

Computational Storage란 무엇인가

크라프트톤 정글 8기 301 09 박창현

서론

디지털 데이터의 급속한 증가와 실시간 데이터 처리 수요가 높아지면서 기존의 호스트 중심 구조는 데이터 이동으로 인한 병목 현상이 심화되고 있다. 이를 해결하기 위한 새로운 기술로, 데이터를 저장장치에서 직접 처리하는 컴퓨테이셔널 스토리지가 주목받고 있다. 이 보고서에서는 컴퓨테이셔널 스토리지의 개념, 기술 원리, 장단점 및 주요 기업 사례를 살펴본다.

본론

1. 개념정의

컴퓨테이셔널 스토리지는 저장장치(SSD, HDD 등) 내부에 CPU, FPGA, ASIC 등의 연산 장치를 탑재하여 데이터 저장뿐 아니라 직접 연산 작업을 수행하는 기술이다. 이는 데이터 이동을 최소화하고 병렬 처리를 통해 전체 시스템 성능과 에너지 효율성을 높인다.

2. 기술 원리

컴퓨테이셔널 스토리지는 데이터이동을 줄이고 연산을 데이터에 가깝게 수행한다는 원리를 중심으로 한다. 이러한 원리는 저장장치 내부에 프로세서나 하드웨어 가속기를 탑재하는 방식으로 구현되며, 주로 데이터 필터링, 압축, 추론 등의 연산 작업을 저장장치에서 직접 수행하는 형태로 나타난다. 아래에 컴퓨테이셔널 스토리지의 기술 원리

데이터 근접 처리(Near-Data Processing)

컴퓨테이셔널 스토리지는 '연산을 데이터가 있는 곳으로 이동'시키는 근본 원리를 따른다. 저장장치 내부 혹은 아주 인접한 영역에서 데이터를 처리함으로써, 호스트-스토리지 간 대용량 데이터 전송 비용을 최소화하고 지연 시간을 단축한다.

명령어 오프로딩(Offload Command)

기존 NVMe/SATA 명령에 연산 관련 확장 명령을 추가하거나, 별도의 관리 채널을 통해 연산 요청을 전달한다.

- In-band: 표준 NVMe 명령으로 연산 태스크를 제출하여 호스트와 동일한 PCIe 경로로 처리한다.
- Out-of-band: 컨트롤러 전용 채널로 명령을 전달해 호스트 I/O와 독립적으로 연산을 수행한다.

임베디드 컴퓨트 유닛(Embedded Compute Unit)

저장장치 컨트롤러 칩 내부에 탑재된 프로세서 코어(ARM, RISC-V 등)와 하드웨어 가속기(DSP, FPGA, 암호화·압축 가속 블록 등)가 연산을 수행한다.

- 프로세서 코어: 제어 로직, 간단한 필터링·집계·암호화 연산 담당
- 하드웨어 가속기: 대역폭이나 연산량이 높은 작업(ML 추론, 암호화 등) 하드웨어 수준 가속

메모리 계층과 데이터 흐름

1. 플래시/NVM: 비휘발성 셀에 원본 데이터 저장
2. 컨트롤러 DRAM 캐시: 처리 대상 데이터 블록을 로드하여 저지연 액세스 제공
3. DMA 엔진: 메인 메모리(DRAM)와 컨트롤러 캐시 간의 효율적 데이터 이동 담당
4. 호스트 인터페이스(PCIe/NVMe-oF): 결과 전송 및 제어 명령 주고받기

펌웨어·런타임 아키텍처

경량화된 OS(RTOS/Linux) 기반 펌웨어로 연산 태스크를 스케줄링하고, 컨테이너나 사내 런타임 환경을 통해 사용자 정의 연산 코드를 안전하게 실행한다. 작업 간 자원 격리를 위해 프로세스 샌드박스 또는 메모리 보호 기법을 적용한다.

보안 및 격리

- 암호화 가속: 데이터 at-rest-in-flight 암호화 연산을 장치 내에서 처리
- 접근 제어: 사용자·테넌트별 권한 검증 및 메모리 공간 분리
- 펌웨어 무결성: 서명된 펌웨어 업데이트와 안전 부팅(secure boot)

품질 보장(QoS) 및 자원 관리

I/O 우선순위 및 연산 자원(CPU 사이클, 메모리 버퍼)을 동적으로 조절하여, 호스트 워크로드와 스토리지 내 연산이 상호 간섭 없이 목표 성능을 달성하도록 관리한다.

