

Chapter
04연산 스토리지 기반 스프레드
스케줄러

김영환_한국전자기술연구원 팀장

I. 결과물 개요

개발목표시기	2023. 12.	기술성숙도 (TRL)	개발 전	개발 후
			4	7
결과물 형태	SW-Platform	검증방법	자체검증, 시험인증	
Keywords	쿼리 오프로딩, 연산스토리지, Query Offloading, CSD			
외부기술요소	100% 개발기술, Open Source 사용	권리성	특허, SW-IP, 설계서	

II. 기술의 개념 및 내용

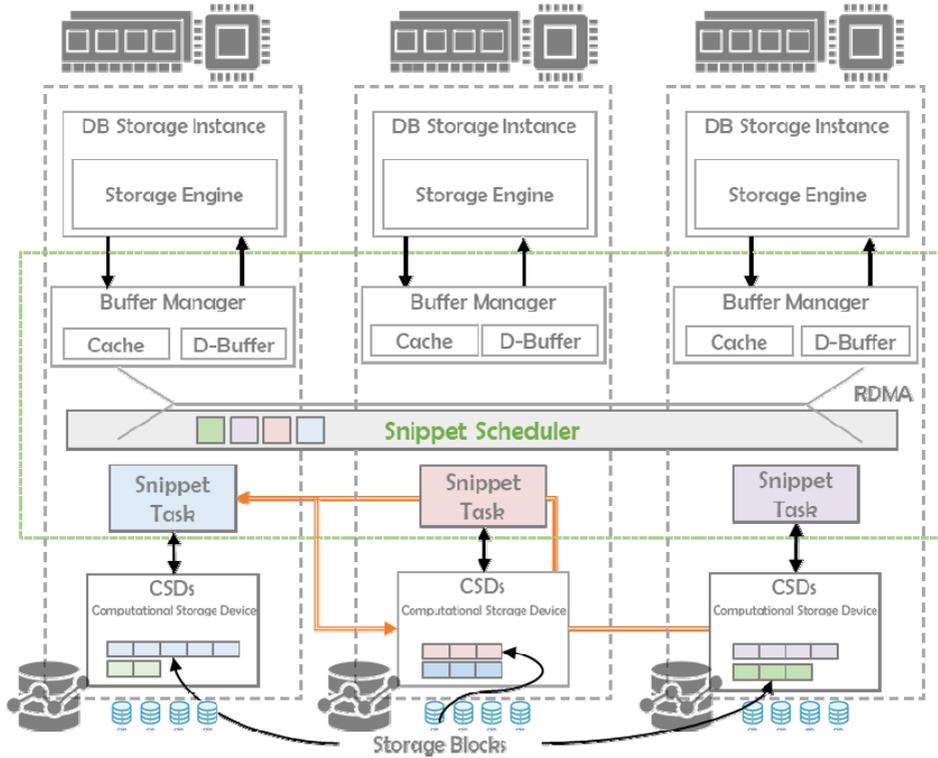
1. 기술의 개념

- 클라우드 네이티브(Cloud-Native) DBMS 스토리지 개념의 핵심은 클라우드 서비스의 탄력성 및 자원 확장성에 제약이 있는 기존의 독립형(Stand-Alone) DBMS 스토리지 기술의 한계를 극복하기 위해 컴퓨팅과 스토리지가 서로 분리(Disaggregation)된 클라우드 네이티브 환경에서 데이터베이스의 대용량 데이터를 효율적으로 저장하고 관리하

* 본 내용은 김영환 팀장(☎ 031-789-7550, yhkim93@keti.re.kr)에게 문의하시기 바랍니다.

