

서버리스 컴퓨팅 환경에서 FIFO와 CFS를 결합한 하이브리드 스케줄러 연구

컴퓨터정보공학부
2021202029 이지연

Part 1

서버리스 컴퓨팅이란?

서버 관리의 추상화

이벤트 기반
실행 모델

사용한 만큼만
지불하는 과금 방식

FaaS (Function as a Service) / BaaS (Backend as a Service)

서버리스 컴퓨팅의 주요 장점

서버리스 환경의 특징 및 과제

짧은 실행 시간

높은 동시성 요구

빈번한 함수 호출&종료

기존 OS 스케줄러(CFS)가
서버리스 환경에
최적화되지 않음

잘못된 컨텍스트 스위칭으로 인한
성능 저하 및 비용 증가

Part 2

OPTIMIZED HYBRID SCHEDULER

최적화된 하이브리드 스케줄러

FIFO

짧은 함수를
빠르게 처리
오버헤드 최소화

결
합

CFS

긴 함수에 대한
공정한 자원 할당



QUESTION.

왜 CFS는 서버리스에
비효율적인가?

Part

왜 CFS는 서버리스에 비효율적인가?

CFS(Completely Fair Scheduler)?

Serverless workloads에서의
Overhead

잡은 컨텍스트 스위칭의 부담

짧은 작업 완료를 지연시키는 요인

CFS 사용 시 비용이
최대 10배까지 증가할 수 있음

In Serverless, OS Scheduler Choice Costs Money:
A Hybrid Scheduling Approach for Cheaper FaaS[†]

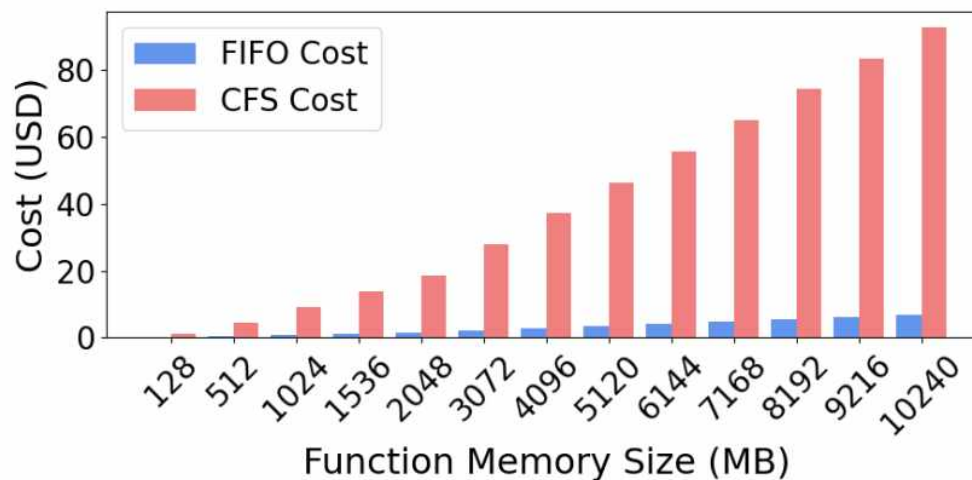


Figure 1: Cost for FIFO and CFS OS scheduling policies calculated using AWS Lambda pricing. The workload is using the first 12,442 functions in the Microsoft Azure trace. Although FIFO cost is significantly lower, it introduces unacceptably large latencies for functions that simply wait in queues. We explore this trade-off in the remainder of the article.

효율적인 자원 관리를 위한 설계

1

CPU 코어 그룹 분할: FIFO 그룹과 CFS 그룹 으로 분리

FIFO 그룹과 CFS 그룹으로 분리하여, 각 그룹은 짧은 작업과 긴 작업에 특화된 스케줄링 정책을 독립적으로 적용하여 오버헤드를 줄이고 효율성을 높임. 이를 통해 시스템은 다양한 성격의 서버리스 워크로드를 효과적으로 처리할 수 있게 됨.

