최대 10X 빠른 KV 캐시 복구: TurboQuant, 플래시블레이 드(FlashBlade)를 만나다

첫 번째 TurboQuant 게시물에서 Google Research의 KV 캐시 압축 알고리즘에 대한 수학을 살펴보았습니다. 짧은 버전: 원 래 FP16 데이터에 대해 3비트 정밀도로 최대 5배의 압축, 교정 불필요. 수학적 분석을 원한다면 여기에서 시작하세요.

이 게시물은 TurboQuant를 Everpure™ FlashBlade//S™ 시스템 위에 올려놓으면 어떤 일이 일어나는지에 대한 것입니 다. 적절한 스토리지 스택이 없는 압축은 대부분의 성능을 저하시키기 때문입니다.

펀치라인은 다음과 같습니다.

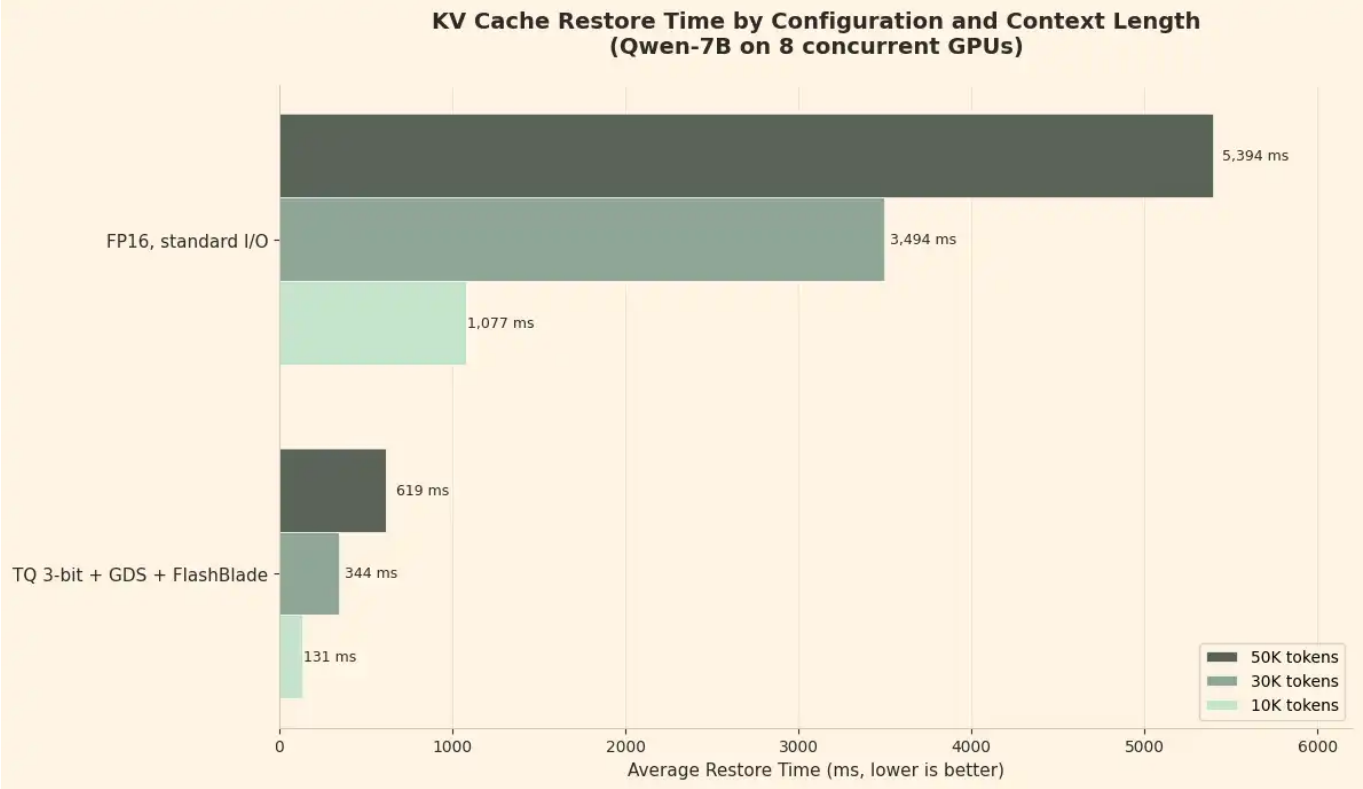


그림 1: TQ 3비트 + GDS + 플래시블레이드(FlashBlade) 대 FP16 표준 I/O를 사용한 KV 캐시 복구 시간 비교

이는 OS 페이지 캐시를 우회하고 원시 플래시블레이드(FlashBlade)® 성능을 분리O_DIRECT하기 위해 측정된 8개의 동시 GPU에 대한 평균 복구 시간입니다. vLLM과 같은 프로덕션 추론 엔진에서는 엔진이 페이지 캐싱, 사전 할당된 버퍼 및 비동기 I/O를 통해 KV 캐시 주입을 최적화하기 때문에 모든 행에 대해 절대 시간이 더 빨라집니다. 상대적인 속도 향상은 중요합니다. CPU가 없는 I/O 경로를 통해 20% 더 적은 데이터를 이동하는 물리학은 엔진이 더 빨라질 때 변경되지 않습니다.

이를 염두에 두고 비율에 집중하세요. 컨텍스트가 증가함에 따라, 압축되지 않은 복원 시간이 풍선처럼 길어지고, GPU 직접 스토리지(GDS) 가 탑재된 TurboQuant(TQ)는 50K 토큰에서도 1초 미만의 시간을 유지합니다. 컨텍스트가 길수록 더 많은 압축이 이루어집니다.

