0-00036).

† Corresponding Author

편차가 완료시간 추정 정확도에 미치는 영향을 분석한다.

2.3 오차 원인 분석 — d_eff 가변성

동일 프롬프트를 사용한 요청 30건을 연속 처리한 probe 실험에서 요청 순서에 따른 실측 디코드 시간(d_obs)의 단조 증가를 확인하였다(그림 1). 초기 10건의 평균 d_obs는 0.00881 s/tok이었으나, 후반 10건의 평균은 0.01008 s/tok으로 증가하였다. 이는 연속 처리 과정에서 발생하는 KV cache 사용량, 메모리 관리 오버헤드, 스케줄링 상태 변화 등 런타임 요인이 유효 디코드 시간에 영향을 줄 수 있음을 시사한다. 다만 본 실험은 현상 관찰에 초점을 두었으며, 개별 요인의 기여도 분석은 향후 과제로 남긴다. 따라서 d_eff는 연속 처리 과정에서 일정하지 않은 값을 가지며, 실제 시스템 상태에 따라 변할 수 있음을 확인하였다. 그러나 기존 RWT Estimator는 d_eff를 고정값으로 가정하므로, 연속 처리 중 발생하는 성능 변화를 추정에 반영하지 못한다.

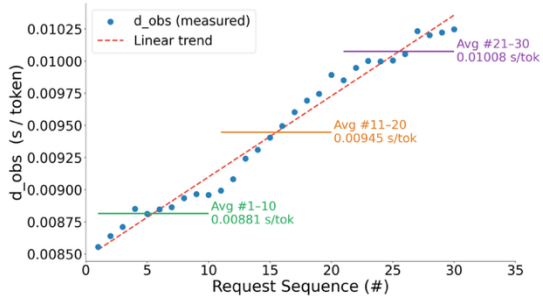

그림 1. 연속 처리 시 d_obs(s/tok) 단조 증가

2.4 오차 원인 분석 — μo 편차

μo는 RWT Estimator에서 그룹별 평균 출력 토큰 수를 나타내며, 사전 설정된 μo는 실제 출력 분포를 충분히 대표해야 한다. 그러나 동일 프롬프트를 30회 반복 실행한 결과 출력 토큰 수가 3~8,000 토큰에 걸쳐 극단적으로 분산되었다(변동계수 CV=1.02). 전체 출력 분포는 500 토큰 미만과 3,500 토큰 이상에 집중된 이중봉(bimodal) 형태로, 단일 고정 평균값은 두 분포 구간을 모두 대표하지 못한다. 본 실험의 워크로드에서도 이러한 편차가 확인되었다. 본 논문에서 I는 지연에 민감한 Interactive 요청 그룹, B1-B3은 Batch 요청 그룹을 의미하며, 세부 워크로드 구성은 표 2에 제시한다. 실험에서 그룹별 관측 평균은 I와 B1이 2,618, B2가 2,443, B3가 1,916 토큰으로 나타났으며, 설정값 대비 최대 27.6%의 편차를 보였다. 또한 프롬프트 길이와 출력 길이 사이의 선형 관계는 본 실험 워크로드에서 매우 약하게 나타났고($r=+0.029$, $n=52$), ShareGPT 데이터셋[3]에서도 회귀 분석 결과 $R^2 \approx 0.01$에 그쳐 입력 길이 기반의 단순 회귀 모델이 본 실험 조건에서 출력 길이 변동을 충분히 설명하지 못함을 보여준다.

3. 제안 방법

본 논문은 앞서 확인한 d_eff의 런타임 변동성과 μo의 출력 분포 편차를 보정하기 위해, EMA 기반 Dynamic d_eff 갱신과 그룹별 Dynamic μo 온라인 갱신을 결합한 온라인 파라미터 적응 알고리즘을 제안한다.

