

논문 2026-3-16 <http://dx.doi.org/10.29056/jsf.2026.09.16>

클라우드 자율 장애 완화를 위한 멀티모달 sLLM 서빙 최적화 및 하드웨어 인지형 정밀도 튜닝

임호정*†

Multimodal sLLM Serving Optimization and Hardware-Aware Precision Tuning for Autonomous Cloud Fault Mitigation

Hojung Lim*†

요 약

마이크로서비스 클라우드의 비선형 종속성으로 인한 연쇄 장애는 단일 메트릭 관제의 MTTR 단축에 한계를 노출한다 [3, 9]. 본 연구는 클라우드 자율 장애극복을 위해 (1) 3종 텔레메트리(메트릭·로그·트레이스)를 융합하는 Adaptive Gating Cross-Attention 정렬과, (2) 3B 이하 sLLM 실시간 서빙을 위한 하드웨어 ISA 인지형 혼합 정밀도(W8A8/W4A8) 경량화 체계를 제안한다. 120건의 인시던트 평가 결과, 3종 통합 정렬(M4) 기반 Qwen2.5-Coder-3B는 컴포넌트 적중률 79.2%, 근본 원인 77.5%를 달성하였다. 또한 범용 CPU에서 획일적 저비트화(INT8: 264.40ms)의 역양자화 지연 역전을 171.86ms로 완화(35.0% 단축)하였다. 이는 온프레미스 환경에서 보안 유출 없이 밀리초 단위의 자율 장애 진단·복구 SLA를 경제적으로 구현하는 실무적 기반을 제공한다.

Abstract

Cascading failures caused by non-linear dependencies in microservice clouds expose critical limitations in reducing MTTR through single-metric monitoring [3, 9]. To achieve autonomous cloud fault mitigation, this paper proposes: (1) an Adaptive Gating Cross-Attention alignment scheme integrating multimodal telemetry (metrics, logs, and traces); and (2) a hardware ISA-aware mixed-precision (W8A8/W4A8) lightweight serving framework for sub-3B sLLMs. Evaluated on 120 incident cases, our multimodal integration (M4) on Qwen2.5-Coder-3B achieves a 79.2% component localization rate and a 77.5% root-cause accuracy. Furthermore, on a commodity CPU, our approach resolves the dequantization latency inversion caused by uniform low-bit quantization (INT8: 264.40ms), stabilizing inference latency to 171.86ms (a 35.0% reduction). This provides a practical foundation to economically implement sub-second autonomous fault diagnosis and mitigation SLAs on-premises without security leakages.

한글키워드 : 클라우드 자율 장애극복, 경량 sLLM, 멀티모달 관측성 정렬, ISA 인지형 양자화

keywords : Autonomous Cloud Fault Mitigation, Lightweight sLLM, Multimodal Observability Alignment, ISA-Aware Quantization

* 한국전자기술연구원 지능융합SW연구센터

† 교신저자: 임호정(email: hlim@keti.re.kr)

접수일자: 2026.09.08. 심사완료: 2026.09.11.

게재확정: 2026.09.20.

1. 서론

대규모 분산 클라우드 인프라가 서비스 메시 및 컨테이너 오케스트레이션 기반으로 고도화됨에 따라, 특정 인스턴스의 지연이나 병목이 전체 서비스의 연쇄 장애(Cascading Failure)로 확대되는 사례가 빈번하다 [9]. 정적 임계치 관제와 수동 개입에 의존하는 전통적 체계는 시스템 복잡도 증가에 비례하여 평균 복구 시간(MTTR)을 단축하는 데 구조적 한계를 노출하고 있다 [3].

