제1장. 소개 및 AI 시스템 개요

이 작품은 AI를 사용하여 번역되었습니다. 여러분의 피드백과 의견을 환영합니다: translation-feedback@oreilly.com

---

2024년 말, 중국에 위치한 작은 스타트업인 DeepSeek.AI는 당시 최신 NVIDIA GPU를 사용하지 않고도 *최첨단* 대규모 언어 모델(LLM)을 훈련시켜 인공지능 커뮤니티를 놀라게 했습니다. 수출 제한으로 인해 DeepSeek의 엔지니어들은 최상위 NVIDIA Blackwell(B200, B300 등) 또는 Hopper(H100, H200 등) GPU를 구할 수 없었기 때문에, 당시 현지에서 구할 수 있고 수출 규정을 준수하는 NVIDIA H800 GPU를 비롯한 대안을 사용했습니다. 그들은 커스텀 커널과 모델 증류와 같은 고급 최적화 기법을 사용하여 성능이 다소 떨어지는 이 GPU들로부터 최대한의 성능을 끌어냈습니다.

이러한 한계에도 불구하고 DeepSeek.AI는 DeepSeek-R1 모델을 훈련시켜 당시 최고 성능의 NVIDIA 칩으로 훈련된 선도적인 프론티어 모델에 근접한 추론 능력을 달성했습니다. 이 사례는 AI 시스템 성능 엔지니어링에 능숙한 실무자와 연구자들이 제약 조건과 상관없이 사용 가능한 하드웨어에서 최대한의 성능을 끌어낼 수 있음을 보여줍니다.

예를 들어, DeepSeek 엔지니어들은 통신 대역폭을 *희소 자원으로* 간주하여, 해당 인프라에서는 불가능하다고 여겨졌던 성과를 달성하기 위해 전송되는 모든 바이트를 최적화했습니다. 그들은 제한된 대역폭의 상호 연결로 연결된 수천 개의 이러한 제약된 GPU로 확장하면서, 이러한 한계를 극복하기 위해 독창적인 소프트웨어 및 알고리즘 최적화를 활용했습니다.

DeepSeek의 접근법을 미국과 유럽의 주요 AI 연구소들이 취한 '무차별적 접근법'과 비교해 보십시오. 이들 연구소는 여전히 더 큰 컴퓨팅 클러스터와 더 큰 모델을 추구합니다. 모델 규모는 수백만 개에서 수십억 개, 이제는 수조 개의 매개변수로 폭발적으로 증가했습니다. 규모가 10배 증가할 때마다 질적으로 새로운 능력이 열리긴 했지만, 이는 막대한 비용과 자원을 요구합니다.

예를 들어, OpenAI의 GPT-4(2023년) 훈련 비용은 약 1억 달러로 추정되며, Google의 Gemini Ultra(2023년 말) 훈련 비용은 무려 약 1억 9,100만 달러로 추산됩니다. 이는 모델의 규모와 비용이 증가함에 따라 향후 자원 효율성의 필요성을 보여줍니다.

DeepSeek는 자사의 초고성능 모델 DeepSeek-R1이 600만 달러 미만의 컴퓨팅 비용으로 훈련되었다고 주장합니다. 이는 GPT-4나 Gemini Ultra 같은 모델보다 한 차원 낮은 수준입니다. 동시에 DeepSeek-R1은 수십 배 더 많은 비용이 든 경쟁 모델들의 성능과 맞먹습니다.

는 600만 달러 주장의 타당성(예: 단일 훈련 실행만 포함)과 제외 항목(예: 실험 및 모델 개발 파이프라인)에 대해 일부 의문을 제기했지만, 이 발표는 미국 금융 시장을 일시적으로 충격에 빠뜨렸습니다. 특히 NVIDIA 주가는 이 소식에 하루 만에 약 17% 급락했습니다. 이는 딥시크의 효율성 혁신이 향후 엔비디아 하드웨어 수요를 감소시킬 것이라는 우려에서 비롯되었습니다. 비록 이러한 시장 반응이 다소 과장되었고 엔비디아 주가는 이후 거래 세션에서 회복되었지만, 이는 AI 효율성 분야의 혁신이 글로벌 금융 시장에 미칠 수 있는 상당한 재정적 영향을 보여줍니다.

모델 훈련을 넘어, DeepSeek는 현대적 최첨단 LLMs의 핵심인 트랜스포머 아키텍처에 하드웨어 인식 알고리즘 개선을 도입해 추론 효율성을 크게 향상시켰습니다. DeepSeek는 지능적인 AI 시스템 성능 공학적 최적화가 초대형 AI 모델 훈련 및 추론의 경제성을 뒤집을 수 있음을 분명히 입증했습니다. 이러한 최적화 기법은 이 책의 나머지 부분에서 다루고 있습니다.

핵심은 이러한 규모에서는 시스템에서 짜낸 성능의 모든 부분이 수백만 달러, 심지어 수십억 달러의 절감으로 이어질 수 있다는 깊은 깨달음입니다. 제거된 모든 병목 현상은 훈련 처리량과 추론 지연 시간에 막대한 영향을 미칠 수 있습니다. 이는 결국 비용을 절감하고 최종 사용자의 전반적인 만족도를 높입니다. 간단히 말해, AI 시스템 성능 엔지니어링은 단순히 속도만을 위한 것이 아닙니다. 이전에는 불가능했던 것을 가능하고 경제적으로 실현하는 것입니다.

제1장에서는 대규모 인공지능 시대에 핵심적인 역할을 맡게 된 AI 시스템 성능 엔지니어에 대한 심층적 탐구를 시작합니다. 이 장은 현대 AI 시스템에 미치는 이 직업의 다각적인 책임과 중대한 영향을 이해하기 위한 포괄적인 가이드 역할을 합니다.

먼저 AI 워크로드의 진화 과정을 추적하며, 전통적인 컴퓨팅 패러다임에서 현대 AI 애플리케이션의 요구사항으로의 전환을 조명합니다. 이러한 맥락은 AI 분야에서 전문적인 성능 엔지니어링의 필요성을 이해하는 토대를 마련합니다.

이어서 본 장은 AI 시스템 성능 엔지니어에게 요구되는 핵심 역량에 대해 심층적으로 다룹니다. 하드웨어 아키텍처에 대한 깊은 이해, 소프트웨어 최적화 기법, 시스템 수준 통합 등 이 역할에 필수적인 기술적 전문성을 살펴봅니다. 또한 AI 프로젝트의 학제적 특성을 효과적으로 관리하는 데 중요한 문제 해결 능력, 커뮤니케이션, 협업과 같은 소프트 스킬의 중요성도 논의합니다.

본 장의 상당 부분은 역할의 실무적 측면에 할애됩니다. 성능 엔지니어가 시스템 병목 현상을 분석하고, 최적화 전략을 구현하며, AI 시스템의 확장성과 신뢰성을 보장하는 방법을 탐구합니다. 이러한 개념을 설명하기 위해 실제 시나리오와 사례 연구를 제시하여 현장에서 마주하는 도전과 해결책의 구체적인 사례를 제공합니다.

또한 성능 엔지니어가 일반적으로 사용하는 도구와 방법론을 논의하며, 성능 테스트, 모니터링, 벤치마킹 관행에 대한 통찰력을 제공합니다. 여기에는 업계 표준 도구의 개요와 시스템 성능 평가 및 향상에 적용되는 방식이 포함됩니다.

1장을 마치면 독자는 AI 시스템 성능 엔지니어의 역할, 이 직무에서 탁월함을 발휘하기 위해 필요한 기술, 그리고 AI 시스템의 성공적인 배포 및 운영에 있어 성능 엔지니어링의 중요성에 대한 철저한 이해를 갖게 될 것입니다. 이 기초 지식은 후속 장에서 AI 성능 엔지니어링의 탁월함을 정의하는 구체적인 기술, 기술 및 모범 사례를 더 깊이 탐구하는 토대를 마련합니다.

AI 시스템 성능 엔지니어

AI 시스템 성능 엔지니어 는 AI *모델과* 이를 실행하는 기반 시스템의 성능 최적화에 집중하는 전문 역할입니다. 이 엔지니어들은 AI 훈련 및 추론 파이프라인이 빠르고 비용 효율적이며 성능이 우수할 뿐만 아니라 안정적이고 고가용성을 유지하도록 보장합니다. 규모가 커질수록 AI 시스템 성능 엔지니어의 역할은 더욱 중요해집니다.

AI 시스템 성능 엔지니어는 최고 수준의 연봉을 받으며, 그 이유는 분명합니다. 우리의 작업은 수익성에 직접적인 영향을 미칩니다. 우리는 하드웨어, 소프트웨어, 알고리즘에 걸친 전문성을 융합합니다. 저수준 OS 고려사항, 메모리 계층 구조, 네트워킹 기본 원리, Python 및 C++와 같은 다양한 언어는 물론 PyTorch, OpenAI의 Triton, NVIDIA의 컴퓨트 통합 디바이스 아키텍처(CUDA)와 같은 다양한 AI 프레임워크 및 라이브러리를 이해해야 합니다.

AI 시스템 성능 엔지니어는 일상적으로 저수준 GPU 커널 효율성 검토, OS 스레드 스케줄링 최적화, 메모리 접근 패턴 분석, 네트워크 처리량 효율성 증대, 분산

훈련 알고리즘 디버깅 등의 업무를 수행합니다. AI 시스템 성능 엔지니어의 주요 책임에는 벤치마킹, 자질, 디버깅, 최적화, 확장성 확보, 자원 효율적 관리가 포함됩니다. 성능 엔지니어는 하드웨어, 소프트웨어, 알고리즘의 조합에 특화될 수 있지만, 핵심은 이러한 전문 분야들이 함께 설계되어야 한다는 점입니다(그림 1-1 참조). 따라서 각 분야의 장단점과 상호 영향을 이해하는 것이 중요합니다.

