

분산 지능 시스템의 도래와 컴퓨터 구조의 전환

Computational Storage와 RISC-V를 중심으로

8-301-18
이원규

COMPUTER HARDWARE
&
INDUSTRY PERSPECTIVE

1. 서론

하드웨어를 공부하다 보면, 단순히 부품을 분해하고 조립하는 수준에서 벗어나 **각 부품이 '어떤 역할'을 수행하고 '어떻게 동작'하는지를 보는 눈**이 생긴다. 특히 내가 정글(Jungle) 과정에서 CS를 공부하면서, 컴퓨터가 단순히 '작동한다'는 것에서 벗어나 **메모리, 명령어, 캐시, 입출력 장치들이 실제로 어떻게 연결되고 상호작용 하는지**를 어느 정도 생각하게 되었다.

처음 'Computational Storage'라는 기술을 보았을 땐, 단지 저장장치가 좀 더 똑똑해진 건가 싶었다. 하지만 해당 기술에 대해 더 공부하다 보니, 이건 단순한 기술 용어나 하드웨어 기법이 아니라, **컴퓨터의 전체 시스템 구조와 철학이 바뀌는 흐름**이라는 걸 느꼈다.

이 글에서는 내가 'Computational Storage'를 공부하면서 가졌던 생각들, 그리고 이 기술을 계기로 더 깊게 고민하게 된 **컴퓨터 시스템의 미래 구조**에 대한 내 생각을 정리해보려 한다. 단순한 기술 소개가 아니라, 내가 이해한 바를 토대로 정리한 하나의 흐름과 연결된 이야기다.

2. 본론

2.1 Computational Storage란 무엇인가?

기존 SSD는 데이터를 단순히 저장하고 꺼내는 역할만 했다.

그런데 이제는 **그 안에서 직접 연산을 수행**한다.

ARM 프로세서나 FPGA 같은 연산 유닛을 SSD 내부에 탑재하여,

데이터를 CPU로 가져가지 않고 **저장된 위치에서 직접 필터링, 변환, 정제**하는 것이다.

이 기술은 특히 다음과 같은 작업에 강점을 보였다.

- **대규모 로그 처리**: 조건에 맞는 줄만 추출 → 전체의 1% 미만 필요
- **AI 전처리**: 이미지 필터링, 벡터 정규화, 크기 조정 등
- **DB 반복 쿼리**: 특정 키워드, 범위, 집계 조건 검색 등

즉, 대부분 고성능 컴퓨팅이 필요 없는 단순하지만 연산량이 많은 작업들이다.

해당 작업들에 한하여 이 장치들의 벤치를 보면 성능 향상 폭이 크다.

- 삼성 SmartSSD: SQL 검색 성능 최대 11배, CPU 사용률 97% 감소, 전력 90% 절감
- NGD Systems: AI 전처리 속도 24배, 데이터 이동량 80% 감소
- Kioxia: 압축/암호화 처리 속도 최대 5배, CPU 점유율 90%까지 감소

이처럼 CSD(Computational Storage Device)는 단순히 연산 성능이 뛰어난 게 아니라,

데이터 이동 비용이 큰 시스템 환경에서 구조적인 효율을 제공하여

병목현상 해소와 전력 및 자원 절감이 가능하다는 점이 핵심이다.

2.2 GPU와의 차이점

요즘 대부분의 연산은 GPU가 하고 있다고 해도 과언이 아니다.
하지만 **GPU와 CSD(Computational Storage)**는 전혀 다른 목적과 위치를 가진다.