그리고 이러한 벤치마크는 콜드 프리필이 아닌 캐시 인젝션을 측정합니다. 주입은 항상 재계산보다 빠릅니다. 플래시블레이드(FlashBlade) 에 GDS가 탑재된 TurboQuant는 더 빠르게 주입할 수 있습니다.

병목 현상이 실제로 발생하는 곳

사용자가 대화로 돌아가면, 추론 서버는 KV 캐시를 영구 스토리지에서 GPU 메모리로 다시 로드해야 첫 번째 토큰을 생성할 수 있습니다. 그때 는 재방문 사용자를 위한 첫 번째 토큰을 만들 때입니다. 약 500ms 이상의 모든 것이 느려지기 시작합니다. 그래프는 10K 토큰에서 1초, 30K 토큰에서 3.5초, 530K 토큰에서 5초 이상 기존 경로와 가까운 곳이 없음을 보여줍니다50K. 컨텍스트가 증가함에 따라 문제가 악 화되고 동시 사용자도 여전히 악화됩니다.

병목 현상은 스토리지가 아닙니다. 이것이 바로 길입니다. 기존 복원은 CPU를 바이트당 중간에 둡니다. 스토리지는 CPU 메모리로 판독하고, CPU는 DMA를 관리하며, CPU는 PCIe를 통해 이를 GPU 메모리로 복사합니다. 한 번에 한 세션씩 진행해도 괜찮습니다. 8개의 GPU를 동시에 복구하는 경우, CPU는 하나의 공유 메모리 서브시스템을 통해 8복제 분량의 작업을 수행하고 있습니다. 이 수정은 CPU 속 도 향상이 아닙니다. CPU를 완전히 제거해야 합니다.

GPU 다이렉트 스토리지는 CPU를 완전히 제거합니다.

GPU 직접 스토리지(GDS)는 데이터 경로를 다음과 같이 변경합니다.

스토리지 → CPU 메모리 → GPU 메모리

to:

스토리지 → GPU 메모리(RDMA를 통해 직접)

CPU가 관여하지 않습니다. 플래시블레이드(FlashBlade) 시스템은 GPU HBM에 직접 쓰고, GPU는 압축 해제를 처리합니다.

