사물인터넷융복합논문지 Vol. 12, No. 3, pp. 117-123, 2026 https://doi.org/10.20465/KIOTS.2026.12.3.117

리눅스 파일시스템에 따른 Gemma 4 E2B 로컬 추론의 초기 구동 성능 분석

백영미, 박정규*
대진대학교 AI융합대학 컴퓨터공학전공 교수

Effects of Linux Filesystems on the Startup Performance of Local Inference with Gemma 4 E2B

Youngmi Baek, Jung Kyu Park*
Professor, Division of AI Convergence, Major in Computer Science, Daejin University

요 약 본 연구는 Linux 환경에서 Gemma 4 E2B 로컬 추론 시 ext4, XFS, Btrfs 파일시스템이 초기 구동 성능에 미치는 영향을 분석하였다. 동일한 하드웨어와 실행 조건에서 파일시스템과 cache 상태만을 변수로 두고 각 조건을 5회 반복 측정하였으며, wall time과 load duration을 중심으로 성능을 평가하였다. 또한 전체 응답시간의 변화가 실제로 어느 단계에서 발생하는지를 확인하기 위해 total duration, prompt evaluation duration, evaluation duration도 함께 수집하였다. 실험 결과, cold cache 조건에서는 파일시스템에 따른 차이가 뚜렷하게 나타났고, 본 실험 환경에서는 Btrfs가 가장 낮은 wall time과 load duration을 보였다. 반면 warm cache 조건에서는 세 파일시스템의 성능 차이가 크게 줄어들어 거의 유사한 수준으로 수렴하였다. 특히 전체 응답시간의 변화는 생성 단계보다 모델 로딩 단계에서 더 크게 나타났으며, 이는 초기 구동 성능이 저장장치 계층의 영향을 크게 받는다는 점을 시사한다. 이러한 결과는 로컬 LLM의 초기 구동 성능이 연산 성능뿐 아니라 파일시스템 선택과 page cache 상태의 영향을 함께 받는다는 점을 보여준다. 또한 실제 로컬 AI 운용 환경에서는 워크로드 특성을 고려한 저장장치 선택이 필요함을 확인하였다.

주제어 : Gemma 4 E2B, 로컬 LLM 추론, 리눅스 파일시스템, 초기 구동 성능, 페이지 캐시

Abstract This study examines how Linux filesystem choice affects the startup performance of Gemma 4 E2B in a local inference environment. Using a fixed hardware platform, we compared ext4, XFS, and Btrfs under cold-cache and warm-cache conditions, and repeated each experiment five times under the same model, prompt, and runtime settings. The analysis focused on wall time and load duration, while additional internal metrics were collected to identify which stage contributed most to the observed delay. Under cold-cache conditions, the three filesystems showed clear performance differences, and Btrfs achieved the lowest wall time and load duration in the tested setup. In contrast, the performance gap became much smaller under warm-cache conditions. The results also show that the change in end-to-end latency was driven more by the model-loading stage than by the generation stage itself. These findings indicate that startup behavior in local LLM deployment is influenced not only by compute capability but also by filesystem choice and page-cache state. The study suggests that storage configuration should be considered together with workload characteristics when designing practical local AI systems.

Key Words : Gemma 4 E2B, Local LLM Inference, Linux Filesystems, Startup Performance, Page Cache

*교신저자 : 박정규(jkpark@daejin.ac.kr)
접수일 2026년 05월 06일 수정일 2026년 06월 11일 심사완료일 2026년 06월 18일

1. 서론

최근 생성형 인공지능의 활용이 빠르게 확산되면서, 대규모 언어모델을 클라우드 API에만 의존하지 않고 로컬 환경에서 직접 운용하려는 수요도 함께 증가하고 있다. 이러한 수요는 응답 지연, 사용 비용, 네트워크 의존성, 데이터 통제와 같은 운영 요구와 직접 연결된다. 최근 해외 연구들은 자원 제약이 있는 edge 환경에서 대규모 언어모델을 실행하기 위한 설계와 실행 기법을 체계적으로 정리하고 있으며, small language model의 역할에 대해서도 활발히 논의하고 있다[1,2]. 국내에서도 검색 증강 생성 기반 소형 모델 활용, 모델 간 성능 및 응답시간 비교, 추론 효율 향상을 위한 문맥 축소 기법과 같은 연구가 이어지고 있다[3-5].