3.1 Dynamic d_eff (EMA 기반)

앞서 관찰한 d_eff 가변성을 보정하기 위해 각 요청 완료 시마다 실측 디코드 시간(d_obs)을 계산하고 지수 이동평균(EMA)으로 d_eff를 점진적으로 갱신한다.

$$d\_obs = \frac{duration - P}{completion\_tokens}$$

$$d\_eff(t) = \alpha \cdot d\_obs + (1-\alpha) \cdot d\_eff(t-1), \quad \alpha = 0.1$$

감쇠 계수 α=0.1로 설정하며, 완료 토큰 50 미만인 단답은 신뢰도가 낮아 갱신에서 제외하고 워밍업 5건 미만 구간에는 사전 설정값을 유지한다. 그림 2는 EMA가 사전 설정값($d\times\varepsilon$=0.00952)에서 실측 수렴값(0.01027)으로 수렴하는 과정을 보여주며, 이를 통해 hat_T가 실제 시스템 부하를 반영하게 된다.

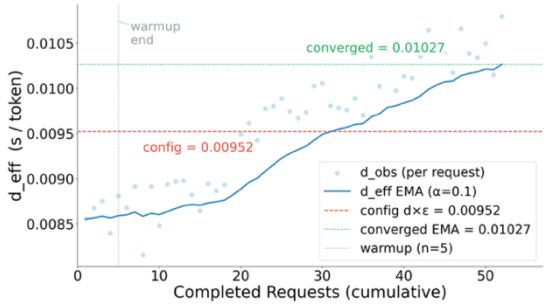

그림 2. Dynamic d_eff EMA 수렴 과정

3.2 Dynamic μo (런타임 온라인 갱신)

QLM이 오프라인에서 그룹별로 피팅한 μo는 런타임 중 변화하는 실제 출력 분포를 반영하지 못한다. 이를 해결하기 위해, 그룹 내에서 실제 완료된 요청들의 출력 토큰 수를 누적하여 해당 그룹의 μo를 런타임에 지속적으로 갱신한다. 시스템 전체 공통 EMA 방식도 검토하였으나, 초기 샘플 편향으로 다른 그룹의 추정 정확도를 저하시킬 수 있어 채택하지 않았다. 그룹별 방식에서는 유효 샘플 수 5건 이상, 완료 토큰 50 이상을 충족해야 보정값이 반영된다.

4. 실험 및 평가

4.1 실험 환경

표 1. 실험 환경

구성 요소	사양
LLM 모델	Qwen2.5-1.5B-Instruct
GPU	NVIDIA RTX 2080 SUPER (8 GB)
CUDA / vLLM	CUDA 12.8 / vLLM v0.16.0
정밀도	float16
최대 시퀀스 길이	8,192 tokens

실험 워크로드는 실제 사용자가 공유한 LLM 대화 기록으로 구성된 ShareGPT 데이터셋[3]을 기반으로 하였으며, 총 60건의 요청을 표 2와 같이 4그룹으로 구성하였다. 각 그룹의 SLO는 교정된 파라미터 기반의 $\widehat{T}$ 추정값 대비 충분한 여유를 갖도록 설계하였다. 이후 실험에서는 실제 완료시간과 추정 완료시간 사이의 오차를 MAPE로 측정하여 파라미터 적응 기법의 효과를 평가하였다.

표 2. 워크로드 구성

그룹	요청 수	SLO	hat_T (교정 후)
I (Interactive)	10	430 s	314 s
B1 (Batch-1)	15	520 s	463 s
B2 (Batch-2)	20	860 s	612 s
B3 (Batch-3)	15	1,320 s	463 s

4.2 Dynamic d_eff 효과

동일 워크로드에 대해 서버 재시작 후 Static 조건과 Dynamic d_eff 조건으로 각 1회씩 독립 실행하였다. 그 결과 Dynamic d_eff 적용 시 $\widehat{T}$ MAPE가 28.2%에서 25.7%로 2.5pp 감소하였다. d_eff의 EMA 수렴값은 사전 설정값 0.00952에서 실측값 0.01027로 갱신되었으며, 이를 통해 연속 처리 중 관측된 유효 디코드 시간 증가가 추정기에 반영되었음을 확인하였다.

4.3 Dynamic μo 효과 및 그룹별 분석

표 3은 정적 추정기, Dynamic d_eff 적용, Dynamic μo 추가 적용 조건에서의 그룹별 $\widehat{T}$ MAPE 변화를 나타낸다. Dynamic μo를 추가 적용한 결과, 전체 평균 MAPE는 25.7%에서 16.9%로 8.8pp 추가 감소하였다.