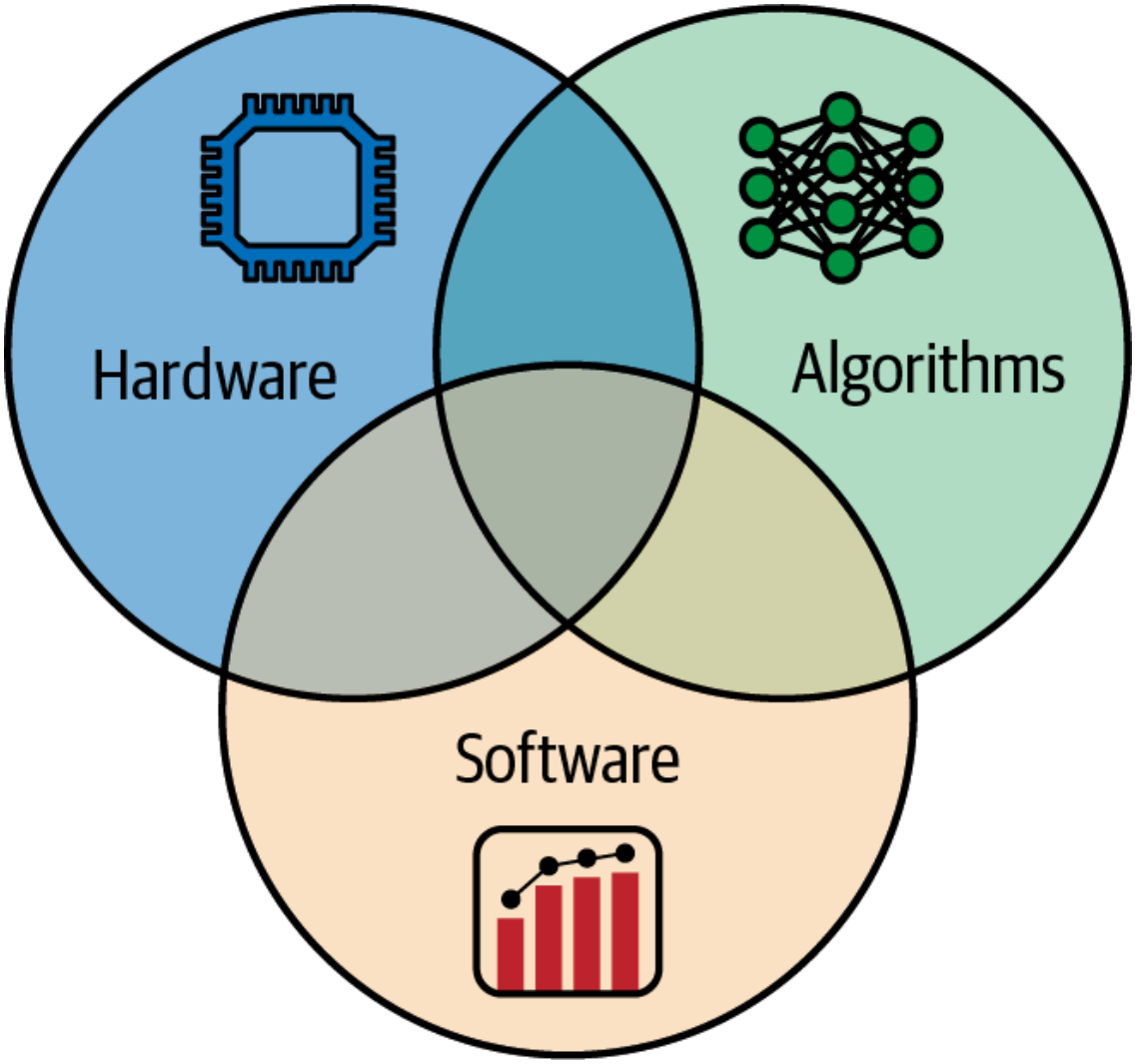

그림 1-1. 하드웨어, 소프트웨어, 알고리즘의공동 설계

벤치마킹과 자질

벤치마킹과 자질은 다양한 워크로드(훈련 및 추론 포함) 하에서 AI 모델의 지연 시간, 처리량, 메모리 사용량 및 기타 성능 지표를 측정하는 작업입니다( ). 병목 현상을 식별하려면 NVIDIA Nsight Systems 및 NVIDIA Nsight Compute를 PyTorch 프로파일러와 함께 반복적으로 사용해야 합니다( ). 이러한 도구를 결합하면 AI 시스템의 전반적인 성능을 지속적으로 개선해 나가면서 스택의 다양한 수준에서 병목 현상을 정확히 찾아내고 시간 경과에 따른 성능을 추적하는 데 도움이 됩니다.

개발 주기 초기에 성능 저하(regression)를 포착하기 위해 자동화된 성능 테스트를 설정하는 것이 중요합니다.

성능 문제의 디버깅 및 최적화를 위해서는 성능 문제의 근본 원인을 추적해야 합니다. 이는 비효율적인 CUDA 커널, 불필요한 통신 오버헤드, 또는 훈련 또는 추론 워크로드의 불균형 등일 수 있습니다.

한 경우, 최신 NVIDIA Transformer Engine(TE) 하드웨어를 활용하는 더 효율적인 행렬 연산을 사용할 수 있습니다. 이 하드웨어는 트랜스포머 아키텍처를 사용하는 현대적 LLMs에 최적화되어 있습니다. 다른 경우, "병렬 처리가 쉬운" 추론 작업 부하에 대해 더 높은 수준의 병렬성을 구성함으로써 소프트웨어 프레임워크를 개선할 수 있습니다. 또 다른 경우에는 더 나은 메모리 관리를 구현하고 필요한 GPU 연산량 대비 GPU RAM으로 이동하는 메모리 양을 줄여 트랜스포머의 어텐션 알고리즘을 개선할 수 있습니다.

사소한 코드 수정조차도 큰 성과를 낼 수 있습니다. 예를 들어, Python으로 작성된 데이터 전처리 단계가 전체 훈련 파이프라인을 지연시키고 있을 수 있습니다. 해당 코드를 C++로 재구현하거나, CPU와 GPU 모두에 배열 연산 워크로드를 분산할 수 있는 NumPy 대체품인 NVIDIA cuPyNumeric을 사용해 이 병목 현상을 제거할 수 있습니다.

분산 훈련 및 추론 확장

소규모 연구 워크로드 를 초대형 클러스터의 대규모 생산 워크로드로 확장할 때, 8개 GPU에서 80,000개 GPU로 전환하더라도 시스템이 최소한의 오버헤드와 효율성 손실로 확장되도록 보장해야 합니다. 이를 위해서는 훈련 과정에서 흔히 사용되는 all-reduce와 같은 분산 집합 연산을 위해 NVIDIA Collective Communications Library(NCCL, '니켈'로 발음)를 활용한 통신 최적화가 필요합니다. 또한 NVIDIA 추론 전송 라이브러리(NIXL)는 분산 추론을 위해 GPU 메모리와 스토리지 계층 간 고처리량, 저지연, 지점 간 데이터 이동을 제공합니다. 이 통신은 단일 노드 내 GPU 간 또는 수천 개의 노드 간에 이루어질 수 있습니다. 또한 모델 훈련 및 추론 과정에서 광범위하게 사용되는 올-리듀스(all-reduce), 올-투-올(all-to-all), 올-개더(all-gather)와 같은 통신 및 집단 집계 작업을 최적화합니다.

데이터, 텐서, 파이프라인 병렬화를 활용하여 노드 간에 데이터를 지능적으로 배치할 수 있습니다. 또는 모델이 너무 커서 단일 GPU에 수용되지 않는 경우 텐서 병렬화나 파이프라인 병렬화를 사용하도록 워크로드를 재설계해야 할 수도 있습

니다. *전문가 혼합* (MoE) *모델*을 사용하고 있다면 전문가 병렬화의 이점을 활용할 수 있습니다.

자원 효율적 관리

모델 이 CPU 코어, GPU 메모리, 상호 연결 대역폭, 스토리지 I/O와 같은 리소스를 활용하는 방식을 최적화하는 것이 중요합니다. 여기에는 GPU에 데이터를 최대 속도로 공급하고, 스레드를 특정 CPU 코어에 고정하며, 컨텍스트 전환 오버헤드를 줄이고, 메모리 사용을 조정하며, 대규모 모델로 훈련 및 추론 시 GPU에서 메모리 부족(OOM) 오류를 방지하는 등 다양한 노력이 포함될 수 있습니다. GPU 가상화(예: NVIDIA의 멀티 인스턴스 GPU[MIG])와 같은 기술은 작업에 전체 GPU 성능이 필요하지 않을 때 GPU 리소스를 분할하여 전반적인 활용도를 높일 수 있습니다.

팀 간 협업

팀 간 협업은 AI 시스템 성능 엔지니어에게 절대적으로 중요합니다. 연구원, 데이터 과학자, 애플리케이션 개발자뿐만 아니라 네트워킹 및 스토리지를 포함한 인프라 팀과도 긴밀히 협력하는 것이 중요합니다.

성능 개선을 위해 모델 코드 수정이 필요할 수 있으며, 이는 연구진과의 협조를 필요로 합니다. 또는 효율성 향상을 위해 새로운 GPU 드라이버를 배포해야 할 수도 있는데, 이 경우 인프라 팀의 협력이 필요합니다.

성능 개선 작업은 종종 여러 팀에 걸쳐 수행됩니다. 예를 들어 버그 수정 및 효율성 향상을 위한 CUDA 드라이버 또는 CUDA 버전 업데이트에는 DevOps, 인프라, 지원 팀과의 세심한 협조가 필요합니다. 또한 성능 향상을 위한 모델 코드 개선 작업에는 연구진과의 긴밀한 협업이 요구됩니다. 성능 엔지니어는 이러한 다학제적 영역의 교차점에 위치하며 AI, 컴퓨터 과학, 시스템 엔지니어링의 언어를 구사합니다.

투명성과 재현성

성능 엔지니어링에서는 모든 것을 측정하고 추측이 아닌 데이터를 신뢰하는 것이 매우 중요합니다. 작업을 공개함으로써 다른 사람들이 여러분의 연구 결과를 배우고 재현하며 발전시킬 수 있습니다.

DeepSeek의 이야기 에서 주목할 만한 점은 인프라 최적화 방식을 공개적으로 공유했다는 점입니다. 2025년 2월 DeepSeek 오픈소스 주간 동안 FlashMLA,

DeepGEMM, DeepEP, 전문가 병렬 로드 밸런서 (EPLB), DualPipe, Fire-Flyer 파일 시스템 (3FS) 등 일련의 오픈소스 GitHub 저장소를 공개했습니다. 각 프로젝트는 실제 운영 환경에서 검증되었으며 하드웨어 성능을 극대화하는 데 목적을 두었습니다. 이 프로젝트들은 DeepSeek-V3 기술 보고서에 상세히 기술되어 있습니다.