그러나 로컬 LLM의 사용성은 단순히 정답률이나 생성 품질만으로 설명되기 어렵다. 실제 사용자가 먼저 체감하는 것은 모델이 처음 적재되는 시간, 첫 응답이 출력되기까지의 지연, 반복 실행 시의 반응 안정성과 같은 시스템 수준의 성능이다. 최근 연구에서는 LLM 추론의 시작 구간에서 모델 로딩 자체가 주요 병목으로 작용할 수 있으며, 페이지 캐시 정책이 초기 응답시간에 유의미한 영향을 줄 수 있음을 확인하였다 [6]. 또한 국내 연구에서도 리눅스 I/O 인터페이스의 선택이 키-밸류 저장소의 처리량과 응답시간을 크게 바꿀 수 있음이 확인되었다 [7]. 따라서 로컬 LLM의 초기 응답 성능을 평가할 때는 GPU 연산 성능뿐 아니라 저장장치와 운영체제 계층도 함께 고려해야 한다.

리눅스의 대표 파일시스템인 ext4, XFS, Btrfs는 구조와 운용 특성이 달라 워크로드에 따라 서로 다른 성능을 보인다 [8-10]. 기존 연구들은 NVMe 기반 환경에서 ext4, XFS, Btrfs의 성능을 비교하거나, 리눅스 파일시스템의 진화 과정과 구조적 특성을 분석해 왔다[8,9]. 또한 저장장치 추상화와 메타데이터 관리 구조가 실제 성능에 미치는 영향도 지속적으로 논의되어 왔다[11-15]. 그러나 이들 연구는 주로 전통적인 입출력 벤치마크와 메타데이터 처리, 분산 스토리지 환경에 초점을 두고 있었다. 다시 말해, 양자화된 로컬 언어모델을 읽고 적재하는 과정에서 파일시스템 차이가 실제 체감 응답시간으로 어떻게 이어지는지를 분석한 연구는 아직 충분하지 않다.

본 연구는 Linux 환경에서 Gemma 4 E2B 로컬 추론을 대상으로 ext4, XFS, Btrfs 파일시스템의 성능 차이를 비교하고, cold cache와 warm cache 조건에서 그 변화 양상을 분석한다. 이를 위해 사용자가 실제로 느끼는 wall time(전체 응답시간)과 load duration(모델 로딩 시간)을 중심 지표로 삼아, 파일시스템 선택이 초기 구동 성능에 어떤 영향을 미치는지를 정량적으로 확인하고자 한다. 본 연구의 목적은 특정 파일시스템의 절대적 우위를 주장하는 데 있지 않다. 오히려 로컬 LLM을 실제 환경에서 운용할 때, 어떤 조건에서 저장장치 계층의 차이가 의미 있게 드러나는지를 밝히고, 이에 기반한 실무적 선택 기준을 제시하는 데 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 정리하고, 3장에서는 실험 환경 및 방법을 설명한다. 4장에서는 실험 결과와 분석을 제시하며, 5장에서는 결론과 향후 과제를 정리한다.

2. 관련 연구

2.1 로컬 LLM 실행 환경과 소형 모델 연구

최근 로컬 또는 edge 환경에서 대규모 언어모델을 실행하려는 연구가 빠르게 늘고 있다. 관련 해외 리뷰 논문들은 제한된 메모리와 연산 자원을 가진 환경에서 LLM을 구동하기 위해 양자화, 경량화, 실행 엔진 최적화, 온디바이스 배치 전략이 중요하다고 정리하고 있다[1,2]. 국내 연구 역시 소형 LLM에 검색 증강이나 미세조정을 결합하는 방향, 여러 모델의 응답시간과 성능을 함께 비교하는 방향으로 확장되고 있다[3,4].