특히 μo 편차가 가장 컸던 B3에서는 Dyn. d_eff 단독 적용 시 오히려 MAPE가 일시적으로 상승하였으나, Dynamic μo 추가 적용 후 24.8%에서 10.9%로 가장 크게 개선되었다. 이는 d_eff 보정만으로는 출력 길이 추정 오차를 해결하기 어렵고, μo의 정확한 추정이 완료시간 예측 정확도에 중요한 영향을 미침을 보여준다. Dynamic μo는 실제 관측된 출력 토큰 수를 반영하여 그룹별 평균 출력 길이를 보정함으로써 이러한 오차를 완화하였다. B2와 같이 일부 그룹에서는 개선 효과가 제한적으로 나타났으나, 전체 평균 MAPE는 25.7%에서 16.9%로 감소하였다. 이는 출력 분포 변화에 대응하는 온라인 갱신이 전반적인 추정 정확도 향상에 효과적임을 보여준다.

표 3. 그룹별 $\widehat{T}$ MAPE 비교

그룹	Static	+Dyn. d_eff	+Dyn. μo	MAPE 변화
B1	16.4%	16.4%	3.8%	−12.6pp
I	46.3%	31.5%	27.2%	−19.1pp
B2	25.4%	19.0%	25.5%	+0.1pp
B3	24.8%	35.7%	10.9%	−13.9pp
평균	28.2%	25.7%	16.9%	−11.3pp

4.4 SLO 달성률 개선 실증

그룹별 출력 길이를 달리 설정한 추가 실험(85건, 3그룹)을 통해 Dynamic Estimator의 스케줄링 개선 효과를 검증하였다.

Static Estimator는 I 그룹의 완료 시간을 과대추정하여 불필요한 Eviction을 유발하였고, 그 결과 I 그룹 SLO를 위반하였다. 반면 Dynamic Estimator 적용 시 관측 토큰 수가 추정값에 반영되어 불필요한 Eviction을 피할 수 있었고, I 그룹은 SLO를 만족하였다.

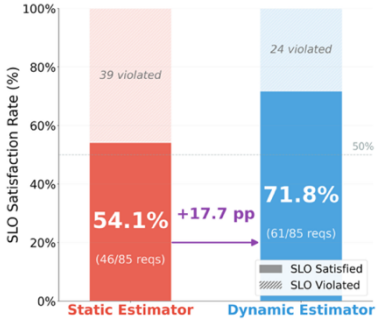

그림 3. Static vs Dynamic Estimator SLO 달성률 비교

전체 요청 기준 SLO 달성률은 Static 54.1%에서 Dynamic 71.8%로 17.7pp 향상되었다(그림 3). 이는 hat_T 추정 정확도 향상이 Eviction 및 재정렬 판단의 정확도를 높여 실질적인 SLO 성능 개선으로 이어질 수 있음을 보여준다.

5. 결 론

본 논문에서는 QLM의 RWT Estimator가 가정하는 정적 파라미터의 두 가지 한계를 probe 실험으로 실증하고, 이를 보정하기 위한 온라인 파라미터 적응 알고리즘을 제안하였다. 제안 알고리즘은 EMA 기반 d_eff 갱신과 그룹별 μo 온라인 갱신으로 구성된다. 제안 방법은 완료시간 추정 오차를 28.2%에서 16.9%로 감소시켰으며, 추가 실험에서 잘못된 스케줄링 결정을 방지하여 SLO 달성률을 17.7pp 향상시켰다. 이를 통해 추정 정확도 개선이 실질적인 서빙 품질 향상으로 이어질 수 있음을 보였다. 본 실험은 소규모 환경에서 수행되었으며, 대규모 모델, 고부하 조건, 반복 실행 기반 통계 검증, 디코드 시간 증가 원인에 대한 정량 분석은 향후 과제로 남긴다.

참 고 문 헌

[1] A. Patke, D. Reddy, S. Jha, H. Qiu, C. Pinto, C. Narayanaswami, Z. Kalbarczyk, and R. Iyer, "Queue management for slo-oriented large language model serving," ACM SoCC, pp. 18–35, 2024.

[2] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," ACM SOSP, pp. 611–626, 2023.

[3] RyokoAI, "ShareGPT52K Dataset," Hugging Face, 2023. [Online]. Available: https://huggingface.co/datasets/RyokoAI/ShareGPT52K