** 본 내용은 필자의 주관적인 의견이며 IITP의 공식적인 입장이 아님을 밝힙니다.

***정보통신기획평가원은 현재 개발 진행 및 완료 예정인 ICT R&D 성과 결과물을 과제 종료 이전에 공개하는 "ICT R&D 사업화를 위한 기술예고"를 2014년부터 실시하고 있는 바, 본 칼럼에서는 이를 통해 공개한 결과물의 기술이전, 사업화 등 기술 활용도 제고를 위해 매주 1~2건의 관련 기술을 소개함



[그림 1] 연산 스토리지 기반 스니펫 스케줄러 개념도

기 위한 스토리지 기술임

- (연산 스토리지) 호스트 CPU에서 담당하던 연산 처리의 일부를 스토리지에서 분담해서 처리하는 아키텍처로 기존의 방식에서는 호스트에서 스토리지에 해당 데이터를 불러와서 처리하였으나 연산 스토리지는 연산 할당량을 스토리지로 오프로드하여 불러오는 데이터의 양을 절감함으로써 성능 향상 및 소비전력 절감이 가능
- DBMS 태스크 연산의 고속 오프로딩 및 클라우드형 데이터 연산을 위한 스토리지 엔진 스니펫 스케줄러 기술개발

III. 국내외 기술 동향 및 경쟁력

1. 기술의 특성 및 성능

- 컴퓨팅과 스토리지가 분리된 차세대 DBMS 기술은 국내 토종 DBMS와 클라우드 사업자들이 차기 클라우드 기술로 많은 관심을 가지고 있으나 단일 및 독립형 DBMS 구조이거나 클라우드에서 단순히 컴퓨팅 인스턴스에 DB를 설치하여 서비스하는 정도이며, 연산 스토리지 기술에 대한 연구는 전무하고 대부분 Google, AWS 등 외산 솔루션을 활용한 서비스 개발에 집중
- 세계적으로 클라우드 네이티브 기반 Disaggregated 데이터베이스 스토리지 기술에 대한 관심이 높아지고 있는 가운데 글로벌 클라우드 제공업체들은 자사 데이터센터의 네트워크 트래픽을 최소화하고 이에 따른 소비전력 절감을 위한 연산 스토리지 기술 개발에 집중하고 있음
- 스토리지 장치들은 DBMS와 물리적으로 분리되어 있는 스토리지 서버에 구축되어 있고, 스토리지 엔진은 이러한 원격 스토리지를 RDMA와 같은 고속 네트워크를 통해 접근하는 분리 환경에서 DBMS의 스토리지 엔진을 통해 원격 접근이 가능한 연산 스토리지 기술 개발을 통해 기존의 독립형(Stand-Alone) DBMS 스토리지 기술의 한계를 극복하는 것이 가능

2. 경쟁기술/대체기술 동향 및 현황

- (한국전자기술연구원) “SW 컴퓨팅 산업원천기술개발 사업(Global Creative SW: GCS)”인 “1,000 Core 이상 Scale Out 가능한 클러스터 데이터베이스 플랫폼 개발” 과제에서 DB 테이블 저장 성능 향상을 위해 스토리지 엔진의 블록(페이지) 입출력 가속 및 캐시 기술을 개발하여 다양한 스토리지 플랫폼으로 활용 중
- (글루시스) 옛지용 연산 스토리지 연구 개발 진행으로 Key/Value API를 지원하는 NVMeoF 기반 연산 스토리지 및 2020년부터 자사 AnyStor Enterprise 기반 스케일아웃 스토리지에 SNIA 표준 KV API를 지원하기 위한 연구 개발을 진행하고 있으며 2023년 상용화를 목표로 함

- (오픈시스템) “NFV 기반 가상화 데이터베이스 상에 대용량 로그 관리를 통한 비정상 탐지 시스템 개발”을 통해 클라우드 환경 NFV 가상화 플랫폼 상에 Scale In/Out이 가능한 비정형 데이터베이스 개발을 수행하였으며, 이를 통해 IoT 단말, 통신시스템 및 데이터센터에서 발생하는 다양한 비정형 로그 데이터를 수집하고 실시간 분석을 통한 비정상 탐지를 수행하는 시스템 개발을 완료

3. 우수성 및 차별성

경쟁기술	본 기술의 우수성/차별성
라운드로빈, 랜덤 스케줄링 구조	스니펫 스케줄러 - Multi-Queue 기반 다중 Offload 요청 처리 - CSD 내 데이터 위치(단일/분산) 기반 스니펫 Task 생성 및 데이터 병합 - CSD 내 가속 엔진 적용/비적용 플래그 처리(가속 엔진 적용 효율 극대화)

