**클라우드–엣지–IoT(Cloud–Edge–IoT) 통합 구조에서의 자원 관리 시스템 연구**

국민대학교 / 소프트웨어융합대학원 / 인공지능응용

K2025029 금동환

**1) 배경**

클라우드–엣지–IoT(Cloud–Edge–IoT, 이하 CEI) 환경은 이기종 자원과 가변적인 무선·백홀 상태로 인해, 정적 정책으로는 서비스 수준 목표(Service Level Objective, SLO)를 일관되게 보장하기 어렵습니다. 특히 지연·지터·대역폭·비용·에너지·가용성이 얽힌 복합 트레이드오프가 존재하며, 컨테이너 오케스트레이션(Kubernetes, K8s)은 스케일링은 우수하지만, 계층 간 오프로딩·압축·캐싱·경로 결정을 통합적으로 다루지 못합니다. 본 연구는 SLO 인지형 정책엔진과 강화학습(Reinforcement Learning)·밴딧(Multi‑armed Bandit), 혼합 유향 비순환 그래프(Directed Acyclic Graph, DAG) 실행을 결합하여 CEI 전 과정을 자동화하는 프레임워크를 제안합니다.

**2) 기여**

• CEI 전과정(수집→정화→추론/분석→저장/옵스) 자동화 아키텍처와 참조 구현 제공.

• SLO 인지형 멀티티어 스케줄러와 적응형 오프로딩/압축/캐싱 정책을 결합한 정책 엔진 설계.

• 혼합 스트림/배치 DAG 실행의 분할·배치·데이터 이동 비용을 수식화하고 경량 근사 해법 제시.

• 재현 가능한 테스트베드·측정 스크립트·평가 매트릭스를 공개하여 실증 가능성 강화.

• 조건별 운영 가이드라인(튜닝 파라미터·정책 가드레일·롤백 절차) 정량 제시.

**3) 참고 문헌**

• Satyanarayanan et al., “The Case for VM‑Based Cloudlets in Mobile Computing,” IEEE Pervasive Computing, 2009.

장점: 사용자 근접 소형 데이터센터(Cloudlet) 개념 정립으로 근접 연산 가치 선도.

단점: 비전 성격이 강해 대규모 정량 비교·다양한 워크로드 실험은 제한.

• Shi et al., “Edge Computing: Vision and Challenges,” IEEE Internet of Things Journal, 2016.

장점: 엣지 컴퓨팅 개념·아키텍처·연구과제를 포괄 정리한 대표 서베이.

단점: 표준 KPI 기반의 아키텍처 교차 실험 비교는 부족.

• Mao et al., “A Survey on Mobile Edge Computing,” IEEE Communications Surveys & Tutorials, 2017.

장점: 통신 관점 오프로딩·자원관리 분류 체계화, 고피인용 서베이.

단점: Cloudlet/Fog/연합 아키텍처 간 실험적 교차 비교·오케스트레이션 지연 논의는 약함.

• Chiang & Zhang, “Fog and IoT: Research Opportunities,” IEEE Network, 2016.

장점: Fog‑IoT 융합의 대역폭 절감·근원지 전처리 가치와 연구 기회 체계화.

단점: 응용별 KPI·운영 자동화의 정량 평가 부족.

• Li et al., “Computation Offloading and Resource Allocation,” IET Communications, 2019.

장점: 오프로딩·자원할당 결합 최적화를 수학적으로 정식화.

단점: E2E 오케스트레이션 지연·이동성 시나리오 실증 제한.

• Kong et al., “AccuMO,” ACM MobiCom, 2023.

장점: 정확도 중심 멀티태스크 오프로딩으로 AR 정확도–지연 트레이드오프 개선.

단점: 특정 태스크/디바이스 중심이라 범용 워크로드·다양한 인프라 일반화 필요.