GDS가 없는 압축은 SUV를 스포츠카로 교환한 후 동일한 그리드로 고정된 고속도로를 이용하는 것과 같습니다. 차량은 더 작지만 여전히 같 은 교통 상황에 처해 있습니다. GDS는 스포츠카가 그리드락으로 집업할 수 있는 고속 차선입니다.

TurboQuant 및 GDS를 사용하면 복구 경로가 다음과 같이 됩니다.

플래시블레이드(FlashBlade)(RDMA) → GPU 메모리 → GPU 압축 해제

30K 토큰에서는 세션당 349.5MB만이 와이어를 통해 이동합니다(FP16의 1.72GB에서 압축된 3비트 인덱스, 규범, 회전 매트릭 스 및 코드북 감소). GPU 메모리에 직접 들어옵니다. GPU는 3비트 인덱스의 압축을 풀고 역회전을 밀리초 미만의 시간으로 적용합니다. 비 트 조작과 매트릭스 곱셈은 대규모 병렬 프로세서가 설계한 것과 같습니다.

8개의 GPU에 걸친 총 읽기 대역폭은

구성

10K □□

30K □□

50K □□

FP16, □□ I/O

4.19GB/s

3.28GB/s

4.22GB/s

TQ 3□□ + GDS

7.11GB/s

8.13GB/s

7.53GB/s

TurboQuant with GDS는 전체 데이터를 5X 적게 읽음에도 불구하고 모든 컨텍스트 길이에서 다른 구성보다 높은 집계 읽기 대역 폭을 제공합니다. 각 GPU는 컨텍스트 길이에 관계없이 자체 GDS/RDMA 연결을 통해 약 900~1,000MB/s를 일관되게 유지 합니다. 플래시블레이드(FlashBlade) 시스템은 GPU가 수용할 수 있는 속도만큼 데이터를 제공합니다.

측정 내용

벤치마크는 NFS 기반의 Everpure 플래시블레이드(FlashBlade)에 연결된 8개의 A100-40GB GPU와 RDMA를 탑 재한 NVIDIA DGX 시스템에서 실행되었습니다. 모델은 Qwen2.5-7B-Instruct로, 헤드 치수가 128인 TurboQuant에 중요한 차원입니다. 이는 치수가 증가함에 따라 알고리즘 작업을 더 강하게 만드는 측정 현상의 농도가 높아지기 때문입니다.

KV 캐시 크기는 일관된 4.9배 감소로 컨텍스트 길이에 따라 선형으로 확장됩니다.

컨텍스트 길이

FP16 KV 캐시 크기

TQ 3□□ □□

10K □□

573MB

116.5MB

30K □□

1.72GB

349.5MB

50K 토큰

2.87GB

582.4MB

각 벤치마크는 독립적인 체크포인트 및 복구 주기를 수행하는 8명의 동시 GPU 작업자를 운영했습니다. 이것이 바로 프로덕션의 모습이기 때문 입니다. 한 번에 한 세션씩 체크포인트를 하는 사람은 없습니다.

쓰기는 예측 가능하며 읽기는 성공입니다.

체크포인트(쓰기) 성능은 모든 구성에서 일관적이었습니다. 작업자당 쓰기 시간은 낮은 차이를 보였으며, 이는 플래시블레이 드(FlashBlade) 시스템이 8명의 동시 작성자를 원활하게 처리했음을 의미합니다.

비대칭은 중요합니다. 프로덕션 환경에서 체크포인트는 쓰기 작업이 한 번이며 읽기 작업이 많습니다. 세션이 유휴 상태가 되면 체크포인트를 확인 하고, 사용자가 다시 돌아오면 복구합니다. 읽기 속도가 8~10배 더 빨라지는 대신 쓰기 속도가 약간 느려지는 것은 매번 발생하는 절충입니다.

