

## < CHIP 2주차 선각 레포트 >

### - Computational Storage에 대한 고찰

KJ-8-301-05 김재현

데이터 기반 컴퓨팅 환경이 고도화됨에 따라, 시스템 내에서 데이터를 처리하는 방식에도 근본적인 변화가 요구되고 있다. 특히 인공지능, 빅데이터 분석, 엣지 컴퓨팅 등에서 생성되는 대규모 데이터는 기존의 중앙 처리 방식(CPU 중심 구조)으로는 처리 효율에 한계가 있다. 데이터가 저장장치에서 CPU로 이동하는 과정에서 발생하는 지연(latency)과 전력 소비가 병목으로 작용하며, 이는 전체 시스템의 성능 저하로 이어진다.

이러한 문제를 해결하기 위한 대안으로 등장한 개념이 컴퓨테이셔널 스토리지(Computational Storage)이다. 이는 저장 장치 자체에 연산 기능을 탑재함으로써, 데이터가 CPU나 메모리로 이동하지 않고 저장 장치 내부에서 직접 연산이 수행되도록 하는 구조를 말한다. 즉, 저장소를 단순한 보조 기억장치가 아닌 '데이터 연산의 일차 주체'로 활용하는 패러다임이다.

컴퓨테이셔널 스토리지 시스템은 일반적으로 세 가지 구성 요소로 나뉜다. 첫째는 연산 능력이 내장된 저장 장치인 CSD(Computational Storage Device), 둘째는 여러 개의 CSD를 묶은 CSA(Computational Storage Array), 셋째는 실질적인 연산을 수행하는 논리 유닛인 CSE(Computational Storage Engine)이다. 이들 구성 요소는 ARM 프로세서, FPGA, 혹은 특수 목적 SoC(System on Chip)를 활용하여, 압축, 필터링, 정렬, AI inference 등 다양한 작업을 스토리지 내에서 수행할 수 있도록 한다.

기술적으로 보면, 이러한 접근은 단순히 저장장치에 연산 유닛을 붙이는 것을 넘어서, 데이터 근처에서 연산을 수행하는 'Near-Data Processing'의 대표적인 구현 형태라고 할 수 있다. 이로 인해 기존 Von Neumann 아키텍처에서 나타나는 '메모리 벽(memory wall)' 문제를 완화할 수 있으며, 메모리 대역폭의 한계를 넘어서는 새로운 시스템 설계 방식이 가능해진다.

컴퓨테이셔널 스토리지는 여러 가지 이점을 제공한다. 첫째, 데이터 이동이 줄어들어 전체 시스템의 대역폭 사용량이 감소하며 병목 현상이 완화된다. 둘째, CPU의 부하가 감소하므로 더 높은 수준의 병렬 처리가 가능해진다. 셋째, 불필요한 I/O를 줄이기 때문에 전력 효율성이 향상된다. 넷째, 엣지 컴퓨팅이나 실시간 분석 환경에서의 지연 시간을 크게 줄일 수 있다. 특히 로그 필터링, AI 모델 추론, 영상 데이터 전처리 등에서 높은 실효성을 보여주고 있다.

하지만 해결해야 할 과제도 명확하다. 컴퓨테이셔널 스토리지는 기존의 CPU-RAM-Storage 중심 시스템과는 구조적으로 이질적이기 때문에, 운영체제 및 소프트웨어 계층과의 통합이 쉽지 않다. 개발자 입장에서는 새로운 API, 드라이버, 실행 모델을 학습해야 하며, 이를 지원하는 툴체인도 아직 미성숙한 단계에 머물러 있다. 또한, 연산 기능이 내장된 저장장치는 보안 이슈를 동반하며, 하드웨어 오류가 시스템 전반으로 확산될 위험도 높다. 결국, 하드웨어 설계뿐 아니라 소프트웨어와 시스템 아키텍처 전반에서의 재설계가 함께 요구된다.