**4) 제안하는 기법**

**4.1 아키텍쳐**

* 컨트롤 플레인: 정책 엔진, SLO(서비스 수준 목표, Service Level Objective) 스케줄러, 카탈로그, 텔레메트리.
* 데이터 플레인: 수집(MQTT(경량 메시징 프로토콜, MQTT)/HTTP/Kafka)–처리(Flink/Spark)–저장(TSDB(시계열 데이터베이스, Time Series Database)/객체저장소)–서빙(Ray/TF/ONNX).
* 보안/프라이버시: 인증/암호화/프라이버시 보존 처리(연합학습/차등프라이버시).

**4.2 핵심 모듈**

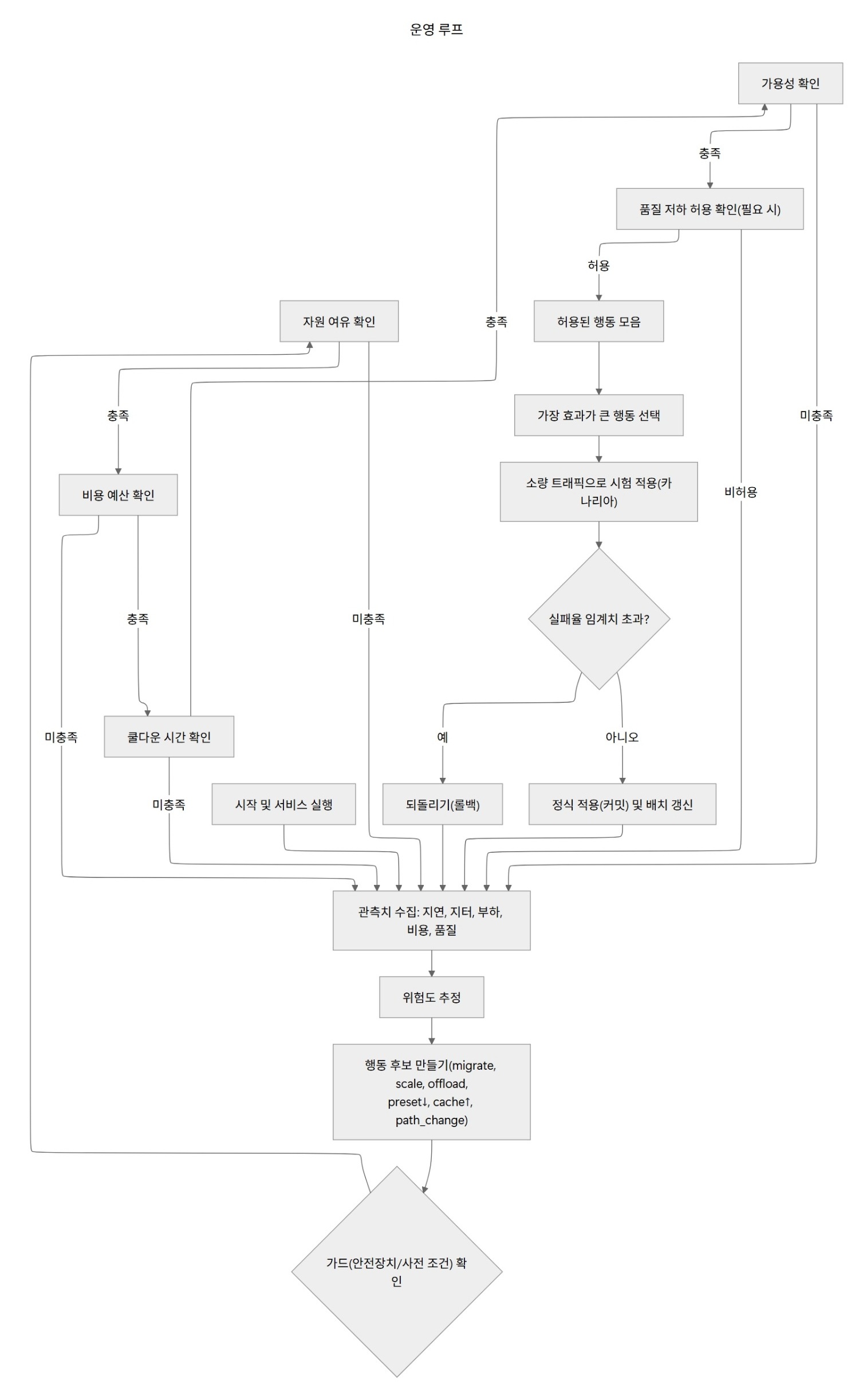
* 지능형 스케줄러(RL(강화학습, Reinforcement Learning)+휴리스틱): 지연/비용/정확도 가중 최적화.
* 적응형 오프로딩/압축/캐싱: 네트워크/부하/품질 신호 기반.
* DAG(유향 비순환 그래프, Directed Acyclic Graph) 컴파일러: 워크로드를 계층별로 분할 배치, 데이터 이동 비용 포함.
* 품질/옵스: 데이터/모델 품질 규칙, 자동 재학습/롤백/카나리아.

