상태 저장 서버리스 컴퓨팅을 위한 분산 공유 로그 시스템에 대한 연구 동향

추교현, 김경연, 박상오
중앙대학교

khchoo@cslab.cau.ac.kr, ky0207@cau.ac.kr, sopark@cslab.cau.ac.kr

A Study on Distributed Shared Logs System for Stateful Serverless Computing

Choo Kyo Hyun, Kim Kyeong Yeo, Park Sang Oh
Chung-Ang Univ.

요 약

서버리스 컴퓨팅 환경에서 상태 저장 워크로드를 지원하기 위한 방안으로 분산 공유 로그가 주목받고 있다. Boki, FlexLog, IndiLog 세 가지 주요 시스템의 발전 과정을 통해 분산 공유 로그 기반 상태 저장 서버리스 시스템의 연구 동향을 분석한다. Boki 는 메타로그를 통해 서버리스 컴퓨팅을 위한 공유 로그 런타임을 최초로 제안했다. FlexLog 는 영구 메모리와 유연한 순서 보장 프로토콜을 도입하여 Boki 의 스토리지 및 순서 보장 계층의 성능 병목을 해결했다. 나아가 IndiLog 는 컴퓨팅 계층과 인덱스를 분리하는 인덱싱 아키텍처를 제시하여 동적 확장성과 자원 안정성을 확보했다. 이러한 시스템들의 진화 과정은 서버리스 환경에 최적화된 분산 공유 로그 시스템이 고성능 하드웨어의 도입을 넘어, 각 구성 요소를 분리하고 전문화하는 아키텍처로 발전하고 있음을 시사한다.

Ⅰ. 서 론

서버리스 컴퓨팅은 개발자가 인프라 관리 부담 없이 코드를 실행할 수 있게 해주는 패러다임으로 자리 잡았다. 그러나 서버리스 환경 위에서 구축되는 애플리케이션들은 점차 복잡한 상태를 관리해야 하는 상태 저장 워크로드가 필요하다. 이러한 불일치는 상태 저장 서버리스 시스템이 해결해야 할 도전 과제로 부상했다.

현재 애플리케이션은 상태 관리를 위해 범용 클라우드 스토리지 서비스에 의존한다. 하지만 이런 서비스들은 서버리스 워크로드가 요구하는 낮은 지연 시간, 강력한 일관성, 그리고 내결함성을 동시에 만족시키기 어렵다. 특히 여러 함수로 구성된 워크플로우에서 특정 함수가 중간에 실패할 경우 애플리케이션의 전체 상태가 불일치 될 위험이 존재한다.

이러한 문제에 대한 해결책으로 분산 공유 로그가 주목받고 있다. 분산 공유 로그는 전체적인 로그의 순서가 보장되고 내결함성을 가지는 레코드의 시퀀스를 제공하는 메커니즘이다. 이 모델은 상태 기계 복제(State Machine Replication, SMR)를 효율적으로 지원하며 개발자가 복잡한 합의 프로토콜을 직접 관리할 필요 없이 강력한 일관성과 내결함성을 달성할 수 있게 된다.

본 논문은 Boki [1], FlexLog [2], IndiLog [3] 라는 세 가지 분산 공유 로그 시스템을 중심으로 공유 로그 기반 상태 저장 서버리스 시스템의 연구 동향을 분석한다. 이 시스템들의 발전 과정은 단순히 개별적인 개선의 나열이 아니라, 서버리스 환경에 공유 로그를 적용하는 과정에서 발생하는 병목 지점을 식별하고 해결해 나가는 방향을 보여준다. 각 시스템의 핵심 기여와 한계를 분석하고, 이를 바탕으로 미래 연구 방향을 제시하고자 한다.