FlashMLA는 CUDA C++로 작성된 최적화된 신경망 모델( ) 어텐션 커널입니다. DeepGEMM은 FP8 최적화 행렬 곱셈 라이브러리로, 보고에 따르면 고밀도 및 저밀도 연산 모두에서 다수 벤더 커널을 능가하는 성능을 보입니다. DeepEP(Deep Experts Parallelism)는 전문가 혼합(MoE) 모델을 위한 고도로 튜닝된 통신 라이브러리입니다. EPLB는 과부하 전문가들을 복제하여 추가 부하를 처리하는 중복 전문가 전략을 구현합니다. DualPipe는 전방/후방 계산 및 통신 단계를 중첩시켜 파이프라인 버블을 줄이는 양방향 파이프라인 병렬화 알고리즘입니다. 그리고 3FS는 고성능 분산 파일 시스템으로, AI 시스템에서 최대 성능을 끌어내기 위해서는 파일 시스템을 포함한 모든 계층이 최적화되어야 함을 상기시켜 줍니다.

DeepSeek는 2025년 2월 '오픈소스 위크' 기간 동안 GitHub 에 이 프로젝트들을 오픈소스화함으로써, 타인이 결과를 재현할 수 있도록 하여 주장의 신뢰성을 입증했을 뿐만 아니라 커뮤니티에 기여했습니다. 이러한 투명성은 다른 개발자들이 DeepEP/DualPipe 파이프라인 병렬화를 통한 통신 중첩이나 3FS를 통한 NVMe SSD/RDMA 대역폭 포화 등 그들의 방법론을 벤치마킹하고 재현하며 배울 수 있게 합니다.

DeepSeek의 오픈 인프라 인덱스 같은 공개 노력은 에 귀중한 기준점과 도구를 제공합니다. 다양한 AI 하드웨어 구성에 대한 실제 성능 측정값을 제공하며, 동등한 조건 비교와 재현성을 장려합니다. 마찬가지로 MLPerf 공개 벤치마크 제품군은 하드웨어 및 소프트웨어 구성 전반에 걸쳐 훈련 및 추론 성능을 재현 가능하게 비교할 수 있는 표준을 제공합니다.

MLPerf와 같은 산업 벤치마크는 하드웨어 세대 간 이러한 종류의 코디자인 최적화를 정량화해 왔습니다. MLPerf Training v5.0(2025)에서 Blackwell 기반 NVIDIA GB200 NVL72 시스템은 동등한 Hopper 시스템 대비 GPU당 최대 2.6배 높은 훈련 처리량을 기록했으며, 이는 그림 1-2에 표시되어 있습니다. MLPerf Inference v5.0(2025)에서 Blackwell NVL72는 GPU당 성능 향상과 훨씬 더 큰 NVLink 도메인 덕분에 동급 Hopper 클러스터 대비 GPU당 약 3.4배 높은 추론 처리량을 달성했습니다. 이 결과는 그림 1-3에 표시되어 있습니다.

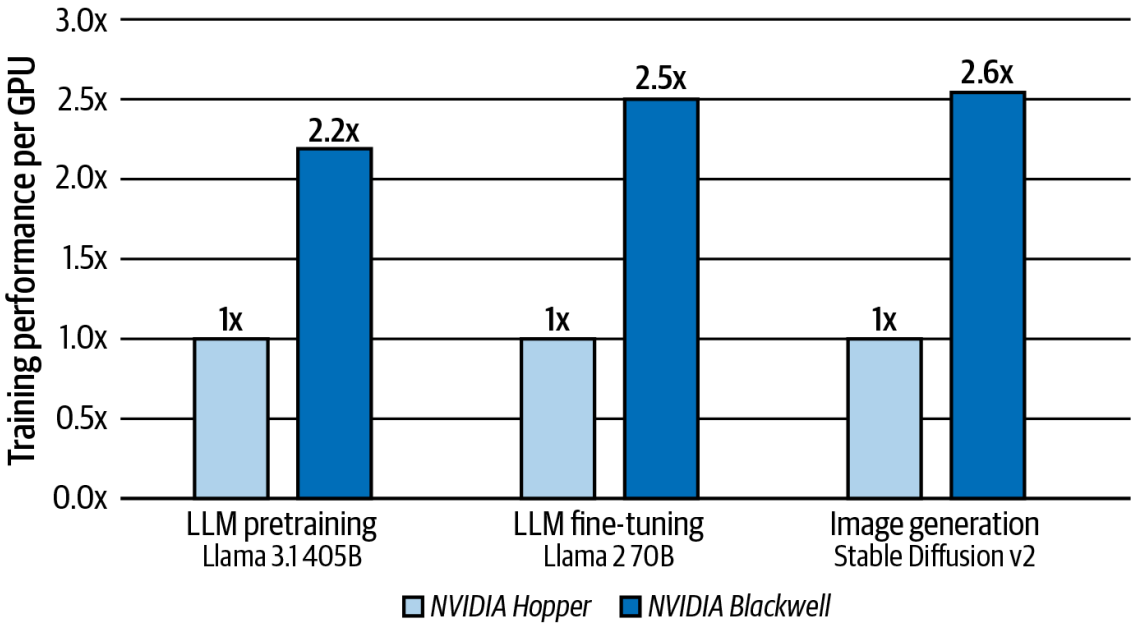

그림 1-2. NVIDIA GB200 NVL72 MLPerf Training v5.0의 GPU당 처리량 향상률 (동등한 NVIDIA Hopper 클러스터 대비) (출처: https://oreil.ly/Ao1l8)

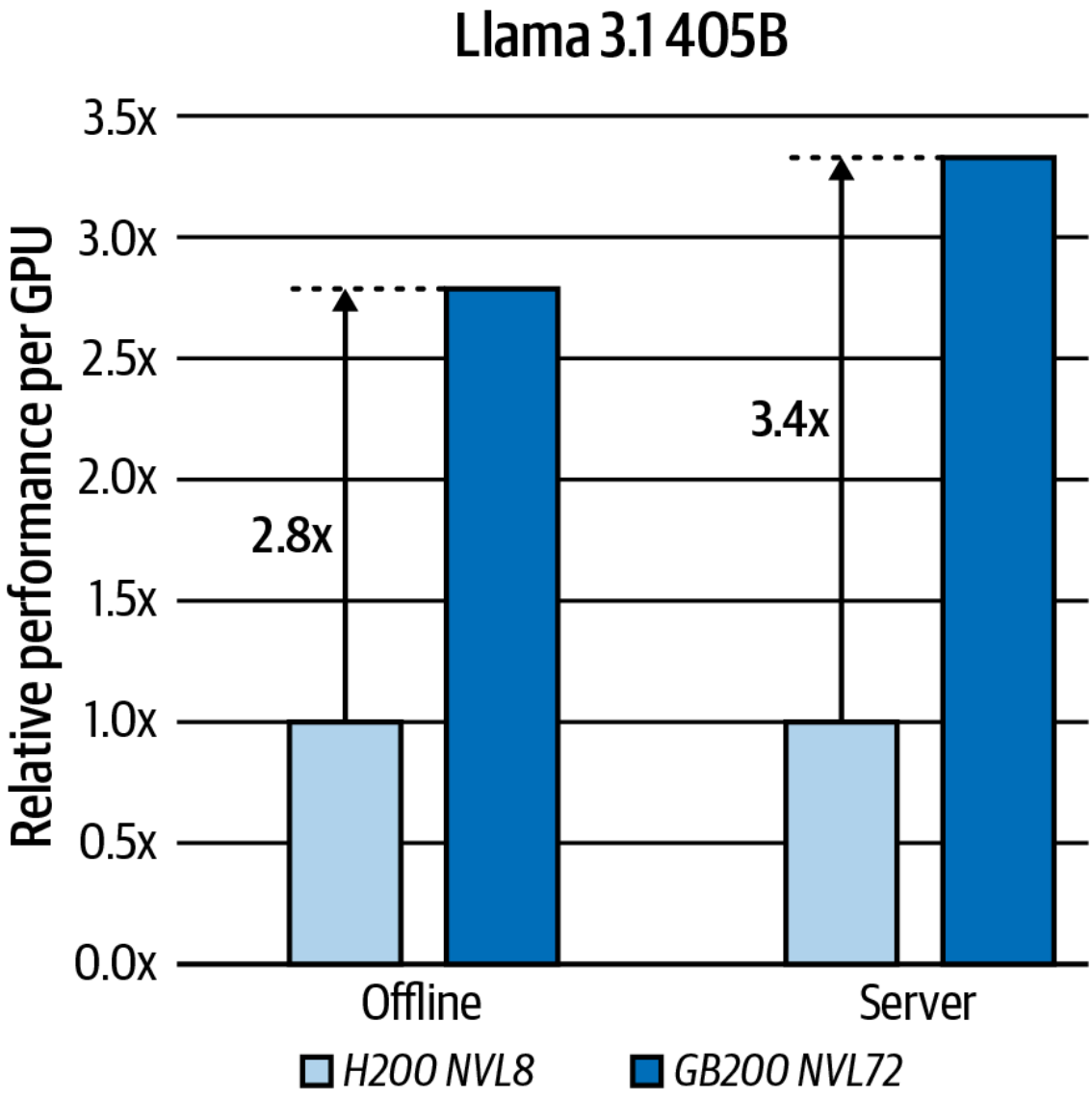

그림 1-3. NVIDIA GB200 NVL72 MLPerf 추론 v5.0에서 동등한 NVIDIA Hopper 클러스터 대비 GPU당 처리량 향상률 (출처: https://oreil.ly/V-jze)

후속 장에서는 다양한 성능 튜닝 개념을 뒷받침하기 위해 이러한 공개 벤치마크 중 일부를 참조할 것입니다. 예를 들어 GPU 커널 최적화를 논의할 때, NVIDIA

GPU에서 맞춤형 커널이 어떻게 거의 최대 메모리 대역폭 활용도를 달성했는지 보여주는 DeepSeek의 공개 자질을 참조할 것입니다.