**4.3 상태/행동/보상(SAR) 표**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **항목** | **정의/예시** | **주석** |
| 상태 S | 네트워크 | RTT(왕복 지연, Round Trip Time), 손실, 지터, 대역폭 | OpenTelemetry/에뮬레이터 |
| 부하/자원 | CPU/GPU/메모리/IO/큐길이 | Prometheus/GPU-telemetry |
| 데이터 품질 | 결측/스큐/드리프트 지표 | 데이터 품질 엔진 |
| 모델 품질 | 정확도/PSNR(피크 신호대잡음비, PSNR)/VMAF(체감 품질 지표, VMAF)/지연 | 온라인/섀도우 평가 |
| 비용/전력 | 분당 과금/소비전력 | 요금 계산기/전력계 |
| 행동 A | 배치/마이그 | migrate(v,t→t') | 라이브/콜드 옵션 |
| 스케일 | scale\_out/in, GPU on/off | 쿨다운 필요 |
| 오프로딩 | offload(v, Dev↔Edge/Cloud) | 임계 기반 |
| 압축/프리셋 | encode\_preset↓, compression↑ | 지연↔품질 트레이드오프 |
| 캐싱/경로 | cache↑, QoS(서비스 품질, Quality of Service) path change | SDN(소프트웨어 정의 네트워킹, Software-Defined Networking)/우회경로 |
| 보상 R | 식 | R = -α/SLO\_viol - λ\_c/Cost - λ\_e/Energy + λ\_q/QoE(체감 품질, Quality of Experience) | 가중치 튜닝 |