Ⅱ. 관련 연구

Boki [1]는 상태 저장 서버리스 컴퓨팅을 위해 특별히 설계된 분산 공유 로그 런타임을 최초로 제안한 시스템이다. Boki 의 아키텍처는 Nightcore [4] 라는 FaaS 런타임 위에 구축되었으며, 서버리스 함수가 실행되는 노드에 LogBook 엔진을 함께 배치하는 구조를 특징으로 한다. 이 엔진은 로그의 모든 레코드 위치를 찾기 위한 로그 인덱스를 메모리에 유지하며 읽기 지역성을 향상시키기 위한 레코드 캐시를 관리한다. 이러한 설계는 인덱스 읽기를 항상 로컬에서 처리함으로써 서버리스 함수가 낮은 읽기 지연 시간을 달성하도록 한다.

Boki 에서 쓰기 요청이 발생하면 LogBook 엔진은 해당 데이터를 함수가 실행된 노드와 매핑된 스토리지 노드의 특정 샤드에 기록한다. 각 스토리지 노드는 변경된 로그들을 시퀀서 노드에 보고하고 주요 시퀀서는 이 정보를 취합하여 메타로그(Metalog)에 추가함으로써 로그의 전체 순서를 확정한다. 각 LogBook 엔진은 메타로그를 구독하여 자신의 로컬 인덱스와 캐시를 비동기적으로 갱신한다.

기존의 Boki 시스템은 스토리지 계층을 SSD 에 최적화된 RocksDB 엔진 위에 구축한다. SSD 기반 아키텍처는 내구성을 보장하기 위해 운영체제의 동기화 시스템 콜에 의존하게 되는데, 이에 따라 Boki 는 1~3ms 수준의 높은 읽기 지연 시간을 보이는 한계가 있었다. 이를 해결하기 위해 FlexLog [2]는 영구 메모리(Persistent Memory, PM)를 저장 매체로 채택했다. FlexLog 는 최근 접근한 데이터를 위한 DRAM 캐시, 충돌 일관성을 보장하며 데이터를 저장하는 PM 그리고 오래된 데이터를 SSD 로 옮기는 계층적 저장소 아키텍처를 구현한다. PM 의 낮은 지연 시간을 활용하여 FlexLog 는 Boki 의 저장소 계층보다 더 빠른 처리량을 달성했다.

또한, Boki 는 모든 로그에 대해 전역적으로 단일 전체 순서 보장을 위해 Paxos 합의 프로토콜을 사용한다. 이 방식은 강력한 일관성을 제공하지만, 모든 요청마다 값비싼 Paxos 프로토콜을 실행해야 하므로 성능 병목이 된다. FlexLog 는 유연한 순서 보장을 제공하는 확장 가능한 아키텍처를 도입했다. FlexLog 의 순서 보장 계층은 트리 구조의 시퀀서 노드들로 구성된다. 각 시퀀서는 자신이 담당하는 '색상' 내에서만 전체 순서를 보장하며 서로 다른 색상의 로그는 병렬적으로 처리될 수 있다. 애플리케이션은 필요에 따라 단일 색상으로 기록하여 강력한 일관성을 얻거나, 여러 색상으로 기록하여 느슨한 일관성으로 성능을 높일 수 있다. 이러한 프로토콜 덕분에 FlexLog 는 Boki 의 순서 보장 계층보다 2~4 배 더 낮은 지연 시간을 달성했다.

Boki 는 모든 컴퓨팅 노드가 전체 로그에 대한 인덱스를 로컬 RAM 에 저장해야 한다는 구조적 한계를 가진다. 이는 함수 인스턴스가 급증하여 새로운 컴퓨팅 노드를 추가할 때, 새 노드는 인덱스가 없어 기존 노드에 원격으로 조회해야만 한다. 이 과정에서 발생하는 리소스 경합은 시스템 전체의 성능을 저하시키고 서버리스 환경의 핵심인 확장성을 저해한다. IndiLog [3]는 컴퓨팅 계층과 인덱스를 분리하고 항상 가동하는 전용 인덱스 계층을 도입하여 이 문제를 해결했다. 새로운 컴퓨팅 노드는 인덱스 정보가 필요할 때 다른 컴퓨터 노드가 아닌 전용 인덱스 계층에 질의하므로, 노드 간 리소스 경합 없이 동적 확장이 가능해진다.