해외에서는 생성형 LLM의 서빙 효율을 높이기 위한 연구도 활발히 진행되고 있다[5]. 그러나 이러한 연구는 주로 모델 성능, 응답 품질, 추론 가속에 초점을 두고 있어, 저장장치 계층이 초기 응답시간에 미치는 영향은 상대적으로 적게 다루고 있다. 최근 관련 시스템 연구에서 모델 로딩과 페이지 캐시가 초기 응답시간에 영향을 주는 것을 검증하였다[6].

2.2 리눅스 파일시스템 및 성능 연구

리눅스 파일시스템에 관한 선행연구는 ext4, XFS, Btrfs의 구조적 차이와 성능 특성을 꾸준히 분석해 왔다. Lu 등은 Linux 파일시스템의 진화 과정을 정리하면서 ext 계열 파일시스템의 구조 변화와 성능 특성을 분석하였고, Rodeh 등은 Btrfs의 Copy-on-Write 구조와 스냅샷, 체크섬, 확장성 중심 설계를 설명하였다[9,10]. 또한 ext4, XFS, Btrfs를 동일 조건에서 비교한 연구들은 워크로드에 따라 서로 다른 장단점이 나타난다고 보고하

였다[8,11]. 국내 연구에서도 NVMe 기반 환경에서 ext4, XFS, Btrfs를 비교하여 랜덤 I/O, 순차 I/O, 메타데이터 처리, SQLite 수행시간이 파일시스템 구조에 따라 달라질 수 있음을 제시하였다[8].

저장장치 소프트웨어 계층 전반을 다룬 연구들은 파일시스템 성능이 단순히 디스크 자체의 속도만으로 결정되지 않음을 보여준다. 리눅스 I/O 인터페이스의 차이가 응용 성능에 영향을 줄 수 있다는 연구[7], SSD 추상화 계층과 소프트웨어 스택의 변화에 대한 논의[15], 그리고 메타데이터 관리 구조와 I/O 자원 분배 방식이 대규모 저장환경의 성능과 확장성에 영향을 준다는 연구가 수행되었다[12-14]. 그러나 이러한 연구의 중심은 전통적인 저장장치 워크로드나 분산 스토리지 환경에 머물러 있으며, 로컬 LLM의 모델 로딩과 같은 최신 AI 워크로드에 직접 적용한 사례는 아직 많지 않다.

2.3 기존 연구와의 차별성

기존 연구는 모델·응용 연구와 저장장치·파일시스템 연구로 구분할 수 있다. 하나는 소형 LLM, RAG, 모델 비교, 추론 최적화와 같은 모델·응용 중심 연구이고[1-6], 다른 하나는 ext4, XFS, Btrfs의 구조와 전통적 I/O 성능을 다룬 저장장치·파일시스템 연구이다[7-15]. 전자는 모델의 활용성과 응답 특성을 설명하는 데 강점이 있고, 후자는 파일시스템 구조와 워크로드의 관계를 분석하는 데 강점이 있다. 그러나 두 흐름을 직접 연결하여, 로컬 LLM의 초기 실행 성능을 파일시스템 관점에서 비교한 연구는 제한적이다.

이에 본 연구는 동일한 하드웨어 환경에서 Gemma 4 E2B를 대상으로 ext4, XFS, Btrfs를 비교하고, cold cache와 warm cache 조건을 구분하여 성능을 분석한다. 또한 사용자 체감 응답시간에 해당하는 wall time과 모델 파일 적재 비용을 반영하는 load duration을 함께 비교함으로써, 파일시스템 차이가 실제로 어느 단계에서 나타나는지를 살펴본다. 본 연구의 차별점은 파일시스템 비교를 로컬 LLM의 초기 구동 문제에 직접 연결했다는 데 있다. 특히 모델 품질이 아니라 초기 구동 성능과 로딩 경로를 분석 대상으로 삼았다.