항목	GPU	Computational Storage
연산 위치	메모리 근처	SSD 내부
특징	병렬 연산 강력	데이터 이동 자체를 줄임
필요 조건	데이터를 먼저 전송	필터링 후 필요한 것만 전송
병목 처리	연산 속도는 강력하지만 I/O 병목 발생	I/O 자체를 줄이거나 제거
사용 예시	딥러닝 학습/추론, 3D 렌더링	로그 필터링, AI 전처리, DB 검색

GPU는 연산력의 끝판왕이지만, I/O 병목은 여전히 존재한다.
CSD는 **아예 병목 자체를 없애는 발상**이다.
둘은 경쟁이 아니라 공존이 가능하다. GPU는 연산 중심, CSD는 데이터 정제 중심이다.

2.3 '미니 컴퓨터화'된 주변 장치

이런 기술이 SSD에만 적용되는 것이 아니다.
오늘날의 컴퓨터 시스템에서는 **거의 모든 부품이 계산을 수행하고 있다**.

부품	사례	내장 연산 기능
SSD	Samsung SmartSSD	FPGA 탑재, SQL 필터링, 압축 처리 등
NIC	SmartNIC, DPU	TCP 오프로딩, 트래픽 분석, 보안 필터링
GPU	NVIDIA RTX	CUDA Core, Tensor Core, 병렬 벡터 연산
RAM	PIM (Processing-In-Memory)	연산용 로직이 탑재된 메모리 셀
모니터	Apple Studio Display	A13 Bionic SoC 내장, 영상 처리 가능

이런 구조를 가리켜 **heterogeneous computing (이기종 연산)** 또는
****domain-specific architecture (도메인 특화 설계)****라고 부른다.
중앙 집중형 구조에서 **분산형 협업 구조로 진화**하고 있는 것이다.

하지만 단순히 계산을 수행한다고 해서 이들을 '미니 컴퓨터'라고 칭하는게 아니다.

이 장치들은 예전부터 고유의 연산 기능을 수행해왔다.

(RAM은 연산 장치, 저장 장치, 입출력 장치라는 컴퓨터 구조의

기본 요소들이 없기에 엄밀히 말하면 '미니 컴퓨터' 라고 하기엔 부족하긴 하다.)

그럼에도 불구하고, 지금의 흐름이 '변화' 처럼 여겨지는 이유는,

"자율성의 수준"과 "역할의 중심 이동"에 있다.

1. "부가 기능" → "핵심 연산 주체"로 역할 변화

- 예전: SSD는 CPU의 요청을 받는 수동적 저장소
- 지금: SmartSSD는 CPU 없이 직접 쿼리를 처리함
- 예전: NIC는 단순 전송만
- 지금: SmartNIC은 로드밸런싱, 암호화, 방화벽까지 담당

2. 연산 유닛의 '범용성' 증가

- 과거: 펌웨어 수준의 고정 기능 (ex. CRC 체크)
- 현재: ARM, RISC-V, FPGA 탑재 → 실제 프로그래밍 가능

3. '단순 보조'에서 '분산 협업자'로 진화

- 이제는 시스템 전체 성능을 위해 '주도적으로 역할을 나눠 맡음'
- GPU는 메인 연산 장치가 되었고, SSD는 전처리 허브가 되며, NIC는 네트워크 방패가 됨

2.5 폰 노이만 구조의 한계와 시스템 재정의의 필요성

현대 컴퓨터는 여전히 1945년 폰 노이만이 제안한 구조를 기반으로 하고 있다.

CPU ↔ Memory ↔ I/O의 세 파트가 명확히 나뉘어 있고,

연산은 항상 CPU가 수행해야 하며, 모든 데이터는 반드시 CPU를 거쳐야 한다.

그런데 오늘날의 AI, 데이터 홍수, 실시간 처리 요구로 인해 시스템의 병목이

CPU나 RAM이 아닌 **I/O와 전송에 발생하는 시대**가 왔다.

연산보다 데이터 이동이 더 비용이 커진 것이다.

그렇기에 이를 극복하기 위해 부분 보강(GPU, DPU, CSD, Smart NIC, PIM 등)이

이루어지고 있다. 일종의 'Patch'인 것이다.

하지만 '기워서' 버티기엔 너무 시스템이 너무 낡은 것이 아닐까...?