또한, Boki 의 로컬 인덱스는 전체 로그 크기에 비례하여 결국 컴퓨팅 노드의 메모리를 고갈시키고 OOM(Out-of-Memory) 오류로 시스템을 충돌시키는 안정성 문제를 야기했다. IndiLog 는 컴퓨팅 노드에 엄격하게 제한된(size-bounded) 로컬 인덱스를 유지하는 방식으로 이 문제를 해결했다. 이 로컬 인덱스는 20MB 정도로 크기가 제한되어 OOM 문제를 방지하고 시스템 안정성을 확보한다.

Ⅲ. 결론

본 논문은 분산 공유 로그 기반의 상태 저장 서버리스 시스템 연구 동향을 Boki, FlexLog, IndiLog 세 시스템을 중심으로 분석했다. 이들의 발전 과정은 서버리스 환경이 요구하는 낮은 지연 시간, 높은 처리량, 그리고 동적 확장성을 달성하기 위해 시스템의 병목 지점을 순차적으로 해결해 나가는 뚜렷한 흐름을 보여준다.

Boki 는 메타로그 기반의 분산 공유 로그 런타임을 제시하며 해당 연구 분야의 기틀을 마련했다. 하지만 SSD 기반 스토리지의 I/O 지연 시간, Paxos 프로토콜 기반의 유연하지 않은 순서 보장 등으로 인해 확장성 측면에서 한계를 드러냈다.

FlexLog 는 이러한 Boki 의 한계 중 스토리지와 순서 보장 계층의 성능 문제에 집중했다. 영구 메모리를 활용한 계층적 스토리지로 I/O 병목을 해결하고, 트리 구조의 시퀀서를 통한 유연한 순서 보장으로 프로토콜 오버헤드를 줄여 Boki 대비 더 나은 성능을 달성했다.

또한, IndiLog 는 컴퓨팅 계층과 인덱싱을 완전히 분리하는 아키텍처를 통해 Boki 의 인덱싱으로 인한 확장성 한계를 극복했다.

결론적으로, 분산 공유 로그 기반 서버리스 시스템의 연구 동향은 Boki 에서 출발하여 성능 최적화를 위한 하드웨어 및 프로토콜 개선한 FlexLog 와 확장성을 위해 컴퓨팅 계층과 인덱스를 분리한 IndiLog 로 진화하고 있다.

향후 연구는 분리된 아키텍처를 기반으로 각 계층을 더욱 정교하게 최적화하고, 다양한 워크로드에 동적으로 적응하는 자율적인 시스템을 구현하는 방향으로 나아갈 것으로 전망된다.

ACKNOWLEDGMENT

"본 연구는 2025 년 과학기술정통신부 및

정보통신기획평가원의 SW 중심대학사업 지원을 받아

수행되었음"(2025-0-00032)

참 고 문 헌

[1] Zhipeng Jia and Emmett Witchel. 2021. Boki: Stateful Serverless Computing with Shared Logs. In ACM SIGOPS 28th Symposium on Operating Systems Principles (SOSP '21).

[2] Dimitra Giantsidi, Emmanouil Giortamis, Nathaniel Tornow, Florin Dinu, and Pramod Bhatotia. 2023. FLEXLOG: A Shared Log for Stateful Serverless Computing. In Proceedings of the 32nd International Symposium on High-Performance Parallel and Distributed Computing (HPDC '23).

[3] Maximilian Wiesholler, Florin Dinu, Javier Picorel, and Pramod Bhatotia. 2024. IndiLog: Bridging Scalability and Performance in Stateful Serverless Computing with Shared Logs. In SYSTOR '24.

[4] GitHub. n.d. ut-osa/nightcore: Nightcore: Efficient and Scalable Serverless Computing for Latency-Sensitive, Inter-active Microservices. Retrieved April 15, 2022 from https://github.com/ut-osa/nightcore