3. 실험 방법

3.1 연구 개요

본 장에서는 Gemma 4 E2B의 로컬 추론 성능을 ext4, XFS, Btrfs 환경에서 비교하기 위한 실험 설계와 측정 방법을 설명한다. 비교 변수는 파일시스템과 cache 상태로 한정하고, 나머지 실행 조건은 동일하게 통제하였다.

특히 실험에서 cold cache와 warm cache 조건을 구분하여 초기 로딩 지연의 차이를 분석하는 데 초점을 두었다. 본 연구에서는 wall time과 load duration을 핵심 지표로 사용하였다. 이를 통해 파일시스템 차이가 실제로 어느 단계에서 나타나는지 확인하고자 하였다.

3.2 실험 환경

실험은 Ubuntu Server 24.04 LTS가 설치된 단일 시스템에서 수행하였다. 실험 장비는 Intel Core 5 245K, 32 GB RAM, NVIDIA GeForce RTX 3060 8 GB, NVMe SSD 500 GB 2개로 구성하였다. 운영체제와 실험 도구는 첫 번째 SSD에 설치하였고, 두 번째 SSD는 파일시스템 비교를 위한 실험 전용 디스크로 사용하였다. 실험 전용 디스크에는 ext4, XFS, Btrfs를 각각 독립 파티션으로 구성하여, 운영체제의 백그라운드 입출력이 실험 결과에 미치는 영향을 줄이고자 하였다. 전체 실험 환경과 주요 실행 설정은 Table 1에 요약하였다.

Table 1. Experimental Environment Summary

Category	Configuration
O/S	Ubuntu Server 24.04 LTS
Hardware	Intel Core 5 245K, 32 GB RAM, NVIDIA GeForce RTX 3060 8 GB
Storage	1. Samsung 970 Evo Plus NVMe 500 GB (OS and tools) 2. Micron P5 Plus NVMe 500 GB (Filesystem experiments)
Filesystems	ext4, XFS, Btrfs
Inference Runtime	Ollama 0.20.4
Model	Gemma 4 E2B
Fixed Runtime Options	OLLAMA_KEEP_ALIVE=0, OLLAMA_CONTEXT_LENGTH=4096, OLLAMA_NUM_PARALLEL=1, think=false, stream=false

추론 런타임은 Ollama 0.20.4를 사용하였으며, 대상 모델은 Gemma 4 E2B(Ollama model tag: gemma4:e2b)로 고정하였다. 본 실험에서는 파일시스템 외의 영향을 줄이기 위해 모든 조건에 동일한 런타임 옵션과 API 요청 설정을 적용하였다.

3.3 실험 절차 및 측정 지표

실험 변수는 파일시스템과 cache 상태의 두 가지로 구성하였다. 본 논문에서 cache는 page cache를 의미하며, 이후에는 cold cache와 warm cache로 표기한다.

파일시스템은 ext4, XFS, Btrfs의 세 수준으로 구분하였고, cache 상태는 cold와 warm으로 나누었다. cold cache 조건에서는 측정 직전에 sync를 수행하고 page cache를 정리한 뒤 요청을 실행하였다. warm cache 조건에서는 동일한 요청을 한 차례 선행 실행하여 관련 페이지를 메모리에 적재한 뒤, 다음 요청을 측정값으로 사용하였다. 각 조건은 5회 반복 수행하였으며, 모든 실험에서 동일한 모델, 동일한 프롬프트, 동일한 실행 옵션을 유지하였다. 이는 파일시스템과 cache 상태를 제외한 다른 변수의 영향을 최소화하기 위한 것이다. 또한 각 파일시스템은 실험 전용 SSD의 독립 파티션에 구성하였으며, 동일한 모델 파일을 각 파일시스템에 저장한 상태에서 비교하였다.