MLPerf 결과를 인용할 때는, MLPerf가 GPU당 결과가 플랫폼 간 주요 지표가 아니라고 경고한다는 점을 기억하십시오. 비교의 올바른 기준으로 시스템 수준의 종단 간 처리량을 사용하십시오. GPU당 수치는 구성 요소 수준 지표로만 고려하십시오.

실험의 투명성과 재현성은 AI 성능 엔지니어링 분야를 발전시키는 데 핵심적입니다. "우리는 X를 했더니 속도가 빨라진 것 같다"는 식의 일화적인 "느낌" 최적화의 함정에 빠지기 쉽습니다. 대신 저는 가설을 수립하고, 재현 가능한 벤치마크로 결과를 측정하며, 결과를 개선하기 위해 조정하고, 벤치마크를 재실행하며, 모든 단계의 결과를 공유하는 엄격한 과학적 접근법을 주장합니다.

DeepSeek, 미국 수출 하드웨어 제한 속에서도 중국에서 약 6,800억 매개변수 모델까지 확장

때로는 시스템 성능 혁신이 필요성에서 비롯되기도 합니다. 앞서 언급했듯이, DeepSeek는 미국 수출 제한으로 인해 NVIDIA의 H800 GPU만 사용해야 하는 제약에 직면했습니다. H800은 Hopper GPU의 수출 규정 준수 변형 모델입니다. H100과 비교할 때, H800은 NVLink 상호 연결 대역폭과 FP64 성능을 줄이면서도 HBM 용량과 대역폭은 대체로 유사하게 유지합니다.

배경을 설명하자면, NVIDIA H100은 GPU당 약 900GB/s의 NVLink 상호 연결 대역폭을 제공하는 반면, H800은 GPU당 약 400GB/s의 대역폭을 제공합니다. 이로 인해 GPU 간 데이터 전송 속도가 느려지고 궁극적으로 다중 GPU 확장성이 제한됩니다. 또한 H100이 3.35TB/s의 메모리 대역폭을 제공하는 반면, H800의 제한된 처리량은 데이터 전송 속도를 현저히 저하시킵니다. 이는 분산 훈련 작업의 병목 현상을 유발하고 확장성 효율성을 제한할 위험이 있습니다.

DeepSeek는 이처럼 극도로 제한된 환경에서 DeepSeek-V3라는 약 6800억 개의 매개변수를 가진 대규모 MoE 언어 모델을 훈련하기 시작했습니다. 이 MoE 모델은 입력 토큰당 약 370억 개의 활성 매개변수만을 사용하며, 약 6800억 개 전체를 동시에 사용하지 않습니다.

이러한 아키텍처를 통해 특정 시점에 모델의 일부만 활성화됩니다. 이는 컴팩트한 H800 설정에서도 계산 부하를 관리하는 데 도움이 됩니다. 구체적으로, 각 토큰에 대해 DeepSeek-V3는 1명의 공유 전문가와 라우터가 선택한 8명의 전문가(256명의 전문가 중)를 사용하며, 그림 1-4에 표시된 대로 토큰당 총 9명의 활성 전문가를 활용합니다.

MoE에 대해서는 후속 장에서 더 자세히 다룰 예정입니다. 다만 환경적 제약을 극복하기 위해 DeepSeek는 H800의 본질적 약점을 가리기 위해 계산과 통신을 정교하게 중첩시키는 새로운 DualPipe 병렬 처리 알고리즘을 구현했다는 점만 알아두시기 바랍니다.

기본 NCCL 통신 콜렉티브 일부를 우회하는 맞춤형 CUDA 커널을 설계함으로써, DeepSeek는 진행 중인 계산과 병행하여 데이터 전송을 조정할 수 있었습니다. 이를 통해 GPU 간 연결 대역폭이 감소했음에도 GPU를 효율적으로 활용할 수 있었습니다. 이러한 통신/계산 중첩은 이 책의 나머지 부분에서 반복적으로 다루게 될 주제입니다.

이 혁신적인 엔지니어링은 성과를 거두었습니다. DeepSeek-V3는 OpenAI, Meta, DeepMind 등 유사 규모의 프론티어 모델에 비해 극히 적은 GPU 시간(및 비용)으로 훈련을 완료했습니다. 이는 더 강력한 클러스터를 사용해 이 규모 모델을 훈련하는 데 필요하다고 여겨졌던 자원의 극히 일부에 불과합니다.

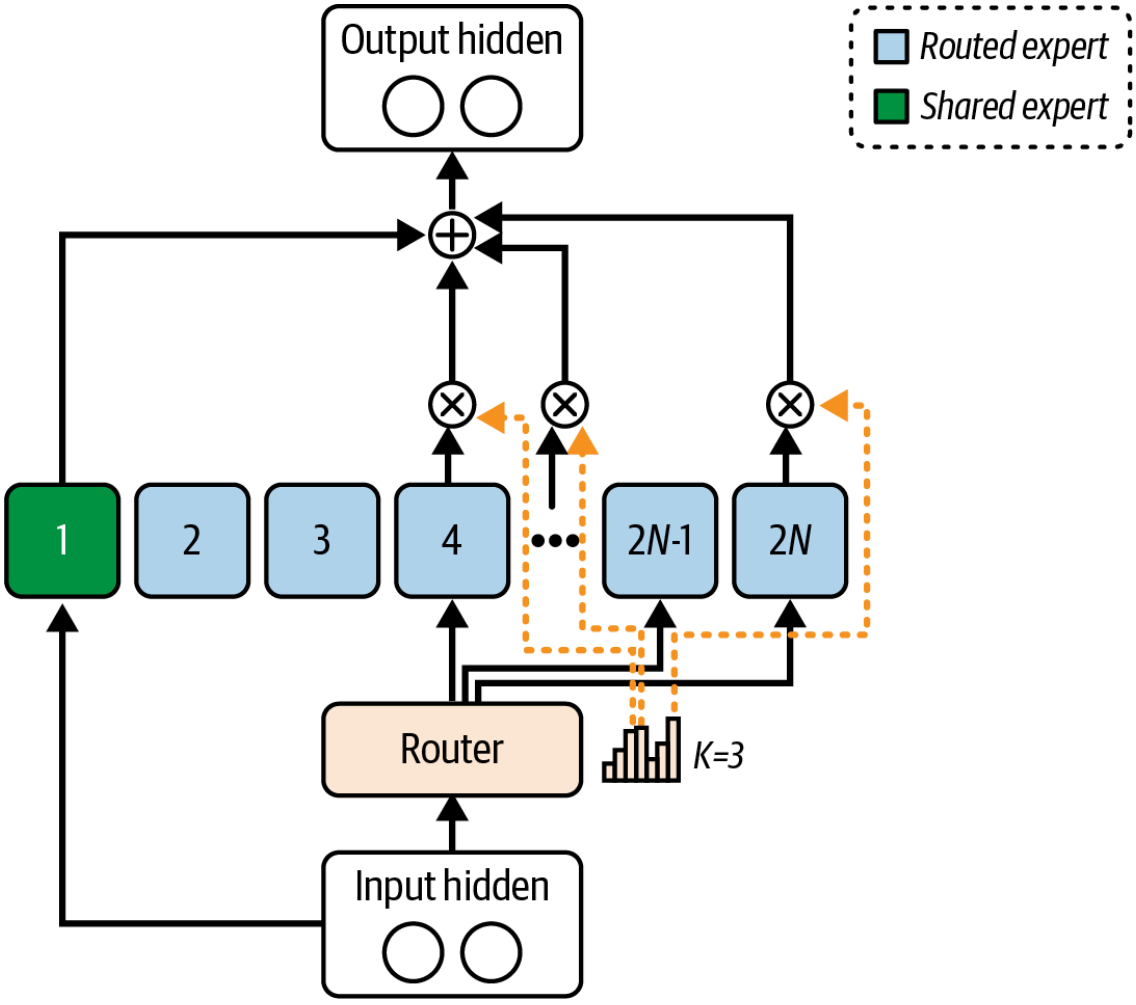

그림 1-4. DeepSeek-V3의 전문가 라우팅

DeepSeek 에 따르면 V3의 성능은 언어 이해, 독해력, 추론 등 여러 표준화된 벤치마크에서 GPT-4에 근접합니다. 일부 테스트에서는 GPT-4와 동등하거나 약간 우월한 결과를 보이기도 합니다. 이러한 비교는 업계 전반에서 사용되는 표준화된 테스트를 기반으로 했습니다. 이는 개방형 MoE 모델이 성능이 낮은 하드웨어를 사용함에도 불구하고 최고의 폐쇄형 모델과 경쟁할 수 있음을 시사합니다.

DeepSeek-V3 모델을 기반으로, 연구팀은 OpenAI의 o1 및 o3 추론 모델 시리즈와 유사하게 구축된 전용 추론 모델인 DeepSeek-R1을 개발했습니다. DeepSeek는 미세 조정을 위해 비용이 많이 드는 인간 피드백 루프에 크게 의존하는 대신, 최소한의 감독 학습 데이터를 사용하는 '콜드 스타트' 전략을 개척했습니다. 대신 강화 학습 기법을 활용하여 사고 과정 추론을 R1에 직접 내장했습니다. 이 접근 방식은 훈련 비용과 시간을 줄였으며, 스마트한 소프트웨어 및 알고리즘 설계가 하드웨어 병목 현상을 극복할 수 있음을 입증했습니다.

이 과정에서 얻은 교훈은 대규모의 희소 활성화 MoE 모델이 제한된 메모리 및 컴퓨팅 예산에서도 효과적으로 확장 가능하다는 점입니다. DeepSeek의 노력에서 입증된 막대한 투자 반환률(ROI)이 보여주듯, 새로운 훈련 일정과 저수준 통신/계산 중첩 최적화를 통해 하드웨어 한계를 극복할 수 있습니다.

DeepSeek의 비전통적 접근법은 막대한 효율성을 가져왔으며 훨씬 낮은 훈련 비용과 시간으로 일련의 강력한 모델을 생성함으로써 ROI가 명확해졌습니다. 통신 대역폭 감소라는 제약 조건 하에서도 NVIDIA H800 GPU의 성능을 극한까지 끌어냄으로써, 팀은 수백만 달러를 절감하면서도 GPT-4 수준의 모델 성능을 달성했습니다. 또한 DeepSeek는 추론 단계인 R1의 미세 조정 과정에서 인간이 라벨링한 데이터의 양을 크게 줄여 추가 비용을 절감했습니다.