이제는 단순히 성능을 높이는 것이 아니라, **구조를 새로 정의해야 할 시점** 아닐까?

그래서 단순한 진화가 아닌, 점프 수준의 패러다임 전환이 필요한 타이밍이 되고 있다.

하드웨어는 이미 준비되어 있다.

이제 필요한 것은,

이를 연결하고 운영할 수 있는 **새로운 시스템 아키텍처와 플랫폼**이다.

2.6 새로운 설계 철학: RISC-V

이 변화의 중심에 떠오르고 있는 이름이 바로 **RISC-V**다.

RISC-V는 단지 또 하나의 CPU 아키텍처가 아니라, **완전히 다른 철학**을 지닌 시스템이다.

항목	기존 시스템(x86, ARM)	RISC-V
ISA 철학	폐쇄적, 복잡	완전 오픈, 단순, 모듈화
명령어 구성	고정식 또는 확장 제한	사용자 정의, 조합 가능
확장성	제약 많음	완전 자유
라이선스	유료, 독점	무료, 누구나 사용가능(오픈 소스)

SSD, SmartNIC, PIM 메모리 등 각 부품마다 적절한 연산 코어를 RISC-V로 구현할 수 있다.

지금 어디까지 왔나?

- AI 칩, SoC, 컨트롤러 등에서 RISC-V 상용화 진행 중
- 리눅스, GCC, LLVM, TensorFlow Lite 등 지원 완료
- Ubuntu/Fedora 등 배포판도 RISC-V 포팅2.7 RISC-V의 통합 이슈와 극복 노력

하지만, RISC-V의 장점은 모듈화이지만, 그로 인해 **통합이 어려워지는 역설**이 존재한다.

- 자유롭지만, 조합이 너무 다양하다 → 다양한 명령어 조합
- OS와 컴파일러가 각 명령어 조합을 일일이 지원해야 한다.
- 하드웨어마다 별도의 드라이버, 커널 옵션, ABI 구조가 필요함
- 개발자가 쓰기엔 진입장벽이 높다.

그래서 이를 해결하기 위한 노력이 동시에 진행 중이다.

- 플랫폼 프로파일 표준화: RV64GC, RVA22 등 표준 조합 정의
- 기업 주도 플랫폼: SiFive, T-Head는 통합 SoC + SDK를 제공해 플랫폼화 진행 중
- 소프트웨어 추상화 계층: 하드웨어 상이함을 추상화하는 소프트웨어 계층 제공

NVIDIA가 GPU + NVLink + Grace CPU + SmartNIC를 한 패키지로 묶는 것도 비슷한 시도이다.

미래에는 "RISC-V 개발"이 아니라, 그냥 "SDK 받아서 개발"하는 세상이 될 수도 있다.

3. 결론

'Computational Storage'는 단지 SSD의 진화가 아니다. 하나의 신호다.
이제는 모든 부품이 지능을 갖기 시작했다.
CPU는 더 이상 모든 것을 계산하지 않고, 조율하고 분산시킨다.
GPU, SSD, RAM, NIC은 각자 필요한 계산을 하고 결과만 전달한다.
시스템은 점점 더 분산 지능 모듈로 구성된다.

우리는 지금, 다음 시스템으로의 전환 직전에 서 있다고 생각한다.
이러한 전환은 서서히 오는 것이 아닌, 조용히 누적된 뒤
모든 것이 맞물리는 순간 '폭발'처럼 올 것이라 생각한다.
옛날 증기기관에서 전기로 넘어갈 때처럼,
진공관에서 트랜지스터로 넘어갈 때처럼 말이다.

이러한 패러다임 전환의 기반은 RISC-V가 마련했고,
이를 실제로 움직이게 할 몫은 소프트웨어에게 주어졌다.

*속력보다 중요한 건 방향이고,
성능보다 본질적인 건 구조다.*