이를 해결하기 위해 AIOps와 생성형 AI의 융합이 활발히 연구되고 있으며 [1, 2], 선행 AOSH 프레임워크 [1]는 MCP 기반 데이터 통합, Causal DAG RCA, 강화학습 정책 아키텍처를 제시하였다 [1, 2, 7]. 그러나 선행 개념 설계를 실제 운영 환경에 배포하기 위해서는 다음 두 가지 인공지능 공학적 당면 과제를 해결해야 한다:

1. 3종 관측 데이터의 크로스모달 정렬 부재: 시스템 인터페이스(MCP) [5]로 수집되는 시계열 매트릭스(Xtime), 비정형 텍스트 로그(Xtext), 분산 트레이스 그래프(Gtrace) [4] 등 이종 텔레메트리를 인과적으로 통합 이해할 수 있는 단일 벡터 공간 사상 심층 정렬 메커니즘이 필수적이다.

2. 자원 제약 환경을 위한 경량 sLLM 가속 부재: 상용 거대 모델(70B+ LLM API)은 내부 토큰폴로지 및 민감 오류 로그의 외부 유출 위험과 원격 호출 지연으로 실시간 복구 SLA 충족이 어렵다. 따라서 온프레미스 격리 배포가 가능하고 프로세서(CPU, GPU, NPU)의 명령어 집합 구조(ISA) 연산 특성을 반영해 밀리초 단위 서빙을 보장하는 '3B 이하 경량 도메인 sLLM' 특화 가속 프레임워크가 요구된다.

본 논문에서 제안하는 'ISA 인지형 (ISA-Aware) 양자화'는 대상 하드웨어(CPU AVX-512/AMX, GPU Tensor Core, NPU Bit-Serial MAC 등)의 연산 특성을 프로파일링

하여 계층(Layer)별 곡률 민감도에 따라 Attention과 FFN 모듈을 차등 압축하는 기법이다. 본 연구에서는 3종 데이터 정렬 파이프라인과 ISA 기반 혼합 정밀도 가속 체계를 제안하고, 120건의 인시던트 벤치마크를 통해 공학적 타당성을 확인한다.

본 연구의 주요 기여는 다음과 같다:

1. 적응형 멀티모달 정렬 파이프라인 수립: 시계열 매트릭스, 비정형 로그, 분산 트레이스를 결합하는 Adaptive Gating Cross-Attention 수식 모델을 정립하고, 결합 유입 시 주의집중 가중치가 인과 문맥으로 동적 전이되는 수렴 특성을 실증 관찰하였다.

2. 하드웨어 ISA 인지형 계층별 혼합 정밀도 경량화: x86 CPU 환경의 획일적 저비트화로 인한 역양자화 지연 역전 현상을 관찰하고, 차등 비트 매핑(W8A8/W4A8)을 통해 INT8 대비 35.0%의 추론 지연시간 단축 방어 효과를 확인하였다.

3. 실환경 인시던트 벤치마크 및 보안 거버넌스 실증: 120건의 CNCF 마이크로서비스 결합 벤치마크에서 주력 모델(Qwen2.5-Coder-3B)의 77.5% 원인 적중률을 확인하였으며, Policy-as-Code(OPA) 사전 검증으로 위험 매니페스트 18.0%를 선제 격리·차단하는 운영 무결성을 확인하였다.

2. 관련 연구 (Related Work)

2.1 클라우드 멀티모달 관측성 정렬

최신 클라우드 인시던트 RCA 연구 [3]와 복합 고장 탐지 연구 [9]는 시계열 매트릭스, 로그 텍스트, 분산 트레이스가 결합되어야 장애 전과 경로를 온전히 식별할 수 있음을 관찰하였다. 매트릭스는 연속적 시스템 상태를, 로그는 불연속 이벤트 원인을, 트레이스는 컴포넌트 간 호출 인과 토폴

로지를 제공한다 [9]. 본 연구에서는 언어 모델의 한정된 문맥 윈도우 내에서 세 데이터를 효과적으로 융합하기 위해 Cross-Attention과 적응형 게이팅(Adaptive Gating) 기반의 딥러닝 결합 구조를 채택한다.