요약하자면, 스마트한 소프트웨어와 알고리즘 설계가 무차별적인 하드웨어 한계를 극복했습니다. 이를 통해 DeepSeek는 제한된 비용과 하드웨어 예산으로 대규모 AI 모델을 개발할 수 있었습니다.

100조 매개변수 모델을 향하여

100조 매개변수 모델 은 AI 분야의 야심찬 목표이며, 종종 인간 신피질의 추정된 100조 개의 시냅스 연결과 비교됩니다. 각 시냅스 연결은 모델 매개변수 하나에 해당합니다. 이론적으로 이 규모의 모델 구현은 가능하지만 , 엄청난 자원과 비용이 필요합니다. 순전히 물리적 역량으로 100조 매개변수 모델까지 확장하는 것은 절대적으로 부유한 기관을 제외하고는 비현실적이다.

대략적인 규모로, 약 29조 개의 토큰 으로 훈련된 고밀도 100조 매개변수 모델은 1.2×10^{29} 플로팅 포인트 연산(FLOPS)이 필요합니다. 토큰 수가 증가하면 이 수치는 선형적으로 증가하지만, 스파스(MoE) 모델을 사용하면 효과적인 연산량을 줄일 수 있습니다. 그러나 스파스성을 적용하더라도, 단순한 무차별적 접근만으로는 초대형 모델의 연산 요구를 충족시키기에 부족함을 보여준다. 100조 매개변수 훈련을 합리적인 시간 내에 가능하게 하려면 새로운 접근법이 필요하다. 이러한 규모를 달성하려면 기존 방법의 단순한 확장보다는 하드웨어, 소프트웨어, 알고리즘의 통합 설계에서 획기적인 효율성이 요구된다.

이 책에서 논의된 최적화 기법은 소규모 모델과 클러스터에도 적용 가능하지만, 저는 계속해서 100조 매개변수 모델을 예시로 삼아 확장성 문제에 단순히 하드웨어를 투입하는 것만으로는 해결할 수 없다는 점을 강조할 것입니다. 컴퓨팅, 네트워킹, 메모리, 스토리지를 포함한 하드웨어와 함께 설계된 지능적인 소프트웨어 및 알고리즘 혁신이 필요합니다.

훈련 비용의 폭발적 증가는 성능 향상, 비용 절감, 제한된 컴퓨팅 자원·전력 제약·예산 하에서 극한 규모 AI 구현을 가능케 할 새로운 AI 시스템 및 소프트웨어 엔지니어링 기법 탐구를 촉진하고 있습니다. 연구자들은 효과적 컴퓨팅 요구량을 줄일 혁신적 기법을 지속적으로 모색 중입니다.

주목받는 아이디어 중 하나는 스파스성 을 활용하는 것이며, 특히 MoE(모델 온 디맨드) 모델을 사용하는 것입니다. MoE와 같은 스파스 모델은 OpenAI가 대중화시킨 일반적인 GPT 시리즈 대규모 언어 모델(LLMs)과 같은 전통적인 덴스 모델과 대조됩니다. 실제로 OpenAI의 독점적인 GPT 시리즈 및 o-시리즈 추론 모델과 같은 일부 최첨단 모델이 MoE 아키텍처를 기반으로 한다는 공개적인 추측이 있습니다.

MoE와 같은 스파스 모델은 각 입력 토큰에 대해 모델의 일부만 활성화합니다. 수많은 내부 '전문가' 중 일부만을 통해 각 입력 토큰을 처리함으로써, 총 매개변수가 증가해도 토큰당 FLOPS는 대략 일정하게 유지됩니다. 이러한 스파스 모델은 수조 개 매개변수 규모의 모델로 확장해도 계산 비용이 비례하여 폭발적으로 증가하지 않음을 입증합니다. 또한 추론 시 활성 매개변수 수가 적어 MoE 모델은 밀집형 모델 대비 요청-응답 지연 시간이 현저히 낮습니다. 이는 100조 매개변수 규모 모델의 훈련 및 서비스에 핵심적인 통찰입니다.

DeepSeek-V3(기본 모델)과 -R1(강화학습 기반 추론 변형)은 MoE 효율성의 훌륭한 사례입니다. 이 모델들은 총 약 6,800억 개의 매개변수를 포함하며, 입력 토큰당 약 370억 개의 활성 매개변수를 가집니다. DeepSeek의 기술 보고서와 공개 자료에 따르면, 토큰당 256명의 전문가 중 1명의 공유 전문가와 8명의 선택된 전문가를 활용하는 MoE 구조를 채택하여 토큰당 약 9명의 활성 전문가와 약 370억 개의 활성 매개변수를 구현합니다. 이로 인해 DeepSeek-V3 및 -R1은 유사한 규모의 밀집형 대규모 언어 모델보다 훨씬 더 높은 자원 효율성을 보입니다. 또 다른 예로 2021년 구글의 Switch Transformer MoE가 있습니다. 이 1.6조 매개변수 MoE 모델은 밀집형 모델과 동일한 정확도를 달성하면서도 계산량의 극히 일부만 소모했습니다. 동급 밀집형 접근법 대비 7배 빠른 속도로 훈련되었습니다.

막대한 컴퓨팅 요구사항 외에도 메모리 역시 주요 병목 현상입니다. 예를 들어, 100조 매개변수 모델의 경우 각 매개변수를 16비트(2바이트) 정밀도로 저장할 때 모델 로딩에 약 182TB의 GPU 메모리가 필요합니다(182TB = 100조 매개변수 × 가중치당 16비트 × 바이트당 8비트). 이는 단일 NVIDIA Blackwell B200 GPU의 192GB(실사용 가능 180GB) GPU RAM에 비해 3차원(1,000배)에 달하는 규모입니다.

100조 개의 모델 가중치 를 단순히 로드하기 위해서는 약 1,000개의 Blackwell B200 GPU(각 192[실사용 가능 180]GB) 또는 700개의 Blackwell Ultra B300 GPU(각 약 288GB)가 필요합니다. 이 추정은 모델 로딩만을 위한 것으로, 활성화 메모리, 최적화기 상태, 추론 입력 데이터는 포함되지 않습니다. 이 요소들은 필요한 총 메모리를 더욱 증가시킬 것입니다.

참고로, 일반적인 B200 GPU 컴퓨팅 노드에는 B200 GPU가 8개만 장착됩니다. 이를 사용하려면 B200으로 모델을 로드하는 데만 약 125개의 GPU 노드가 필요하며, Ultra B300 GPU 컴퓨팅 노드 클러스터(노드당 B300 GPU 8개)로 모델을 로드하려면 약 86개의 GPU 노드가 필요합니다.

또한 훈련 데이터 로딩 도 극도로 어려워집니다. 이처럼 많은 GPU를 모두 가동 상태로 유지할 만큼 빠르게 데이터를 공급하는 것은 결코 쉬운 일이 아니기 때문입니다. 특히 100조 매개변수 모델이 1,000개의 B200 GPU 또는 700개의 B300 GPU에 분산될 경우 GPU 간 통신 오버헤드가 크게 증가합니다. 단일 모델 훈련에는 수백만 GPU-시간과 메가와트시 단위의 에너지가 소모될 수 있습니다. 이는 100조 매개변수 모델을 훈련하고 서비스하기에 충분한 규모로 확장하는 데 드는 막대한 비용이자 에너지 소비입니다.

100조 매개변수 AI 시대는 이 규모에서 훈련과 배포를 실현 가능하게 하기 위해 시스템 설계를 완전히 재고하도록 요구할 것입니다. 하드웨어, 알고리즘, 소프트웨어 모두가 이 새로운 영역에 대응하기 위해 함께 진화해야 합니다.

NVIDIA의 "랙 하나에 담긴 AI 슈퍼컴퓨터"

초대규모 컴퓨팅의 과제를 해결하기 위해 엔비디아( )는 특히 1조 개 규모의 매개변수를 가진 워크로드를 목표로 하는 새로운 등급의 AI 슈퍼컴퓨터를 구축했습니다. 대표적인 예로는 2024년 출시 예정인 NVIDIA Grace Blackwell GB200 NVL72와 GB300 NVL72 Ultra가 있습니다. GB300 NVL72 Ultra는 GPU당 288GB HBM3e를 탑재한 Blackwell Ultra GPU를 사용하며, 72개의 GPU로 구성된 NVLink 도메인을 유지하면서 초당 약 130테라바이트의 총 대역폭을 제공합니다.

베라 루빈 VR200(2026년)과 파인만(2028년) 시스템은 엑사스케일 슈퍼컴퓨터를 단일 데이터센터 랙에 집약하는 이 추세를 이어갑니다. 실제로 엔비디아는 NVL72와 같은 이러한 랙 시스템을 "랙 내 AI 슈퍼컴퓨터"라고 부르는데, 그럴 만한 이유가 있습니다.

각 GB200/GB300 NVL72 랙 은 NVLink를 통해 NVSwitch와 연결된 36개의 Grace Blackwell을 통합하여 랙 스케일 스위칭 패브릭을 제공합니다. 각 Grace Blackwell 슈퍼칩은 NVIDIA Grace CPU(72개 CPU 코어) 1개와 NVIDIA Blackwell GPU 2개로 구성되어 총 36개의 Grace CPU와 72개의 Blackwell GPU를 탑재합니다. 따라서 "NVL72"라는 명칭에 "72"가 포함된 것입니다.

아직 눈치채지 못하셨다면, NVL72의 NVL은 *NVLink*를 의미합니다. 이는 본 시스템의 GPU들이 NVLink로 상호 연결되었음을 상기시켜 줍니다. 혹시라도 잊을 경우를 대비해서요!

각 Grace Blackwell 보드는 랙 내 NVLink 스위치 네트워크인 NVSwitch를 통해 다른 보드와 연결됩니다. 이를 통해 72개의 GPU 모두가 GPU당 양방향 1.8TB/s(18×100GB/s 링크)의 NVLink 5 최대 대역폭으로 상호 통신할 수 있습니다. 랙 전체에서 NVLink 스위치 시스템은 하나의 NVL72 도메인 내에서 GPU 간 약 130TB/s의 총 대역폭을 제공합니다.