마이그레이션 후 100% 토큰 정확도

빠른 복구는 모델이 나중에 다른 토큰을 생성하는지 여부와는 상관 없습니다. 따라서 50,000개의 컨텍스트 토큰으로 전체 세션 마이그레이션을 사용하여 가장 엄격한 테스트를 수행했습니다.

GPU 0은 3비트 TurboQuant를 사용하여 2.87GB KV 캐시를 압축하고, 플래시블레이드(FlashBlade)에 쓰고, 압축된 데이터를 다시 읽고, 압축 해제하고, GPU 1에 로드합니다. 그런 다음, 이 모델은 교사가 시행하는 평가를 통해 50개의 토큰을 생성하고, 각 예측된 토큰을 원래 비압축 캐시의 실제 사실과 비교합니다.

그 결과, 50개의 토큰 중 50개가 일치했습니다. 100% Top-1 정확도. 압축 및 마이그레이션된 모델은 원본과 정확히 동일한 출력을 생 성했습니다. 이는 근사치가 아닙니다. 예측된 모든 토큰은 동일했습니다.

단순한 3비트가 아닙니다. 품질/크기 균형 조정

TurboQuant는 사용자를 단일 압축 레벨로 고정하지 않습니다. 알고리즘은 헤드당 128개의 채널을 분산별로 두 그룹으로 분할하는 혼합 정밀도 모드를 지원하여 높은 비트 폭에서 고가변(이상치) 채널을, 낮은 비트 폭에서 REST 채널을 정량화할 수 있습니다. 이를 통해 효율적인

비트 전송률과 미세한 제어가 가능해 질 수 있습니다.

효과적인 비트

압축률

코사인 유사성

TB당 세션(10K 토큰)

FP16(베이스라인)

1.0x

1.000

1,743

2.5□□

5.3x

0.964

9,297

3□□

4.9x

0.983

8,580

3.5□□

4.0x

0.990

6,973

4□□

3.8x

0.995

6,562

2.5비트 모드는 4비트에서 32개의 이상치 채널을 사용하고 2비트에서 나머지 96개를 사용하여 좌표당 2.5비트의 유효 속도를 생성합니다. 3.5비트 모드는 5비트에서 32채널, 3비트에서 REST 채널을 지원합니다. 각 하위 공간은 적절한 차원에 대해 자체 회전 매트릭스와 코드 북을 계산하므로, 수학적 최적화는 각 하위 그룹 내에서 여전히 유지됩니다.

이는 프로덕션 배포에 중요합니다. 활용 사례마다 근사치에 대한 허용 오차가 다르기 때문입니다. 문서를 검색하고 요약하는 고객 지원 챗봇은 5.3배 압축 및 0.964 코사인 유사성으로 2.5비트에서 완벽하게 만족할 수 있습니다. 모든 토큰이 중요한 코드 생성 도우미는 3.8X 압축 및 0.995 코사인 유사성을 갖춘 4비트를 원할 수 있습니다. 터보퀀트를 사용하면 배포당 또는 레이어당 이러한 선택을 할 수 있습니다.

테라바이트당 5X 더 많은 세션

10,000개의 토큰 세션(대략 7,500단어의 대화)은 FP16에서 573MB를 차지합니다. 3비트 TurboQuant는 117MB 입니다. 전용 KV 캐시 네임스페이스(작은 프로덕션 구축 조각)를 위해 플래시블레이드(FlashBlade) 10TB를 확보하세요. TurboQuant는 FP16에서 17,430개 대신 85,800개의 따뜻한 세션을 제공합니다. 동일한 하드웨어, 동일한 랙 공간, 동일한 전력, 처음부터 다시 계산하는 대신 즉시 복구할 수 있는 5X 더 많은 사용자.

플래시블레이드(FlashBlade)는 GPU 수에 따라 선형으로 확장됩니다.