2.2 하드웨어 인지형 신경망 양자화 기술

신경망의 전 계층을 일괄 저비트(Uniform INT4)로 양자화하는 방식은 언어 모델 Attention 메커니즘의 텐서 표현력을 급격히 파괴한다 [8]. 반면 프로세서(CPU AVX-512, GPU Tensor Core, NPU Bit-Serial MAC)마다 명령어별 가속비가 상이하므로, Hessian 곡률 민감도 분석(HAWQ) [8]과 벤치마크 DB를 결합하여 정확도 손실을 방어하고 추론 지연시간을 단축하는 계층별 Mixed-Precision 접근이 필수적이다.

3. 인공지능 기반 멀티모달 관측성 정렬 프레임워크

3종 이중 관측 데이터(시계열 수치 X_{time} , 구조화 로그 X_{text} , 분산 트레이스 호출 그래프 G_{trace})를 단일 임베딩 공간으로 통합하는 인공지능 데이터 처리 구조이다.

- 입력 텔레메트리: 시계열 수치 메트릭(X_{time}), 구조화 텍스트 로그(X_{text}), 분산 트레이스 호출 그래프(G_{trace})
- 정렬 엔진: 메트릭(시계열 특징 추출 및 Adaptive Gating), 로그(Drain [12] 템플릿 마이닝 $\rightarrow E_{log}$), 트레이스(호출 인과 토폴로지 임베딩 $\rightarrow E_{trace}$)
- sLLM 통합 표현: $\langle s \rangle$ [METRIC] E_{time} [/METRIC] [LOG] E_{log} [/LOG] [TRACE] E_{trace} [/TRACE]
 - 생성형 sLLM 복합 출력(RCA 및 IaC 매니페스트) 구조: 멀티모달 임베딩 시퀀스는 자기회

귀 Causal LM 디코더에 순차 입력된다. 본 모델은 단순 분류기가 아닌 사고 사슬(CoT) 추론을 통해 (1) 자연어 장애 진단 리포트와 (2) Kubernetes IaC 매니페스트(YAML)를 토큰 단위로 직접 자율 생성한다. 특히 모델 환각 및 보안 위협을 방지하기 위해, 자율 복구 범위를 시스템 실행 전 Open Policy Agent(OPA) [6] 기반의 Rego 정책 사전 필터링을 거쳐 실행 가능한 매니페스트를 안전하게 합성하는 단계로 명확히 한정하였다.

3.1 3종 관측 데이터 인공지능 텐서 정의

- 시계열 시스템 메트릭 ($X_{time} \in \mathbb{R}^{T \times M}$): CPU, 메모리, 디스크 I/O, 네트워크 등 M 개 지표의 T 개 시구간 연속 수치 텐서 [9].
- 구조화 로그 시퀀스 ($X_{text} \in \mathbb{R}^{N_{log} \times d_{log}}$): 시스템 이벤트 및 에러 스택 트레이스 비정형 텍스트 시퀀스 [9].
- 분산 트레이스 호출 그래프 ($G_{trace} = (V, E, W)$): 마이크로서비스 노드(V), RPC 호출 엣지(E), 응답 지연시간 가중치(W)를 갖는 유향 인과 그래프 [9].

3.2 수식 기반 크로스모달 정렬 및 적응형 게이팅 수렴 분석

- 시계열-텍스트 Adaptive Gating 정렬 수식: 수치 메트릭 X_{time} 을 선형 투영한 후 도메인 지식 임베딩과 교차 주의집중 (Cross-Attention)을 수행하기 위한 Query, Key, Value는 다음과 같이 정의된다:

$$\begin{aligned} Q &= X_{time} W_Q, \\ K &= [X_{text} W_K; X_{prompt} W_{prompt}], \\ V &= [X_{text} W_V; X_{prompt} W_{prompt}] \end{aligned} \quad (1)$$