실질적으로 NVL72의 내부 패브릭은 72개의 Grace Blackwell GPU를 하나의 고속 클러스터로 통합합니다. 따라서 PyTorch와 같은 훈련 프레임워크나 추론을 위한 vLLM은 효율적인 데이터 텐서 파이프라인과 전문적인 병렬 처리를 위해 단일 NVLink 도메인을 활용할 수 있습니다. CUDA 통합 메모리는 필요 시 NVLink를 통해 페이지 이동 또는 원격 접근이 가능합니다. 그러나 원격 메모리는 비균일한 지연 시간과 대역폭을 가지므로 별도로 처리해야 합니다. Grace와 Blackwell 간 관리형 할당에 의존할 때는 페이지 결함 정지를 줄이기 위해 `cudaMemPrefetchAsync` 및 `cudaMemAdvise` 를 사용한 명시적 프리페치를 선호하십시오.

이 슈퍼컴퓨터의 컴퓨팅, 메모리 및 상호 연결 하드웨어 세부 사항은 후속 장에서 더 자세히 설명됩니다. 현재는 현대 생성형 AI 모델의 맥락에서 이 AI 슈퍼컴퓨터의 전체 성능 사양을 종합적으로 분석해 보겠습니다.

전체 GB200 NVL72 랙은 이론적으로 2:1 구조적 스파스성을 적용한 FP4에서 약 1.44 엑사플롭스(exaFLOPS), FP8에서 약 720 페타플롭스(petaFLOPS)에 도달할 수 있습니다. 72개의 GPU에 걸쳐 약 13.5TB(13,824GB = GPU당 192GB × 72개 GPU)의 HBM3e를 제공하며, 동일한 NVLink 도메인 내 Grace CPU 메모리를 포함하면 총 약 30TB에 달합니다.

NVLink는 72개의 GPU에 걸쳐 풀링된 액세스를 제공하며, CUDA 통합 메모리는 NVLink를 통해 페이지를 마이그레이션하거나 원격으로 액세스할 수 있지만, 원격 액세스는 성능이 다르며 랙은 단일 균일 메모리 장치처럼 작동하지 않습니다.

요약하면, GB200 NVL72는 자체 완결형 72 GPU 1.44 엑사플롭스 30TB 메모리 시스템으로, 벤더 구성 및 냉각 방식에 따라 약 120~132kW의 랙 전력을 소비합니다. 이는 단일 랙 내에서 수조 개 매개변수 모델을 훈련 및 서비스할 수 있는 진정한 AI 슈퍼컴퓨터입니다.

이러한 랙을 결합하여 초대형 클러스터를 구성하면 수조 개 이상의 매개변수를 가진 대규모 모델을 지원할 수 있습니다. 더 나은 점은 Amazon Web Services(AWS), Google Cloud Platform(GCP), Microsoft Azure, CoreWeave, Lambda Labs 등 선호하는 클라우드 공급자를 통해 몇 번의 클릭(그리고 상당한 비용!)만으로 이러한 랙 및 랙 클러스터를 프로비저닝할 수 있다는 것입니다.

이 책은 NVIDIA의 Grace Blackwell 세대 칩에 집중하지만, 논의된 최적화 원칙은 이전 여러 세대 NVIDIA 하드웨어에서 도출된 것입니다. 이러한 최적화는 Vera Rubin(2026), Feynman(2028)을 비롯한 향후 출시될 수많은 NVIDIA 칩 세대에 계속 적용되고 진화할 것입니다. 이 로드맵은 새로운 GPU 세대마다 성능, 메모리, 통합성을 두 배로 늘리는 패턴을 이어갑니다.

이 책에서는 컴퓨팅, 메모리, 네트워킹, 스토리지 분야의 각 세대별 혁신이 어떻게 초스케일 클러스터, 수조 개 이상의 매개변수를 가진 모델, 고처리량 훈련 작업, 극저지연 모델 추론 서버 형태로 AI 확장성에 기여하는지 알아볼 것입니다. 이러한 혁신은 다음 섹션에서 논의될 기계적 공감과 하드웨어-소프트웨어 공동 설계 원칙을 구현하는 하드웨어 인식 알고리즘에 의해 추진됩니다.

기계적 공감: 하드웨어-소프트웨어 공동 설계

*기계적 공감(Mechanical sympathy)*은 소프트웨어 엔지니어링 마틴 톰슨(Martin Thompson)이 영국 F1 챔피언 재키 스튜어트(Jackie Stewart)가 자신의 차량 기계적 세부 사항을 깊이 이해했던 점에 비유하며 처음 제안한 용어입니다. 컴퓨팅 분야에서는 실행되는 하드웨어를 깊이 인식하는 소프트웨어 작성을 의미합니다. AI 맥락에서는 성능 극대화를 위해 하드웨어 역량과 손발을 맞추어 알고리즘을 공동 설계하는 것을 뜻합니다.

실제 경험에 따르면 GPU 커널이나 메모리 접근 패턴의 사소한 조정조차도 비례하지 않는 성과를 낼 수 있다. 대표적인 사례가 하드웨어 인식 방식으로 트랜스포머 어텐션 메커니즘을 재구현한 혁신적 알고리즘 플래시어텐션(FlashAttention)이다.

FlashAttention은 GPU 연산을 '타일' 형태로 분할하여 GPU 메모리에 발행되는 읽기/쓰기 횟수를 최소화합니다. 이로 인해 메모리 이동이 크게 줄어들고 어텐션 계산 속도가 향상됩니다. 기본 트랜스포머 어텐션 메커니즘/알고리즘을

FlashAttention으로 대체하면 긴 시퀀스에서 훈련 및 추론 속도가 2~4배 빨라지며 전체 메모리 사용량도 감소합니다.

이러한 변화로 인해 과거 주요 병목 현상이었던 부분(어텐션)이 전체 실행 시간의 극히 일부로 축소되었습니다. FlashAttention은 모델이 더 긴 시퀀스를 더 빠르고 효율적으로 처리할 수 있게 해준 덕분에 거의 하룻밤 사이에 많은 라이브러리에서 기본값으로 채택되었습니다. FlashAttention 이후 DeepSeek의 다중 헤드 잠재 어텐션(MLA)을 비롯한 수많은 새로운 어텐션 알고리즘이 등장했습니다.

DeepSeek의 MLA 알고리즘은 NVIDIA GPU 커널로 구현되어 2025년 오픈소스화되었으며, 하드웨어-소프트웨어 공동 설계(기계적 공감)의 또 다른 사례입니다. FlashAttention과 유사하게, MLA는 NVIDIA의 메모리 계층 구조와 전용 GPU '텐서 코어'를 더 잘 활용하기 위해 어텐션 연산을 재구성합니다. 이러한 최적화를 통해 MLA는 제약된 H800 GPU 아키텍처를 활용하여 훨씬 낮은 비용으로 더 높은 처리량을 달성했으며, 동일한 H800 시스템에서 FlashAttention의 성능을 뛰어넘었습니다.

이 책 전체는 사실상 기계적 공감에 대한 연구라 할 수 있습니다. DeepSeek 사례에서처럼 새로운 하드웨어 기능 또는 하드웨어 제약이 독창적인 소프트웨어 및 알고리즘 기법을 고무하는 수많은 사례를 살펴볼 것입니다. 반대로 새로운 소프트웨어 알고리즘이 새로운 하드웨어 혁신을 촉진하는 사례도 살펴볼 것입니다.

예를 들어, 트랜스포머 모델 과 정밀도 감소 양자화(예: FP8/FP4)의 부상은 NVIDIA로 하여금 트랜스포머 엔진과 전용 정밀도 감소 텐서 코어 같은 특수 하드웨어를 추가하게 하여 행렬 연산 유닛의 속도를 높였습니다. 이러한 하드웨어 혁신은 연구자들이 새로운 수치 최적화기 및 신경망 아키텍처를 탐구할 수 있게 합니다. 이는 다시 하드웨어 설계자들을 더욱 발전시키고, 이는 또 다른 새로운 알고리즘을 가능케 하는 등 선순환 구조를 이룹니다!

현대 GPU는 FP4 지원 및 마이크로스케일링 기능을 갖춘 최신 트랜스포머 엔진( )을 사용합니다. 최신 세대 NVLink와 결합된 이러한 기능들은 주로 더 빠른 텐서 코어 연산, 더 높은 메모리 대역폭, 개선된 특수 기능 유닛(SFU)을 통해 어텐션 처리량을 증가시킵니다. 예를 들어, 지수 연산 유닛은 트랜스포머의 어텐션 알고리즘에서 집중적으로 사용되는 소프트맥스 연산을 가속화하기 위해 특별히 설계되었으며, 이는 오늘날 LLM의 핵심 부분입니다. 트랜스포머의 대중화로 인해 소프트맥스 지연 시간은 트랜스포머 어텐션 메커니즘에 핵심적이라는 점을 고려할 때 기존 GPU에서 병목 현상이 되었습니다.

현대 GPU를 통해 특수 함수 처리량( )과 고속 수학 파이프라인을 개선함으로써, NVIDIA는 기계적 공감과 하드웨어-소프트웨어-알고리즘 공동 설계의 가장 순수하고 탁월한 형태를 보여줍니다. 그들은 연구자 및 실무자와 직접 협력하여 자사 하드웨어에서 실행되는 현대 LLM 알고리즘(예: 어텐션)의 병목 현상을 해결하고 완화합니다.

GPU와 AI 알고리즘이 함께 진화하는 이 긴밀한 상호작용은 AI의 기계적 공감의 핵심입니다. 이러한 공동 설계 혁신은 하드웨어 기업(예: NVIDIA, ARM), AI 연구소(예: OpenAI, Anthropic), AI 시스템 성능 엔지니어(예: 저희!) 간의 긴밀한 협력을 통해서만 가능합니다.