성능 평가는 wall time, total duration, load duration, prompt evaluation duration, evaluation duration의 다섯 지표를 기준으로 수행하였다. wall time은 외부 실행 시간 측정을 통해 수집하였고, total duration, load duration, prompt evaluation duration, evaluation duration은 Ollama API가 반환한 내부 성능 지표를 사용하였다. load duration은 모델 파일 로딩 및 초기화 시간을 반영하는 핵심 지표이며, total duration은 내부 총 처리시간을 의미한다. 또한 prompt evaluation duration과 evaluation duration은 각각 프롬프트 처리 구간과 생성 구간의 시간을 나타낸다. 본 연구에서는 특히 wall time과 load duration을 중심으로 결과를 분석하였으며, 이는 로컬 LLM의 초기 구동 성능을 해석할 때 모델 로딩 비용이 중요한 변수라는 최근 연구 결과와도 연결된다[6].

4. 실험 결과 및 분석

본 장에서는 ext4, XFS, Btrfs 파일시스템에서 수행한 Gemma 4 E2B 추론 실험 결과를 제시하고, cold cache와 warm cache 조건에 따른 성능 변화를 분석한다. 모든 실험은 동일한 하드웨어와 동일한 실행 조건에서 수행하였으며, 각 조건은 5회 반복 측정한 뒤 평균과 표준편차를 산출하였다. Table 2는 전체 성능 요약을 제시하며, Fig. 1은 cold cache에서 warm cache로 전환될 때 wall time과 load duration의 감소율을, Fig. 2는 total duration에서 각 처리 단계가 차지하는 비율을 나타낸다.

4.1 전체 성능 비교

Table 2에서 보는 것처럼, 세 파일시스템 모두 warm cache 조건에서 cold cache 조건보다 짧은 응답시간을 보였다. cold cache 조건에서는 Btrfs가 가장 낮은 wall time과 load duration을 기록하여 초기 구동 구간에서 상대적으로 유리한 성능을 나타냈다. 반면 warm cache 조건에서는 세 파일시스템의 차이가 크게 줄어들었다. 이 결과는 Gemma 4 E2B의 초기 응답 시간이 생성 연산보다 모델 적재 경로의 영향을 더 크게 받는 것을 보여준다. 이러한 해석은 LLM 추론에서 모델 로딩이 주요 병목이 될 수 있다고 보고한 기존 연구와도 맥락을 같이한다[6].

반면 warm cache 조건에서는 세 파일시스템의 차이가 뚜렷하지 않았다. warm 상태의 wall time 차이는 최대 0.067초, load duration 차이는 최대 0.013초 수준으로 매우 작았으며, 표준편차 역시 전반적으로 줄어들었다. 이는 모델 관련 페이지가 메모리에 유지되는 상황에서는 파일시스템 자체보다 cache 상태가 더 큰 영향을 미칠 수 있음을 보여준다. 또한 파일시스템의 성능은

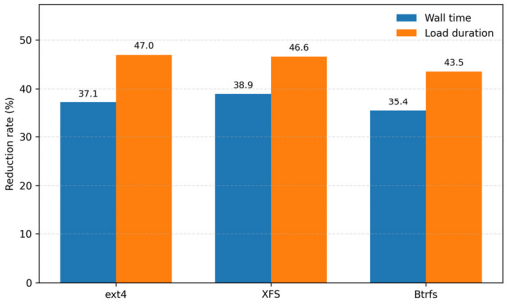
Table 2. Performance Comparison of Gemma 4 E2B across Filesystems and Cache Conditions

File system	Cache status	Wall time(s)	Total duration(s)	Load duration(s)	Prompt evaluation duration (s)	Evaluation duration (s)
ext4	cold	4.633 ± 0.156	4.607 ± 0.156	3.735 ± 0.019	0.028 ± 0.003	0.703 ± 0.151
	warm	2.913 ± 0.076	2.904 ± 0.076	1.981 ± 0.017	0.024 ± 0.000	0.761 ± 0.074
XFS	cold	4.660 ± 0.160	4.634 ± 0.160	3.694 ± 0.021	0.031 ± 0.003	0.764 ± 0.144
	warm	2.846 ± 0.088	2.838 ± 0.087	1.973 ± 0.010	0.024 ± 0.000	0.708 ± 0.074
Btrfs	cold	4.454 ± 0.102	4.430 ± 0.102	3.486 ± 0.020	0.031 ± 0.002	0.773 ± 0.104
	warm	2.878 ± 0.042	2.870 ± 0.042	1.968 ± 0.007	0.024 ± 0.000	0.750 ± 0.039

단일 우열보다 워크로드 특성과 접근 경로에 따라 달라진다는 기존 연구의 관찰과도 일치한다 [8-11].