2

동적 함수 마이그레이션

짧은 함수는 FIFO 그룹에서 우선 실행되어, 도착 순서대로 빠르게 처리되며 불필요한 컨텍스트 스위칭 없이 작업을 완료하여 응답성을 향상시킴. 이는 대부분 짧게 실행되는 서버리스 함수의 특성에 맞춰 성능을 최적화하는 핵심 전략임.

3

일정 시간 초과 시 CFS 그룹으로 이동

FIFO 그룹에서 특정 시간 이상 실행되는 함수는 CFS 그룹으로 자동 이동되어, 장기 실행 작업에 대한 공정성을 확보하고 시스템 전체의 안정성을 유지함. 이 메커니즘은 FIFO 그룹이 긴 작업으로 인해 블록되는 현상을 방지함.

4

적응형 시간 제한 관리

최근 함수 실행 시간 데이터를 기반으로 FIFO 그룹의 시간 제한을 동적으로 조정하여, 워크로드 변화에 실시간으로 대응하고 스케줄링 효율성을 지속적으로 최적화함. 이는 예측 불가능한 서버리스 환경에서 안정적인 성능을 제공하는 데 중요한 역할을 함.

Part 2

하이브리드 스케줄러의 우수성

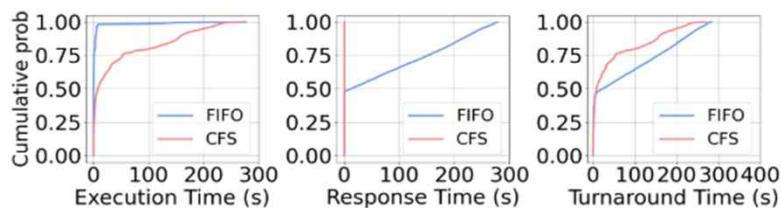


Figure 4: Metrics comparison between FIFO and CFS. FIFO policy achieves good execution time but sacrifices latency.

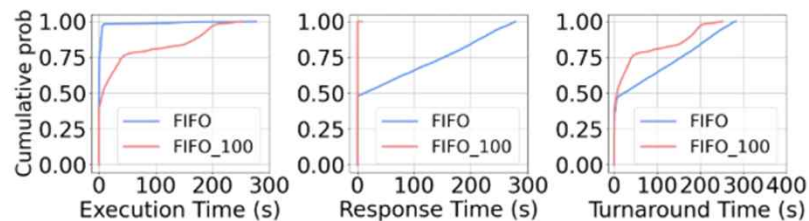


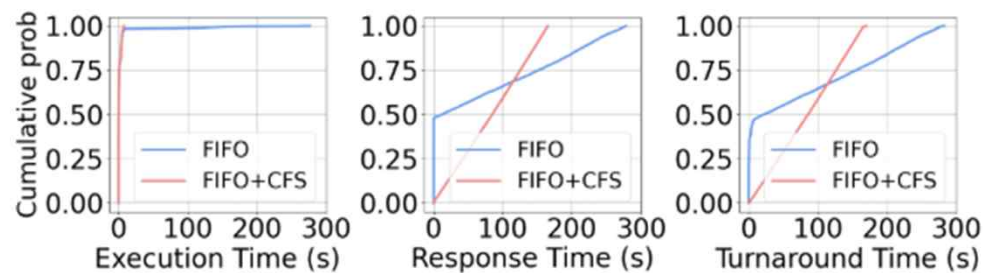
Figure 5: Metrics comparison between FIFO policy and FIFO policy with 100ms preemption. Preemption improves response time at the cost of increasing execution time.