"유효 처리량(Goodput)" 측정

수백, 수천, 수백만 개의 GPU 클러스터를 운영할 때 이론적 하드웨어 성능 중 실제로 유용한 작업을 수행하는 비중을 이해하는 것이 중요합니다. FLOPS나 장치 활용률 같은 전통적인 처리량 지표는 오해의 소지가 있을 정도로 높게 나타납니다. 대부분의 시간이 통신 지연, 유휴 상태의 계산, 실패한 작업 재시작 등에 소비될 가능성이 크기 때문입니다. 바로 여기서 "굿풋(goodput)" 개념이 등장합니다. 메타(Meta)가 2025년 논문에서"실효 훈련 시간 비율(effective training-time ratio)"로 설명한 바로 그 개념입니다.

NVIDIA는 이론적 하드웨어 최대 성능을 *'빛의 속도'라고* 명명합니다. 이는 NVIDIA 블로그, 문서, 웨비나, 컨퍼런스 발표에서 확인하실 수 있습니다.

간단히 말해, 굿풋은 모델 훈련이나 추론에 직접 기여하지 않는 모든 요소를 제외하고 단위 시간당 완료된 유용한 작업(처리된 토큰 수 또는 완료된 추론 요청 수)의 처리량을 측정합니다. 이는 생산적인 모델 훈련 또는 추론 관점에서 본 시스템의 종단 간 효율성이라고 할 수 있습니다. 이후 굿풋을 클러스터의 최대 가능 처리량으로 정규화하여 백분율 효율성 값을 산출할 수 있습니다.

예를 들어, 8개의 GPU를 가진 노드가 10초 동안 100,000개의 토큰을 처리할 수 있다고 가정해 보겠습니다. 이 경우 해당 노드의 굿풋은 초당 10,000 토큰입니다. 해당 노드의 각 GPU가 초당 1,500 토큰의 이론적 최대 처리량을 달성할 수 있고, 8개 GPU 전체로는 초당 12,000 토큰을 처리할 수 있다면, 노드의 효율성은 83.3%입니다(0.833 = 달성 처리량 10,000 / 최대 처리량 12,000).

메타의 AI 인프라 팀은 "신뢰성 재검토(Revisiting Reliability)" 논문에서 굿풋의 중요성을 강조했습니다. 이 논문은 *효과적 훈련 시간 비율 지표*를 소개하며, 헤드라인 활용률이 높더라도 리소스 분할 및 장애로 인한 사전 종료가 실제 훈련 시간을 어떻게 감소시키는지 보여줍니다. 본 논문에서 메타 팀은 사전 종료, 하드웨어 결함, 네트워크 혼잡이 실제 처리량을 어떻게 저하시키는지 분석합니다.

즉, 클러스터가 100% 활용되는 것처럼 보였지만 통신 지연, 비최적 병렬화, 데이터 지연, 장애 복구 등의 오버헤드로 인해 컴퓨팅 자원의 70~75%가 손실되었습니다. 또한 메타의 분석에 따르면 대규모 환경에서 작업 선점, 네트워크 핫스팟, 복구 불가능한 결함 등이 유효 처리량 손실의 주요 원인임이 밝혀졌습니다.

예를 들어, 이상적인 하드웨어에서 이론적으로 초당 1,000개 샘플을 처리할 수 있는 훈련 작업이 입력 파이프라인의 문제와 과도한 동기화로 인해 실제 훈련 처리량이 초당 300개 샘플에 그친다고 가정해 보십시오. 이 작업은 30%의 유효 처리량으로 실행되고 있다고 말할 수 있습니다. 나머지 70%의 용량은 사실상 낭비된 것입니다.

이러한 격차를 파악하고 해소하는 것이 우리 작업의 핵심입니다. 예를 들어, GPU가 스토리지에서 데이터 로딩을 기다리는 경우 캐싱이나 비동기 프리페치를 도입할 수 있습니다. 모델 훈련 과정의 기울기 동기화 단계에서 GPU가 유휴 상태라면, GPU 간 통신(예: 기울기 동기화)과 GPU 계산(예: 기울기 계산)을 중첩시키는 방안을 고려해야 합니다. 우리의 목표는 낭비되고 비효율적인 사이클을 유용한 작업으로 전환하는 것입니다.

이론적 성능과 실제 성능 간의 격차가 바로 AI 시스템 성능 엔지니어 역할의 가치 제안입니다. 우리의 사명은 하드웨어, 소프트웨어, 알고리즘을 포함한 스택의 모든 수준에서 비효율성을 해결하고 비용을 절감함으로써 이 '굿풋(goodput)' 수치를 최대한 높이는 것—이상적으로는 100%에 가깝게 증가시키는 것입니다 .

의 AI 시스템 성능 엔지니어에 대한 투자는 지속적으로 비용 대비 훨씬 높은 반환을 창출합니다. 클러스터 효율성을 20% 향상시키는 데 기여하는 성능 엔지니어를 생각해 보십시오. 이는 대규모 AI 환경에서 하드웨어 비용을 수백만 달러 절감할 수 있습니다 .

굿풋에 집중함으로써 우리는 진정한 핵심 가치, 즉 비용 1달러당, 전력 1줄당 수행되는 유용한 훈련량을 최적화합니다. 굿풋은 순수 FLOPS나 장치 활용률보다 더 중요한 궁극적 성공 지표입니다. 이는 하드웨어, 소프트웨어, 알고리즘이 AI 모델을 더 빠르고 저렴하게 훈련한다는 최종 목표를 향해 얼마나 조화롭게 작동하는지를 포괄적으로 보여주기 때문입니다.

굿풋을 개선하려면 하드웨어(예: CPU, GPU, 네트워크 토폴로지, 메모리 계층 구조, 스토리지 레이아웃), 소프트웨어(예: 운영체제 구성, 페이지 메모리, I/O 활용도), 알고리즘(예: 트랜스포머 아키텍처 변형, 어텐션 메커니즘 대안, 다양한 캐싱 및 배치 전략) 간의 상호작용에 대한 깊은 이해가 필요합니다.

하드웨어, 소프트웨어, 알고리즘을 포함한 여러 분야의 광범위하고 심층적인 이해가 바로 오늘날 AI 시스템 성능 엔지니어가 극히 드문 이유입니다. 또한 제가 이 책을 쓰는 이유이기도 합니다! 다음은 이 책의 나머지 부분을 안내하는 로드맵과 방법론입니다.

책의 로드맵과 방법론

100조 매개변수 AI 시스템 최적화를 어떻게 접근할 것인가? 이 책은 하드웨어 기초부터 소프트웨어 스택과 알고리즘 기법까지 단계별로 안내하며, 각 단계에서 실습 중심 분석에 중점을 둡니다. 책의 나머지 구성은 다음과 같습니다.

2장에서는 NVIDIA AI 시스템 하드웨어를 심층적으로 살펴봅니다. 여기에는 Grace Blackwell Superchip 설계와 NVLink 네트워크를 결합해 AI 슈퍼컴퓨터 수준의 성능/전력 특성을 구현한 GB200/GB300 NVL72 "랙 내 AI 슈퍼컴퓨터"가 포함됩니다.

3~5장에서는 GPU 기반 AI 시스템을 위한 OS 수준, 네트워킹, 스토리지 최적화를 다룹니다. 이러한 최적화에는 CPU 및 메모리 고정(pinning), Docker 컨테이너 및 쿠버네티스 오케스트레이션 고려 사항(GPU 환경을 위한 네트워크 I/O 및 스토리지 구성 포함)이 포함됩니다.

6~12장에서는 혁신적인 하드웨어 인식 알고리즘 개발에 필수적인 NVIDIA CUDA 프로그래밍 기초와 CUDA 커널 최적화를 다룹니다. 플래시 어텐션(Flash-Attention)과 딥시크(DeepSeek)의 MLA 같은 인기 알고리즘이 이에 해당합니다. 이 알고리즘들은 오늘날 생성형 AI 워크로드를 주도하는 트랜스포머 아키텍처의 자원 집약적 어텐션 메커니즘을 대상으로 합니다.

NVIDIA 하드웨어와 CUDA 소프트웨어 환경을 바탕으로, 초대형 모델의 효율적인 훈련 및 서비스를 포함한 분산 통신 최적화에 대해 심층적으로 다룹니다. AI 시스템 스택의 여러 계층에 적용 가능한 패턴인 계산과 통신 중첩과 같은 통신 최소화 전략을 검토합니다.

13장과 14장에서는 PyTorch 컴파일러 스택과 OpenAI의 Python 기반 Triton 언어 및 커스텀 GPU 커널용 컴파일러를 포함한 PyTorch 특화 최적화 기법을 다

룹니다. 이러한 컴파일러는 CUDA 커널 개발에 일반적으로 요구되는 C++에 대한 깊은 이해 없이도 새로운 CUDA 커널 개발의 진입 장벽을 낮춥니다. 이 장들에서는 데이터 병렬화(DP), 완전 분할 데이터 병렬화(FSDP), 텐서 병렬화(TP), 파이프라인 병렬화(PP), 컨텍스트 병렬화(CP), 전문가 혼합(MoE) 등 모델 훈련을 위한 분산 병렬화 기법도 다룹니다. 수조 개 이상의 매개변수를 가진 모델을 다수의 GPU에 효율적으로 분할하여 훈련하는 방법을 보여줄 것입니다. 또한 활성화 체크포인트, 최적화 상태 분할, 더 큰 CPU 메모리로의 오프로딩 등 초대형 모델 훈련 중 메모리 최적화 기법에 대해 논의할 것입니다. 이러한 기법은 모델 크기가 물리적 GPU 하드웨어 한계를 초과할 때 필수적입니다.

15~19장은 고처리량, 저지연 모델 추론 및 에이전트형 AI 시스템을 위한 소프트웨어 및 알고리즘 혁신에 중점을 둡니다. 여기에는 NVIDIA Dynamo 및 커뮤니티 스택에서 이제 지원되는 분산 프리필 및 디코딩이 포함되며, GPUDirect RDMA를 통한 UCX, NCCL 포인트투포인트 또는 프레임워크 제공 전송을 통한 키-값(KV) 캐시 이동이 적용됩니다. 또한 vLLM, SGLang, NVIDIA TensorRT LLM 등 널리 사용되는 모델 서빙 엔진에 대해서도 논의합니다. 이어서 이러한 엔진과 통합되며 분산 프리필 디코딩 환경에서 저지연 KV 캐시 전송을 위한 NIXL을 포함하는 NVIDIA Dynamo 분산 추론 프레임워크를 다룹니다.