8GPU 결과는 단 8개의 GPU보다 더 큰 스토리를 전달합니다. 각 GPU는 자체 GDS/RDMA 연결을 통해 CPU 조정 및 공유 잠 금 없이 세션을 독립적으로 복원합니다. 플래시블레이드(FlashBlade)의 분산 아키텍처(여러 블레이드에 걸친 병렬 플래시)는 수백 GB/s 의 동시 처리량을 유지하도록 구축되었으며, SuperPOD 인증은 동일한 아키텍처가 수백 개의 GPU로 확장된다는 것을 의미합니다. 벤치 마크는 스토리지 제한이 아닌 GPU로 제한되었습니다.

GPU 수

□□ □□

□□ □□ □□ □□□

□□□ □□(30K)

8 (□□)

8

8.13GB/s

344ms

16(□□)

16

~16GB/s

344ms

32(□□)

32

~32GB/s

344ms

64(예상)

64

~65GB/s

344ms

TurboQuant with GDS는 스토리지 대역폭 문제에서 GPU당 지연 시간으로 KV 캐시 복원을 전환하며, 해당 지연 시간은 클러스 터 크기에 관계없이 일정하게 유지됩니다.

16TB의 문제: 하이퍼스케일러가 필요한 이유

NVIDIA는 AI 공장 아키텍처에 GPU당 16TB의 KV 캐시 스토리지가 필요하다고 말했습니다. FP16에서는 데이터센터에서 성능에 가장 민감한 스토리지인 GPU당 16TB로, 수천 개 GPU에서 동시에 1초 미만의 읽기를 대규모로 제공해야 합니다.

이 요건은 TurboQuant가 존재하기 전에 정의되었습니다.

3비트 압축에서는 GPU당 16TB가 3.3TB로 떨어집니다. 1,000-GPU 클러스터에 곱하기 및 TurboQuant는 16페타바이 트에서 3.3페타바이트까지 스토리지 공간을 차지합니다. 델타가 나타내는 달러와 와트 및 랙 공간은 사소한 것이 아닙니다.

하이hyperscale KV 캐시는 더 이상 메모리 문제가 되지 않습니다. 이는 스토리지 인프라 문제이며, TurboQuant 압축과 플래시 블레이드(FlashBlade) 시스템 및 GDS의 결합으로 프리미엄 스토리지의 GPU당 16TB 없이 GPU당 16TB를 실현할 수 있 습니다.

추론 스택의 의미

KV 캐시 문제는 사라지지 않습니다. 컨텍스트 창이 증가하고 있습니다. 동시 사용자 수가 증가하고 있습니다. 모델이 점점 커지고 있습니다. 이러 한 트렌드는 모두 캐시를 더 크게 만들고 복구 문제를 더 어렵게 하며, 벤치마크는 컨텍스트 길이에 따라 문제가 불균형하게 악화됨을 보여줍니다.

표준 플레이북은 HBM을 더 많이 던지는 것이었습니다. 더 많은 GPU를 구매하고, 더 많은 메모리로 사용하고, 모든 것을 상주하세요. 이 플 레이북은 더 이상 작동하지 않습니다.

플래시블레이드(FlashBlade)와 GDS를 갖춘 TurboQuant는 다른 답을 제공합니다. GPU에서 캐시를 압축하세요. RDMA 를 통해 플래시블레이드(FlashBlade)에 쓰기 사용자가 다시 돌아오면 압축된 데이터를 GPU 메모리로 직접 스트리밍하고 압축을 해제합 니다. 8개의 GPU 또는 256개의 GPU를 보유하고 있든 관계없이 동일한 복구 시간. 사용자는 절대 알아차리지 못합니다.

KV 캐시는 더 이상 메모리 용량 문제가 아닙니다. 스토리지 I/O 문제이며 스토리지 I/O는 Everpure가 해결 방법을 알고 있는 문제입니 다.