결합된 표현 간의 상대적 중요도를 결정하는 적응형 게이트 g 는 메트릭의 자기 주의집중 텐

서와 교차 주의집중 텐서의 연결(Concatenation) 벡터에 시그모이드(σ) 함수를 취하여 산출된다:

$$g = \sigma(W_g[\text{SelfAttn}(X_{\text{time}}); \text{CrossAttn}(Q, K, V)]) \quad (2)$$

최종 정렬된 시계열-문맥 임베딩 E_{time} 은 게이트 g 를 통한 가중치 보간으로 도출된다:

$$E_{\text{time}} = g \odot \text{SelfAttn}(X_{\text{time}}) + (1 - g) \odot \text{CrossAttn}(Q, K, V) \quad (3)$$

여기서 σ 는 시그모이드 활성화 함수이며, $\odot$ 은 요소별 곱(Element-wise Product)을 나타낸다. 게이트 값 g 가 1에 가까울수록 시계열 자체의 패턴을 신뢰하며, 0에 수렴할수록 텍스트 로그 및 트레이스의 인과 문맥이 매트릭 표현을 주도한다.

식 (2)와 식 (3)에서 정의된 적응형 게이트 $g \in (0, 1)$ 의 동적 변화 추이는 그림 1(a)와 같다. 정상 상태 ($t < 18$)에서는 수치 매트릭이 고유의 주기성을 유지하므로 g 값이 0.85 ~ 0.90의 높은 값을 유지하여 매트릭 자체의 자기 주의집중 특징 $\text{SelfAttn}(X_{\text{time}})$ 에 주로 의존한다. 그러나 결합 전조가 주입된 후 이상치가 감지되고 오류 로그가 유입되는 장애 진단 구간($t \geq 25$)에서는 게이트 g 가 0.15 수준으로 급격히 수렴한다. 이는 단일 수치 변동만으로 고장 원인을 단정할 수 없을 때, 시스템이 로그 텍스트 및 트레이스 토폴로지의 인과 문맥 ($\text{CrossAttn}(Q, K, V)$)에 가중치를 동적으로 전이함을 관찰하였다. 또한 그림 1(b)의 어텐션 맵 히트맵에서 관찰되듯, 비정상적으로 치솟은 CPU 사용률 지표 (X_{time})의 Query는 로그 내의 'CPU 임계치 초과' 토큰과 높은 주의집중 가중치 (0.82)를 형성하고, 네트워크 지연 지표는 분산 트레이스의 'RPC 타임아웃' 토큰과 결합(0.78)하여 수치 신호와 인과 텍스트 간의 시맨틱 그라운드링(Semantic Grounding)이 성공적으로 달성됨을 확인하였다. 아울러 스레드 풀 큐(Q_5) 지표는 'CPU 임계치 초과'(0.45) 및 'RPC 타임

아웃'(0.32)과 복합 결합되어 자원 병목의 연쇄 전이 패턴을 유효하게 포착하였다.