4.2 cache 상태에 따른 응답시간 변화

Fig. 1은 cold cache에서 warm cache로 전환될 때 wall time과 load duration이 얼마나 감소하는지를 파일시스템별로 비교한 결과이다. wall time 감소율은 ext4 37.1%, XFS 38.9%, Btrfs 35.4%로 나타났고, load duration 감소율은 ext4 47.0%, XFS 46.6%, Btrfs 43.6%로 나타났다. 세 파일시스템 모두에서 load duration 감소율이 wall time 감소율보다 크게 나타났으며, 이는 warm cache 조건에서의 성능 개선이 주로 모델 로딩 구간에서 발생했음을 의미한다. 즉, page cache가 활성화되면 전체 응답시간이 줄어들지만, 그 효과는 wall time 감소보다 load duration 감소에서 더 직접적으로 나타난다.

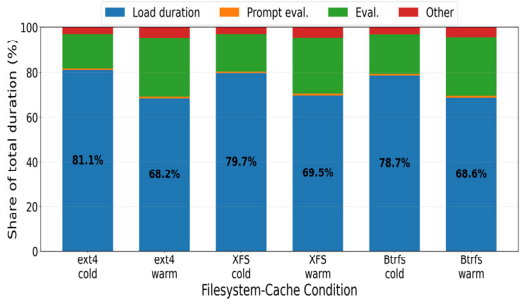


[Fig. 1] Reduction Rates of Wall Time and Load Duration from Cold Cache to Warm Cache

이러한 감소율 차이는 실행 방식에 따라 유리한 파일시스템이 달라질 수 있음을 보여준다. 모델을 자주 내리고 다시 불러오는 개발·실험 환경에서는 cold cache 성능 차이가 실제 체감 품질에 직접 연결될 수 있다. 반대로 모델이 장시간 메모리에 유지되는 서비스 환경에서는 파일시스템 간 차이보다 cache 유지 전략이나 모델 재사용 정책이 더 중요한 변수로 작용할 가능성이 크다. 따라서 로컬 LLM 환경에서 저장장치 계층을 평가할 때에는 일반적인 I/O 벤치마크 결과만이 아니라, 실제 실행 패턴과 서비스 조건을 함께 고려해야 한다[6,8].

4.3 로딩 단계의 성능 분석

Fig. 2는 total duration에서 각 처리 단계가 차지하는 비율을 파일시스템과 cache 조건별로 나타낸다. cold cache 조건에서 load duration 비중은 ext4 81.1%, XFS 79.7%, Btrfs 78.7%였고, warm cache 조건에서는 각각 68.2%, 69.5%, 68.6%로 낮아졌다. 반면 prompt evaluation duration과 evaluation duration의 비중은 상대적으로 작았으며, warm cache 조건에서는 load duration 비중이 줄어드는 대신 evaluation 구간의 상대 비중이 커졌다. 이는 cold cache 조건의 전체 성능 차이가 주로 모델 로딩 단계에서 발생하고, warm cache 조건에서는 그 영향이 완화됨을 보여준다.



[Fig. 2] Breakdown of Total Duration across Filesystems and Cache Conditions

절대값 기준으로 보면, prompt evaluation duration과 evaluation duration은 파일시스템에 따라 큰 차이를 보이지 않았다. prompt evaluation duration은 약 0.024~0.031초, evaluation duration은 약 0.703~0.773초 범위에 머물렀으며, load duration에 비해 변동 폭이 훨씬 작았다. 따라서 본 실험의 성능 차이는 생성 단계보다 모델 파일 로딩과 초기화 경로에서 주로 발생한 것으로 해석할 수 있다.