**4.4 의사코드**

4.4.1 운영루프



4.4.2 정책관점

텍스트, 영수증, 도표이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

**4.5 성능 평가 계획**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **시나리오** | **워크로드** | **KPI (단위)** | **샘플 크기 n** | **반복 k** | **통계 검정** | **베이스라인 → 제안기법** |
| 영상 분석(프라이버시) | RTSP(실시간 스트리밍 프로토콜, Real-Time Streaming Protocol) 다중 스트림 + 객체탐지/마스킹 | E2E 지연(ms), SLO(서비스 수준 목표, Service Level Objective) 위반률(%), 백홀 절감(%) | n=30 세션/조건 | k=5 반복 | 혼합효과모형 + ANOVA(사후검정 Tukey), 95% CI | K8s(쿠버네티스, Kubernetes) 기본+고정 오프로딩 → SLO(서비스 수준 목표, Service Level Objective) 스케줄러+적응 오프로딩/압축/캐싱 |
| 예지 보전(시계열) | 센서 시계열 + 이상탐지 | 지연(ms), 처리량(req/s), 전력(W) | n=30 배치/조건 | k=5 반복 | 혼합효과모형 + t-test/ANOVA, 생존분석(중단시간) | KEDA(이벤트 기반 오토스케일러, Kubernetes Event-driven Autoscaling)/Knative(서버리스 실행 프레임워크, Knative) 기본 → SLO(서비스 수준 목표, Service Level Objective) 스케줄러+DAG(유향 비순환 그래프, Directed Acyclic Graph) 분할/캐싱 |
| 모바일 AR | 분할추론 + 인코딩 파이프라인 | 95%tile 지연(ms), QoE(VMAF(체감 품질 지표, VMAF)), 배터리(mWh) | n=30 세션/조건 | k=5 반복 | ANOVA + Cliff’s δ(효과크기), 95% CI | 고정 배치/프리셋 → 밴딧/RL(강화학습, Reinforcement Learning) 프리셋/마이그레이션 |
| 버스티/장애 복구 | 부하 급증/노드 장애 주입 | 복구시간(ms), 재시도(회), 손실율(%) | n=20 이벤트/조건 | k=5 반복 | 생존분석(Kaplan–Meier) + 로그랭크 테스트 | 기본 HPA(수평 파드 오토스케일, Horizontal Pod Autoscaler)/재시작 → 정책 가드레일+선제 스케일/마이그레이션 |

* 베이스라인: K8s(쿠버네티스, Kubernetes) 기본 스케줄링 + 고정 오프로딩/스케일 + KEDA(이벤트 기반 오토스케일러, Kubernetes Event-driven Autoscaling)/Knative(서버리스 실행 프레임워크, Knative).
* 지표: E2E 지연/지터, SLO(서비스 수준 목표, Service Level Objective) 위반률, 처리량, 비용/전력, 품질 지표, 복구시간, 수렴속도.
* 분석: 혼합효과/ANOVA/생존분석, 파레토 프런티어, 큐잉/코스트 모델, 민감도.

**4.6기대 결과**

* CEI(클라우드–엣지–IoT, Cloud–Edge–IoT) 자동화 프레임워크/코드/데이터
* 학술적 성과: 스케줄링/오프로딩/데이터 품질 통합 모델
* 테스트베드/KPI로 정량 평가 및 가이드, 레퍼런스

**5) Target Conference/Journal 및 일정**

**5.1 Target Conference/Journal**

|  |  |  |
| --- | --- | --- |
| **Venue** | **Type** | **마감 (현행 공지) / 비고** |
| ACM SoCC 2026 | Conference | CFP 미공개  (7월 제출 예상, 11월 컨퍼런스 예상) |
| IEEE ICFEC 2026 | Conference | 2026.1.9 full paper 마감 |
| IEEE TMC / IEEE IoT-J | Journal | 상시 투고 |

**5.2진행 일정표 (IEEE ICFEC)**

|  |  |  |
| --- | --- | --- |
| 구간 | 목표 | 산출물 |
| 2026. 1. 1w | ICFEC 여부 결정 |  |
| 2026. 1. 1~2w | ICFEC 최종 원고, 실험결과 포함 정리 | 초록,본문,부록,스크립트,익명 레포 |
| 2026. 1. 9 | ICFEC 제출 | Full paper |
| 2026. 2. 16 | ICFEC 결과 | Accept/reject 여부 |
| 2026. 3. 15 | ICFEC 최종본 제출 | 최종본 |
| 2026. 4 | ICFEC 발표 준비 | 발표 자료, 데모 |

**5.3진행 일정표 (ACM SoCC)**

|  |  |  |
| --- | --- | --- |
| 구간 | 목표 | 산출물 |
| 2026. 6. | SoCC 2026 대비 실험 스케일업, 시나리오 보강 | SoCC 초록/본문 초안 |
| 2026. 7. (예상) | SoCC 초록/본문 제출 | 초록/본문 |
| 2026. 9 ~ 2026.11 (예상) | SoCC 판정/최종본/발표 |  |