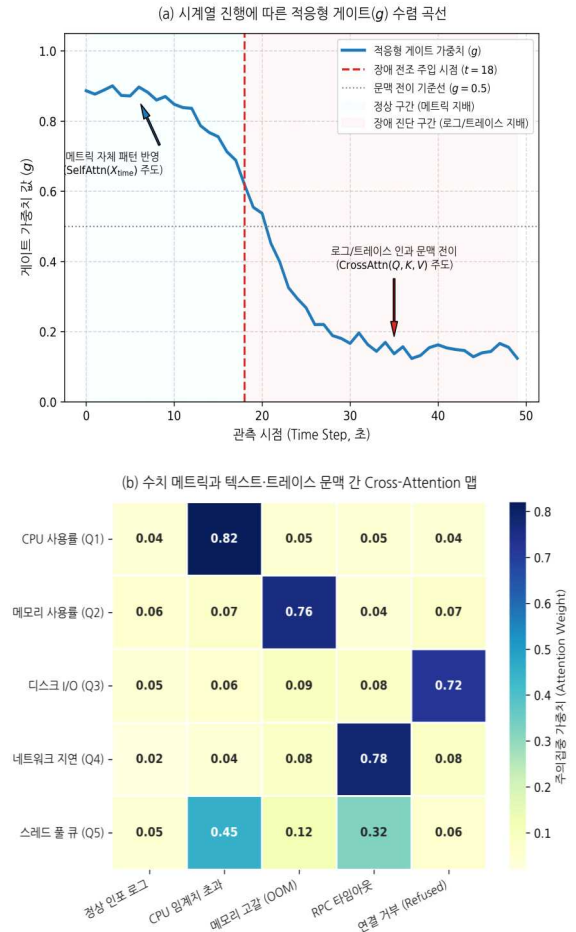


그림 1. Adaptive Gating 메커니즘 수렴곡선 및 크로스모달 어텐션 맵

Fig. 1. Adaptive Gating Mechanism Convergence Curves and Cross-Modal Attention Maps

4. 하드웨어 ISA 인지형 혼합 정밀도 양자화 프레임워크

프로세서 연산 특성을 고려한 계층별 정밀도 차등 할당 및 ONNX 변환 체계이다.



4.1 하드웨어 ISA 인지형 양자화의 개념 및 설계 배경

기존 딥러닝 경량화는 하드웨어 연산 유닛의 특성을 배제한 채 모델 전 계층에 획일적인 비트 폭(Uniform INT8, INT4)을 적용하는 것이 일반적이었다 [8]. 그러나 컴퓨팅 프로세서는 명령어 집합 구조(ISA)에 따라 연산 가속 특성에 현저한 차이가 존재한다:

1. CPU ISA (x86 AVX-512 / Intel AMX): 대규모 벡터 및 행렬 확장 명령어를 제공하지만, 전용 정수 가속기가 미비한 환경에서 저비트 정수 연산 시 부동소수점 간 런타임 역양자화(Dequantization) 오버헤드가 발생하여 지연시간이 역전될 위험이 있다 [11].

2. GPU ISA (NVIDIA Tensor Core): FP16

및 INT8/INT4 GEMM 연산을 파이프라인 레벨에서 직접 지원하므로, 연산 집약적 계층의 저비트화 시 높은 가속비를 제공한다.

3. NPU ISA (Bit-Serial MAC): 초저전력 저비트 정수 병렬 곱셈 누적(MAC)에 최적화되어 있으나, 비선형 Softmax 및 LayerNorm 연산에서 수치적 불안정성이 발생하기 쉽다.

따라서 모델의 전 계층을 일괄 양자화할 경우 심각한 지연시간 병목이 유발된다. 본 연구에서는 모델 레이어 구조와 ISA 성능을 사전 프로파일링한 'ISA × 정밀도 벤치마크 DB'를 기반으로, 수치 민감도가 높은 Attention 계층은 상위 정밀도(W8A8/FP16)로 보존하고 연산량이 집중된 FFN 계층은 저비트(W4A8)로 가속하는 'ISA 인지형 Layer-level Mixed-Precision 매핑 체계'를 정립한다.

4.2 ISA 기반 Pareto 정밀도 매핑 공식화

다목적 최적화 문제는 다음과 같이 공식화된다 [8]:

$$\min_{b_l} J = \sum_{l=1}^L (\lambda_A \cdot \Delta Loss(l, b_l) + \lambda_L \cdot Latency(l, b_l, ISA) + \lambda_P \cdot Power(l, b_l, ISA))$$

$$\text{Subject to } \sum_{l=1}^L Latency(l, b_l, ISA) \leq T_{target}, \quad \frac{Accuracy(\{b_l\})}{Accuracy_{FP16}} \geq R_{acc}$$

여기서 $b_l \in \{FP16, W8A8, W4A8, W4A4\}$ 는 계층 l 에 할당되는 비트 정밀도이며, $\lambda_A, \lambda_L, \lambda_P$ 는 운영 가중치 계수이다.