3. 구현방식

- 컴퓨테이션 스토리지 드라이브(CSD): NVMe SSD 내부에 임베디드 프로세서를 탑재해 장치 자체에서 데이터 연산을 수행한다.
- 컴퓨테이션 스토리지 어레이(CSA): 다수의 CSD를 어레이 컨트롤러로 묶어 병렬 연산을 오프로딩한다.
- 컴퓨테이션 스토리지 오버 패브릭(CSF): NVMe-oF 등 고속 네트워크를 통해 원격 스토리지 노드의 연산 기능을 호출한다.
- 오브젝트 스토리지 레벨 컴퓨트: S3 호환 버킷에 서버리스 함수를 바인딩해 읽기·쓰기 이벤트 시 연산을 자동 실행한다.
- SmartNIC/DPU 기반 Near-Data Processing: 네트워크 인터페이스 칩 내 프로세서와 하드웨어 가속기로 데이터 패킷 처리와 연산을 병행한다.
- FPGA 기반 임베디드 가속: 스토리지 컨트롤러 SoC 내부에 FPGA 로직을 통합해 사용자 정의 연산을 하드웨어 수준에서 가속한다.
- 호스트 관리형 오픈 채널 SSD: 호스트 커널이 플래시 매핑·가비지 컬렉션·웨어 레벨링을 직접 제어하며 SSD 채널에서 연산을 수행한다.
- 컴퓨테이션 스토리지 게이트웨이: SAN/NAS 환경의 어플라이언스에 연산 엔진을 탑재해 데이터 흐름을 가로채 처리한다.

4. 장점 및 단점

장점

- 성능 향상: 데이터 이동 없이 저장장치에서 직접 병렬 처리하여 응답 시간을 크게 단축할 수 있다.
- 전력 효율성 향상: 데이터 이동을 최소화하여 시스템의 에너지 소비를 줄일 수 있다.

- 확장성 향상: 스토리지 장치 수를 증가시킬 때 연산 자원도 함께 선형적으로 증가하여 시스템의 확장성이 뛰어나다.
- 호스트 CPU 부하 감소: CPU의 일부 작업을 스토리지 장치가 대신 수행함으로써, 호스트의 핵심 연산 작업에 자원을 더 집중할 수 있다.

단점 및 한계

- 표준화 미흡: 아직 산업 표준이 명확하지 않아 기업별로 호환성 문제가 존재한다.
- 소프트웨어 지원 부족: 기존 소프트웨어를 저장장치 내에서 연산을 수행하도록 수정해야 하는 복잡성이 존재한다.
- 연산 자원의 제한: 저장장치 내부의 공간 및 전력 한계로 높은 성능의 복잡한 연산에는 제약이 있다.
- 비용 증가: 추가 프로세서, FPGA 또는 ASIC 장착으로 인해 초기 도입 비용이 증가한다.
- 보안 취약성: 저장장치 내부의 연산 장치 추가로 펌웨어 보안, 데이터 보호와 관련된 추가 보안 관리 요구사항이 증가한다.

5. 주요 기업 사례

삼성전자 - SmartSSD

삼성전자는 FPGA와 Arm 기반 프로세서를 내장한 SmartSSD를 출시하여 데이터베이스 검색, 영상 트랜스코딩, 머신러닝 추론 등의 작업을 직접 처리한다. 이를 통해 데이터 처리 시간 최대 50% 단축, 전력 소비 최대 70% 절감하는 성과를 보였다.

NGD Systems - Newport Platform

NGD Systems는 고용량 SSD 내부에 Arm Cortex 프로세서를 탑재하여 데이터 분석 작업을 현장에서 실시간으로 처리한다. 이는 데이터 전송량을 최소화하여 네트워크 트래픽과 전력 소모를 줄이고, IoT 및 엣지 컴퓨팅 환경에서 유용하게 활용된다.

ScaleFlux - CSD 시리즈

ScaleFlux는 데이터 압축/해제 가속 회로를 SSD에 탑재하여 데이터 압축을 통해 최대 4배의 논리 저장 용량을 확보하고, 스토리지의 내구성과 성능을 향상시킨다. 데이터베이스, 빅데이터 분석 및 가상화 환경에서 효과적으로 활용된다.

6. 결론 및 전망

컴퓨테이셔널 스토리지는 병목 현상을 최대한 줄이기 위해 병렬처리를 시도하거나, 데이터 사이즈를 줄이거나, 성능을 가속화하는 등의 방식으로 CPU의 대용량 데이터 처리 부담을 줄이기 위한 여러가지 방식을 고민하는 것으로 보인다. AI가 새로운 기술로 발전함에 따라 지금보다 더 많은 데이터를 처리해야할 상황이 필수적으로 발생할텐데 그 부분까지 포함하는게 가능한지 실질적으로 와닿지 않는 상황이고, 실생활에 접목되었는지도 와닿지 않기 때문에 좀더 지켜봐야할 것이다. 향후 표준화 문제와 소프트웨어 지원 문제, 보안 이슈 등이 해결된다면 데이터센터와 엣지 컴퓨팅 전반에 걸쳐 핵심 기술로 자리매김할 수 있지 않을까.