4.4 요약

실험 결과를 요약하면 다음 두 가지 경향이 확인된다. 첫째, 파일시스템 간 성능 차이는 분명히 존재하지만, 그 차이는 주로 cold cache 조건의 모델 로딩 구간에서 두드러졌다. 둘째, warm cache 상태에서는 세 파일시스템의 성능이 빠르게 수렴하였고, 이때는 파일시스템 자체보다 cache 상태가 더 큰 영향을 미쳤다. 본 실험 조건에서는 Btrfs가 cold cache 조건에서 가장 낮은 로딩 시간을 보였지만, 이를 모든 환경에 일반화하기보다는 현재 하드웨어와 모델 크기, 실행 방식에 한정된 결과로 해석하는 것이 적절하다.

5. 결론

본 연구는 Linux 환경에서 Gemma 4 E2B 로컬 추론을 대상으로 ext4, XFS, Btrfs 파일시스템의 성능 차이를 비교하고, cold cache와 warm cache 조건에서의 응답시간 변화를 분석하였다. 실험 결과, cold cache 조건에서는 파일시스템에 따른 차이가 분명하게 나타났으며, 본 실험 환경에서는 Btrfs가 가장 낮은 wall time과 load duration을 보였다. 반면 warm cache 조건에서는 세 파일시스템의 성능 차이가 크게 줄어들었고, 전체 응답시간의 변화 역시 생성 단계보다 모델 로딩 단계에서 더 크게 나타났다. 이는 로컬 LLM의 초기 구동 성능이 연산 성능뿐 아니라 저장장치 계층과 cache 상태의 영향을 함께 받는다는 점을 보여준다.

따라서 로컬 LLM 운용 환경에서는 파일시스템 선택이 실제 응답성에 영향을 줄 수 있다. 특히 모델을 자주 다시 불러오는 개발·실험 환경에서는 파일시스템의 차이가 의미 있게 작용할 수 있으며, 모델이 장시간 상주하는 서비스 환경에서는 cache 유지 전략이 더 중요한 최적화 요소가 될 수 있다. 본 연구는 파일시스템 성능 비교를 LLM 실행 환경으로 확장했다는 점에서 의의가 있으며, 향후에는 더 큰 모델과 다양한 저장장치 및 하드웨어 환경을 포함한 후속 연구가 필요하다.

REFERENCES

[1] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu and J. Chen, "A Review on Edge Large Language Models: Design, Execution, and Applications," *ACM Computing Surveys*, Vol.57, No.8, pp.1-35, 2025.

[2] F. Wang, Z. Zhang, X. Zhang, Z. Wu, T. H. Mo, Q. Lu, W. Wang, R. Li, J. Xu, X. Tang, Q. He, Y. Ma, M. Huang and S. Wang, "A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness," *ACM Transactions on Intelligent Systems and Technology*, Vol.16, No.6, pp.1-87, 2025.

[3] W. Cho, J. Yoo, S. Kim and J. Jang, "RAG-Enhanced small Large Language Models: Enhancing Battlefield Analysis through Knowledge Distillation of Large Language Models," *Journal of The Korea Society of Computer and Information*, Vol.30, No.3, pp.43-57, 2025.

[4] M. Lee, "A Comparative Study on the Performance of GPT-4o-mini, Claude 4 Sonnet, and Gemini 2.5 Flash Models Using the Prompt Runner Framework," *Journal of The Korea Society of Computer and Information*, Vol.31, No.2, pp.43-50, 2026.

[5] X. Miao, G. Oliaro, Z. Zhang, X. Cheng, H. Jin, T. Chen and Z. Jia, "Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems," *ACM Computing Surveys*, Vol.58, No.1, pp.1-37, 2025.

[6] Y. Liu, H. Li, X. Huang, Y. Wang, H. Guo, H. Chen, Y. Ren and N. Jia. (2026). Accelerating Model Loading in LLM Inference by Programmable Page Cache, in Proceedings of the *24th USENIX Conference on File and Storage Technologies (FAST '26)*, pp.117-132, 2026.