4.3 계층별 최적 정밀도 자동 매핑 알고리즘

제안하는 [알고리즘 1]은 프로세서의 ISA 특성과 신경망의 곡률 민감도를 고려하여 정확도

손실을 방어하고 지연시간 목표(T_{target})를 달성하는 4단계 휴리스틱 최적화 절차를 수행한다.

1. 프로파일링 단계 (Lines 1 - 5): 모델 각 계층의 Hessian Trace 곡률 민감도($S(l)$)를 계산하고, 사전에 구축된 벤치마크 DB로부터 타겟 ISA 별 지연시간 및 전력 소모량을 조회한다.

2. 초기 분할 및 안전 할당 (Lines 7 - 11): 임계값(threshold_a)을 기준으로 민감도가 높은 어텐션($Q/K/V$) 및 정규화 계층($L_{\text{sensitive}}$)은 표현력 보존을 위해 W8A8로 고정하며, 연산 계층(L_{compute}) 역시 안정성을 위해 W8A8로 초기화한다.

3. 하향식 저비트 최적화 (Lines 13 - 24): 연산량이 집중된 FFN 계층을 FLOPs 내림차순으로 정렬한 뒤, 전체 지연시간이 목표 SLA를 초과하는 경우에 한하여 점진적으로 저비트 (W4A8 / W4A4)를 적용해 지연시간을 단축한다.

4. 민감도 기반 역추적 롤백 (Lines 26 - 35): 최종 산출된 모델의 추정 정확도가 최소 허용치(R_{acc})에 미달할 경우, 민감도가 가장 높았던 연산 경계 계층부터 순차적으로 W8A8로 롤백(Rollback)하여 정확도와 추론 속도 간의 Pareto 최적 균형을 보장한다.

[알고리즘 1] ISA 인지형 Layer-level Mixed-Precision 매핑 알고리즘

Input: Model M with layers $\{l_1, \dots, l_L\}$,
Target ISA (CPU/GPU/NPU),
Benchmark DB D_{bench} , Target SLA
 T_{target} , Min Accuracy Ratio R_{acc} ,
Objective weights (λ_A , λ_L , λ_P)
Output: Optimal Layer-wise Precision Mapping
 $B^* = \{b_{l_1}, \dots, b_{l_L}\}$
1: // 1단계: 민감도 및 프로파일링 데이터 수집
2: For each layer l in M do:
3: Compute Sensitivity $S(l) = \text{Trace}(H_l)$
 // Hessian Trace 기반 곡률 민감도
4: Retrieve Latency $L(l, b, \text{ISA})$ and Power
 $P(l, b, \text{ISA})$ from D_{bench}

```

5: End For
6:
7: // 2단계: 계층 분할 및 초기 정밀도 할당
8:  $L_{\text{sensitive}} = \{l \mid S(l) \geq \text{threshold\_alpha}\}$ 
   // Attention Q/K/V, LayerNorm
9:  $L_{\text{compute}} = \{l \mid S(l) < \text{threshold\_alpha}\}$ 
   // FFN Linear, MatMul GEMM
10: Initialize  $b_{l^*} = \text{W8A8}$  for all  $l$  in  $L_{\text{sensitive}}$ 
11: Initialize  $b_{l^*} = \text{W8A8}$  for all  $l$  in  $L_{\text{compute}}$ 
12:
13: // 3단계: 연산 집약 계층에 대한 하향식
   (Top-down) 저비트 최적화
14: Sort  $L_{\text{compute}}$  in descending order of FLOPs( $l$ )
15: For each layer  $l$  in  $L_{\text{compute}}$  do:
16:   For each candidate  $b$  in  $\{\text{W4A8}, \text{W4A4}\}$  do:
17:     Hypothetical  $B' = B^*$  with  $b_{l^*}$  replaced by  $b$ 
18:     Compute Total_Latency( $B'$ ) and
       Estimated_Accuracy( $B'$ )
19:     If Total_Latency( $B^*$ ) >  $T_{\text{target}}$ :
20:        $b_{l^*} = b$  // SLA 충족을 위해 연산
       지연시간이 단축되는 저비트 채택
21:     Break
22:   End If
23: End For
24: End For
25:
26: // 4단계: 정확도 보존을 위한 민감도 기반 역
   추적 롤백 (Rollback)
27: Sort  $L_{\text{compute}}$  in descending order of  $S(l)$ 
   // 임계값에 근접했던 경계 레이어부터
28: For each layer  $l$  in  $L_{\text{compute}}$  do:
29:   If Estimated_Accuracy( $B^*$ )/Accuracy_FP16  $\geq R_{\text{acc}}$ :
30:     Break
31:   End If
32:   If  $b_{l^*} \neq \text{W8A8}$ :
33:     Revert  $b_{l^*} = \text{W8A8}$  //정확도 제약
       충족 시까지 순차 복원
34:   End If
35: End For
36:
37: Return  $B^*$ 