4. 표준화 동향

- 클라우드 자원관리 상호운용성(호환성)을 위한 정보모델, API 등의 표준으로 국내에서는 TTA SPG21에서 표준화를 추진 중
- 단독 클라우드 인프라 환경뿐만 아니라 인터클라우드 환경에서의 시스템/네트워크 보호를 위한 표준으로 TTA SPG21(클라우드), TTA PG504(보안) 등에서 국내 표준화를 진행 중
- 스토리지 관련 사실 표준화 기구인 SNIA(Storage Networking Industry Association)에서는 2012년 6월에 CDMI 표준을 제정하여, 클라우드 환경에서 사용되는 데이터들의 생성, 저장, 갱신, 삭제 등의 데이터 관리에 필요한 인터페이스를 표준화하였으며, CDMI 표준 역시 ISO/IEC JTC1의 표준(ISO/IEC 17826:2012)으로도 제정되었음
- 클라우드 자원관리 상호운용성(호환성)을 위한 정보모델, API 등의 표준으로 OCF(Open Connectivity Foundation)에서 관련 국제 표준을 개발 중

IV. 국내외 시장 동향 및 전망

1. 국내외 시장 동향 및 전망

- ▶ 중소 SW기업과 같은 소규모 조직에서도 빠른 서비스 개발, 런칭을 위해 데브옵스(DevOps) 개발 방법론을 채택하면서 백엔드 서비스 제공을 위한 클라우드형 DBMS 구축 수요가 증가하고 있고, 국내에서도 클라우드 상에서 웹 기반의 앱 서비스를 제공하는 SKT, SK C&C, 메가존, 이노그리드 등이 클라우드 기반 백엔드 서비스로 고성능 DBMS 서비스 제공을 위해 노력중임
- ▶ 글로벌 클라우드기반 DBMS 서비스를 제공하고 있는 Google(GKE/GCE), AWS(ECS), MicroSoft(ASC/AKC), IBM, 액센추어, SAP 등을 중심으로 서비스 수요가 발생

2. 제품화 및 활용 분야

활용 분야(제품/서비스)	제품 및 활용 분야 세부내용
연산 DBMS 스토리지 엔진	DBMS 엔진과 스토리지가 분리 Disaggregation된 클라우드 환경에서 방대한 데이터를 분석/추론함에 있어 연산 스토리지 Computational Storage를 기반으로 성능 및 에너지 절감을 이루기 위한 차세대 DBMS 입출력 SW 기술에 적용 및 활용 가능

V. 기대효과

1. 기술도입으로 인한 경제적 효과

- ▶ 차세대 DBMS 연산 스토리지 기술은 대규모의 컴퓨팅 인프라 기반의 신규 서비스 개발 및 유통을 가속화할 수 있는 기반 기술로, 인공지능, 빅데이터, 지능정보 산업 육성을 통해 국내뿐만 아니라 글로벌 시장 창출 및 선점 효과가 기대됨
- ▶ 국내 토종 DBMS기업이 보유한 제품의 경우 기존 모놀리식 구축형 DBMS로 컴퓨팅과 스토리지가 분리된 클라우드형 차세대 DBMS로 전환할 수 있는 기반을 마련할 수 있고 차별화된 기술로 시장 경쟁력 확보가 가능

2. 기술사업화로 인한 파급효과

- ▶ 사업 기간 중 발생한 연구결과의 공개 SW 라이선스 전략을 통해 산업계의 관련기술 활용에 관한 저변을 확대하고 관련시장 확산을 유도
- ▶ DBMS에 연동한 스토리지 기술에 대한 기술력 확보로 침체되어 있던 국내 서버 및 스토리지 업계 활성화가 기대됨
- ▶ 기존 국내 서버 및 스토리지 업계에 만연하던 해외 벤더 종속성 탈피로 국내 수요자는 기술지원, 공정가격 등에서 혜택을 볼 것으로 기대됨