[7] Y. Song and Y. I. Eom, "Analyses of Linux I/O Interfaces for High-Performance I/O Operations in Key-Value Stores," *Journal of KIISE*, Vol.48, No.12, pp.1274-1280, 2021.

[8] J. K. Park and Y. Baek, "Comparative Performance Analysis of Linux Filesystems (ext4, XFS, and Btrfs) on NVMe Storage," *Journal of Next-generation Convergence Technology Association*, Vol.9, No.12, pp.3384-3391, 2025.

[9] L. Lu, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau and S. Lu, "A Study of Linux File System Evolution," *ACM Transactions on Storage*, Vol.10, No.1, pp.1-32, 2014.

[10] O. Rodeh, J. Bacik and C. Mason, "BTRFS: The Linux B-Tree Filesystem," *ACM Transactions on Storage*, Vol.9, No.3, pp.1-32, 2013.

[11] M. B. Ab Karim, J. Y. Luke, M. T. Wong, P. Y. Sian and O. Hong, "Ext4, XFS, BtrFS and ZFS Linux file systems on RADOS Block Devices (RBD): I/O performance, flexibility and ease of use comparisons," in Proceedings of the *2016 IEEE Conference on Open Systems (ICOS)*, pp.18-23, 2016.

[12] D. Huang, J. Wang, Q. Liu, N. Xiao, H. Wu and J. Yin, "Enhancing Proportional IO Sharing on Containerized Big Data File Systems," *IEEE Transactions on Computers*, Vol.70, No.12, pp.2083-2097, 2021.

[13] H. J. Singh and S. Bawa, "Scalable Metadata Management Techniques for Ultra-Large Distributed Storage Systems: A Systematic Review," *ACM Computing Surveys*, Vol.51, No.4, pp.1-37, 2018.

[14] H. Dai, Y. Wang, K. B. Kent, L. Zeng and C.-Z. Xu, "The State of the Art of Metadata Managements in Large-Scale Distributed File Systems: Scalability, Performance and Availability," *IEEE Transactions on Parallel and Distributed Systems*, Vol.33, No.12, pp.3850-3869, 2022.

[15] X. Zhang, J. Bhimani, S. Pei, E. Lee, S. Lee, Y. J. Seong, E. J. Kim, C. Choi, E. H. Nam, J. Choi and B. S. Kim, "Storage Abstractions for SSDs: The Past, Present, and Future," *ACM Transactions on Storage*, Vol.21, No.1, pp.1-44, 2025.

백 영 미(Youngmi Baek) [정회원]

- 2015년 2월 : 경북대학교 컴퓨터공학과 (공학박사)
- 2015년 7월 ~ 2017년 1월 : 대구경북과학기술원 CPS 글로벌 센터 연구원
- 2017년 2월 ~ 2019년 12월 : 대구경북과학기술원 정보통신융합전공 연구교수
- 2020년 3월 ~ 2026년 2월 : 창신대학교 컴퓨터공학과 교수
- 2026년 3월 ~ 현재 : 대진대학교 AI융합학부 컴퓨터공학전공 교수

〈관심분야〉
System modeling, Network security within automotive cyber-physical systems, Autonomous manufacturing

박 정 규(Jung Kyu Park) [종신회원]

- 2013년 8월 : 홍익대학교 컴퓨터공학과 (공학박사)
- 2014년 1월 ~ 2016년 2월 : 단국대학교 컴퓨터학과 연구교수
- 2015년 10월 ~ 2017년 2월 : 울산과학기술원 전기전자컴퓨터공학부 연구원
- 2018년 3월 ~ 2024년 8월 : 창신대학교 컴퓨터공학과 교수
- 2024년 9월 ~ 현재 : 대진대학교 AI융합학부 컴퓨터공학전공 교수

〈관심분야〉
Data storage, Robotics, System software, Embedded systems