또한 NVL72의 Grace CPU를 활용한 전처리, 추측 디코딩과 같은 고성능 추론 알고리즘을 위한 소형 "초안" 모델의 동시 실행, 추론 시스템의 전체 처리량 극대화를 위한 효율적인 요청 라우팅 및 배치 기법도 살펴봅니다. 4비트 양자화, 더 우수한 '교사' 모델로부터 더 작은 '학생' 모델을 학습시키는 지식 증류, 스파스성 및 프루닝, 특수 TensorRT 커널 활용 등 모델 압축 및 가속 기법도 탐구합니다. 디스어그리게이티드 프리필-디코드 및 런타임에 시스템을 동적으로 조정하는 적응형 기법에 중점을 둡니다.

제20장에서는 커널 및 AI 시스템 성능 최적화를 위한 AI 지원 도구의 현대적 활용 방안을 설명합니다. 이 장에는 수십억 및 수조 매개변수 모델을 효율적으로 훈련하고 서비스하는 사례 연구도 포함됩니다. 또한 자체 개선형 AI 시스템 최적화와 에이전트에 대한 신흥 동향도 다룹니다. 이를 통해 100조 매개변수 규모의 AI 시스템이 향하는 방향과 AI 시스템 성능 엔지니어링의 미래를 대비하는 방법을 그려볼 수 있습니다.

부록에서는 독자의 AI 시스템에 적용할 수 있는 일반적인 성능 최적화 및 비용 절감 팁과 요령을 정리한 체크리스트를 제공합니다. 이는 책 전반에 걸쳐 논의된 실행 가능한 최적화 및 효율성 향상 방안의 요약입니다.

요약하자면, 이 책은 성능 최적화를 적용하기 위한 실용적이고 경험적인 방법론을 구현합니다. 우리는 병목 현상을 이해하고 개선 효과를 검증하기 위해 실제 실

행 결과, 사례 연구, 벤치마크 결과, 프로파일링 데이터를 자주 분석할 것입니다. 이 책을 마치면 초대형 AI 시스템 최적화의 원칙을 이해하고, NVIDIA GB200/GB300 NVL72 AI 슈퍼컴퓨터와 같은 초스케일, 다중 GPU, 다중 노드, 다중 랙 AI 시스템 또는 현재 및 미래의 유사한 AI 시스템에 이러한 최적화를 적용하는 도구에 대한 실무 경험을 얻을 수 있을 것입니다.

핵심 요약

다음과 같은 자질들이 종합적으로 AI 시스템 성능 엔지니어의 역할을 정의합니다. 이 엔지니어는 심층적인 기술 지식과 전략적, 자질 기반 최적화를 결합하는 전문성을 바탕으로 원시 하드웨어를 비용 효율적이고 고성능의 AI 솔루션으로 변환합니다:

좋은 처리량 측정

순수한 FLOPS나 활용률 너머를 보십시오. 대신 GPU가 유용한 작업(예: 전파/역전파 계산)을 수행하는 시간과 데이터 대기 또는 기타 오버헤드에 소요되는 시간의 비율을 측정하십시오. NVIDIA Nsight Systems/Compute 또는 PyTorch 자질을 사용하여 이 비율을 측정하십시오. 이 비율을 개선하기 위해 노력하십시오. 굿스루는 효과적이고 유용한 GPU 활용에 초점을 맞추기 때문입니다.

무분별한 지출보다 숙련된 엔지니어링 최적화를 선택하십시오

하드웨어 증설이 만능 해결책은 아닙니다. 하드웨어가 제한적일 때도 영리한 소프트웨어 및 시스템 최적화로 격차를 해소할 수 있으며, 이는 훨씬 더 비싼 인프라가 필요했을 결과를 가능케 합니다. DeepSeek의 성과에서 이를 확인할 수 있습니다. 통신 및 하드웨어 사용을 최적화함으로써, 그들의 엔지니어들은 제한된 H800 GPU(제한된 상호 연결 대역폭)로 최첨단 모델을 훨씬 적은 비용으로 훈련했습니다. DeepSeek 모델은 훨씬 더 강력한 하드웨어에서 훈련된 최첨단 모델의 성능과 동등했습니다. 즉, 숙련된 엔지니어링이 무차별적인 지출을 능가한 것입니다.

점진적 최적화로 차원이 다른 영향력 추구

규모가 커질수록, 비율로는 작아도 효율성 향상은 수백만 달러를 절약할 수 있습니다. 달리 말하면, 중복 계산이나 느린 데이터 파이프라인 같은 사소한 비효율도 시스템이 확장될수록 비용을 은밀히 증가시킬 수 있습니다.

자질 중심 사고방식으로 성능 튜닝에 접근하라

데이터 및 자질 도구를 활용해 최적화를 안내하십시오. 자질 프로파일러를 사용해 진정한 병목 현상(컴퓨팅 활용도, 메모리 대역폭, 메모리 지연, 캐시 미스, 통신/네트워크 지연 등)을 식별한 후 해당 병목에 대한 표적 최적화를 적용하십시오.

전체적인 관점을 유지하라

AI 시스템 성능 개선은 GPU, CPU, 메모리, 네트워크 등 하드웨어와 알고리즘, 라이브러리 같은 소프트웨어를 아우릅니다. 어느 계층의 취약점도 전체를 병목시킬 수 있습니다. 최고의 성능 엔지니어는 하드웨어-소프트웨어 공동 설계를 고려합니다: 때로는 알고리즘 변경이 하드웨어 한계를 완화하고, 때로는 새로운 하드웨어 기능이 새로운 알고리즘을 가능하게 합니다.

최신 하드웨어, 소프트웨어 및 알고리즘에 대한 최신 정보를 파악하라

현대 AI 하드웨어와 소프트웨어는 급속히 진화하고 있습니다. 통합 CPU-GPU 메모리, 더 빠른 상호 연결, 새로운 수치 정밀도 형식 같은 새로운 기능들은 최적의 전략을 바꿀 수 있습니다. 우수한 성능 엔지니어는 이러한 변화를 주시하며, 이에 따라 자신의 멘탈 모델을 업데이트하여 병목 현상을 신속히 제거합니다. 또한 MLPerf 벤치마크 제품군은 다양한 모델에 대한 AI 하드웨어 성능을 이해하는 데 훌륭한 자료입니다.

결론

이 입문적 분석은 대규모 환경에서 최적화가 선택 사항이 아닌 절대적 필수 요소임을 강조합니다. 이는 작동하는 시스템과 완전히 실용성이 없는 시스템의 차이를 만듭니다. 하드웨어나 알고리즘에 대한 기존 접근법은 이 규모에서는 한계에 부딪힙니다. 발전을 위해서는 첨단 하드웨어와 지능형 소프트웨어 기법이 모두 필요합니다.

AI 모델이 물리적 자원 한계를 압박하고 있음은 분명합니다. 하드웨어는 새로운 모델 아키텍처와 알고리즘을 따라잡기 위해 분주합니다. 그리고 이 모든 고가의 기계가 실제로 성과를 내도록 보장하는 핵심 역할을 수행하는 이들이 바로 성능 엔지니어입니다.

AI 시스템 성능 엔지니어의 역할이 점점 더 중요해지고 있음을 입증했습니다. 단순히 돈과 하드웨어를 쏟아붓는 것만으로는 부족합니다. AI 역량의 다음 도약을 위해 모델 아키텍처, 알고리즘, 하드웨어, 시스템 설계 등 모든 요소를 함께 최적화해야 합니다.

AI 시스템 성능 엔지니어의 업무는 다학제적이며 복잡하고 역동적입니다. 하루는 GPU 자질 분석에 몰두하고, 다음 날은 네트워크 토폴로지를 분석하며, 그다음 날은 알고리즘 복잡도를 연구할 수도 있습니다. 이는 하드웨어와 소프트웨어 양쪽에서 가능한 성능을 한 방울도 남김없이 끌어내는 것을 사랑하는 '풀스택' 성능 전문가의 역할입니다.

본질적으로 AI 시스템 성능 엔지니어의 좌우명은 '기계적 공감'입니다. 우리는 하드웨어와 소프트웨어라는 기계적 구조를 깊이 이해함으로써 전체 스택의 성능 역량을 최대한 활용하는 효율적인 솔루션을 맞춤화합니다.

앞서 살펴본 바와 같이, 100조 매개변수 모델은 현재 하드웨어의 역량을 훨씬 뛰어넘습니다. 엑사스케일 시스템이라 해도 단 한 번의 훈련 실행에 수천 년의 GPU 시간이 소요될 것입니다. 이는 명백히 비현실적이며, 이러한 규모에 도달하기 위해서는 첨단 하드웨어와 함께 지능적인 소프트웨어 최적화가 모두 필요함을 강조합니다.

앞으로의 장들에서는 프로세서부터 메모리, 상호 연결 장치, 소프트웨어 프레임워크에 이르기까지 AI 시스템의 구성 요소를 분해하는 방법을 보여주고, 각 구성 요소를 체계적인 방식으로 최적화하는 방법을 배울 것입니다. 작은 변경이 성능과 비용 측면에서 큰 개선을 가져오는 구체적인 사례 연구를 살펴볼 것입니다. 이를 통해 다차원적인 성능 최적화에 대한 멘탈 모델을 구축하는 데 도움을 드릴 것입니다.

이 여정을 마치면 독자이자 실무자인 여러분은 오늘날의 모범 사례에 대한 지식과 내일의 도전을 해결할 엔지니어링 사고방식을 갖추게 될 것입니다. 현재와 미래에 걸쳐 AI 시스템을 한계까지 끌어올릴 수 있는 다양한 기법을 보유하게 될 것입니다. AI 시스템 성능 엔지니어에게 주어진 임무는 분명합니다. 우리는 이러한 혁신에서 배우고 스택의 모든 수준에서 과감한 최적화를 적용할 준비를 해야 합니다.

이제 배경이 마련되었으니, CPU, GPU, 메모리 기술, 네트워크 패브릭, 저장 메커니즘 등 현대 AI 시스템의 하드웨어 구성 요소를 살펴보겠습니다. 현대 AI 슈퍼컴퓨터의 기반이 되는 구성 요소를 연구함으로써, 후속 장에서 심층적으로 다룰 최적화 기법의 기초를 마련하는 핵심 원리를 습득하게 될 것입니다.