산업계에서는 이미 관련 제품과 연구가 활발히 진행되고 있다. NGD Systems, Samsung, Kioxia 등은 계산 가능한 SSD 제품을 상용화했으며, AI inference, 고속 데이터 압축, 검색 최적화 등에 활용되고 있다. 특히 최근 주목받고 있는 CXL(Compute Express Link) 기술과의 융합 가능성은, 고속

I/O와 공유 메모리 구조를 가능하게 하여 컴퓨테이셔널 스토리지의 성능과 확장성을 한층 끌어올릴 수 있을 것으로 기대된다. 실제로 일부 연구에서는 메모리 계층 전반을 아우르는 연산 구조로서 컴퓨테이셔널 스토리지를 PIM(Processing-In-Memory)의 실용적 대안으로 간주하고 있다.

결론적으로, 컴퓨테이셔널 스토리지는 단순한 저장 공간 개념을 넘어, 데이터 중심 시대의 새로운 시스템 설계 철학을 반영하는 기술이다. 연산과 저장을 분리하던 전통적 방식에서 벗어나, 데이터를 저장하는 순간부터 바로 처리하는 ‘스마트 스토리지’로의 전환은 피할 수 없는 흐름이 되고 있다. 향후 해당 기술의 확산과 실용화는 소프트웨어 표준화, 개발 생태계 성숙도, 그리고 아키텍처 전환 속도에 달려 있을 것이다.

개인적으로는 이 기술이 미래 지향적인 구조임에는 분명하지만, 현재 상태로는 아직 실용화까지 갈 길이 멀다고 느꼈다. 기술 자체는 매력적이지만, 실질적인 사용을 위해 필요한 소프트웨어 스택의 미성숙, 복잡한 프로그래밍 모델, 그리고 운영체제 및 애플리케이션의 통합 어려움은 분명한 진입장벽이 된다. 특히 개발자 입장에서 보면 완전히 새로운 방식으로 사고하고 코드를 작성해야 하며, 현재 사용 중인 도구나 프레임워크와의 호환성도 낮다. 이러한 현실은 결국 기업이나 개인이 기술을 도입하는 데 있어 추가적인 비용과 리스크로 작용할 수밖에 없다.

또한, 저장장치가 연산을 수행한다는 점은 새로운 보안 취약점을 만들어낼 수 있는 가능성도 내포하고 있다. 데이터가 저장된 채로 연산되면서, 공격자가 침투할 수 있는 경로가 늘어나고, 하드웨어 수준에서의 오류가 전체 시스템에 영향을 미칠 가능성도 있다. 스마트해질수록 더 정교한 보안 체계가 필요하다는 점에서, 보안 문제는 단순히 ‘기능’이 아닌 ‘신뢰성’과 직결된 문제라고 생각한다.

그럼에도 불구하고 컴퓨테이셔널 스토리지는 장기적으로 AI·빅데이터·클라우드 컴퓨팅 환경에서 획기적인 자원 활용 방식을 제공할 수 있으며, 특히 엣지 컴퓨팅이나 실시간 응답성을 요구하는 시스템에서 먼저 자리를 잡을 가능성이 높다고 본다. 이러한 기술이 진정으로 확산되기 위해서는 단순한 성능 향상이 아닌, 개발자와 사용자가 ‘쉽게 쓸 수 있도록 만드는 노력’이 병행되어야 한다. 기술 발전의 속도 못지않게 생태계와 표준의 성숙도가 향후 기술 확산의 핵심 열쇠가 될 것이다.

---

## 참고문헌

[1] Tong Zhang et al., “What is Computational Storage?”, *IEEE Computer Architecture Letters*, vol. 20, no. 1, pp. 61–64, 2021. DOI: <https://doi.org/10.1109/LCA.2021.3050292>

[2] Vivek Seshadri et al., “Computational storage: An overview and research roadmap”, *Procedia Computer Science*, vol. 201, pp. 538–545, 2022. DOI: <https://doi.org/10.1016/j.procs.2022.03.068>