A

실험환경 요약:
컨테이너 및 마이크로VM 환경
주요 성능 지표: 실행 시간,
작업 완료 시간(Turnaround Time), 비용

B

실험 결과:
CFS 대비 전반적인 실행 시간 단축
작업 완료 시간 감소
서버리스 함수 사용자 비용 절감 효과 확인

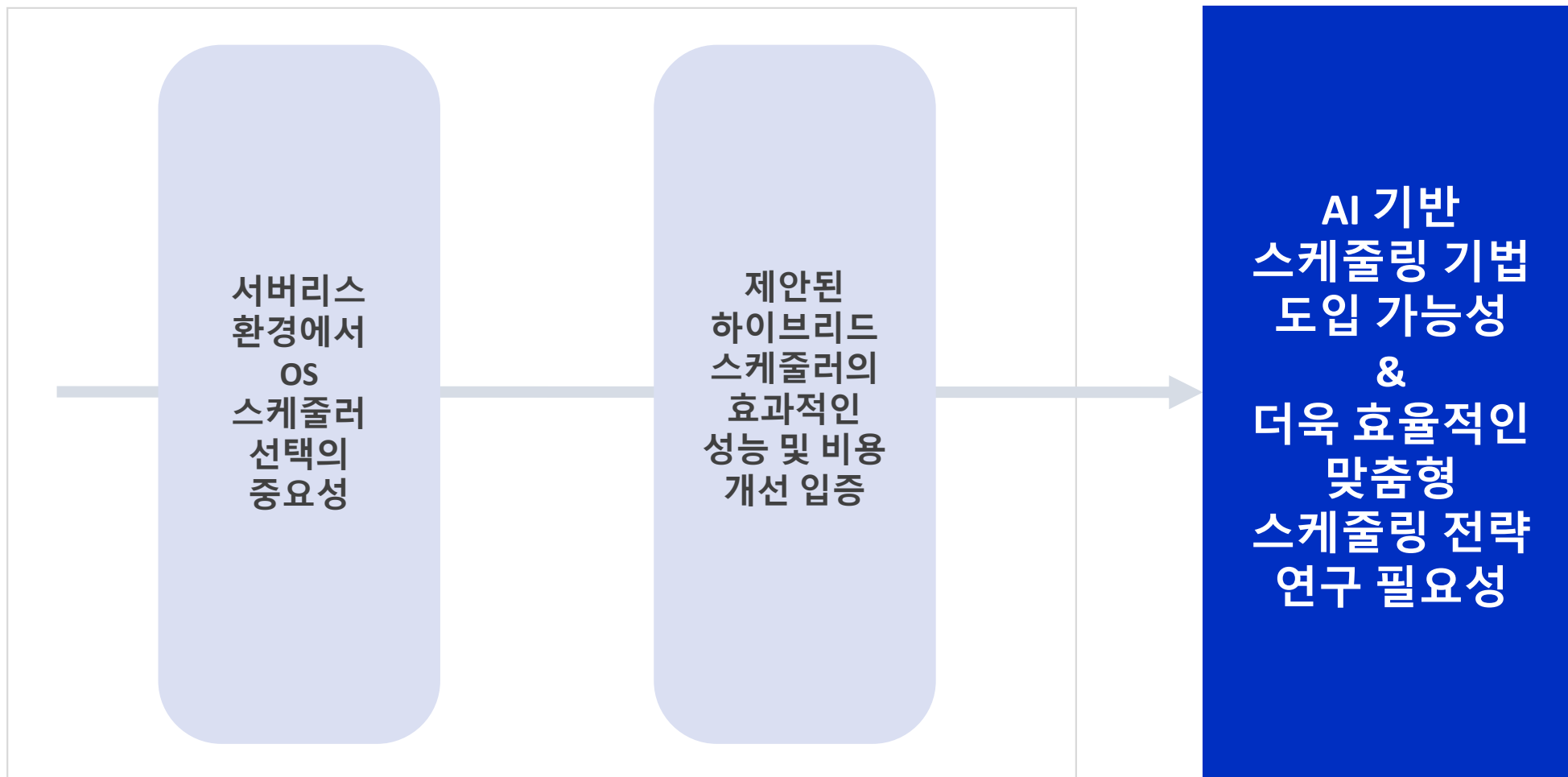




Part 3

결론 및 향후 전망

서버리스 컴퓨팅의 미래를 위한 제안



Q&A