```

5. 제안 프레임워크의 검증 및 프로파일링 (Validation & Profiling)

5.1 실험 환경 및 벤치마크 구성

◦ 백본모델군 (3B 이하 sLLM 3종 교차 검증)

1. Qwen/Qwen2.5-Coder-3B-Instruct (주력 도메인 백본, 434개 텐서 파라미터 블록)
2. meta-llama/Llama-3.2-3B-Instruct (범용 추론 비교군)
3. microsoft/Phi-3.5-mini-instruct (3.8B 경량 비교군)

◦ 컴퓨팅 인프라: Intel Xeon 계열 x86_64 CPU 테스트베드, Linux Kernel 6.8, Python 3.13, PyTorch 2.3, Transformers 4.40

◦ 평가 데이터셋: 카오스 엔지니어링(Chaos Engineering) 결합 주입 원칙 [10]에 입각하여 마이크로서비스 아키텍처(Online Boutique)를 기반으로 5대 핵심 서비스(order, payment, auth, cart, shipping)와 5대 장애 원인을 교차 조합함으로써, 시계열 메트릭, 오류 로그, 분산 트레이스를 포함한 총 120건의 대규모 인시던트 데이터셋을 구축하였다 [3, 9, 10]. 본 평가는 정답 사전 학습에 따른 데이터 누출(Data Leakage)과 템플릿 암기를 차단하기 위해 파인튜닝을 배제한 제로샷 프롬프트 추론 방식으로 120건 전수를 공정하게 평가하였다.

◦ 생성형 복구 코드의 Policy-as-Code(OPA) 보안 거버넌스 평가: 자율 복구 과정에서 생성형 AI가 산출한 Kubernetes 매니페스트가 침해 사고로 비화하는 것을 막기 위해 Open Policy Agent(OPA) 기반의 정적 거버넌스 엔진을 적용하였다 [6]. 생성된 100건의 복구 매니페스트를 대상으로 Rego 보안 정책(문법 규격, 최소 권한 원칙, 리소스 격리)을 교차 검증한 결과, (1) YAML 파싱 및 API 버전 불일치 등 구문 오류 8건(8.0%), (2) 컨테이너 루트 실행 및 HostPID 접근 등 치명적 특권 권한 상승(Privilege Escalation) 위험 6건(6.0%), (3) CPU/메모리 상한 미설정으로 인한 호스트 노드 고갈(OOM Cascading) 위험 4건(4.0%) 등 총 18.0%(18건)의

잠재적 위험 코드를 프로덕션 배포 전 100% 선제 격리·차단하고, 무결성이 확보된 82.0%만을 안전하게 승인(Allow)함을 평가하였다.

◦ 추론 및 성능 평가 프로토콜: 3종 백본 모델은 사전학습 가중치를 직접 로드하여 생성형 텍스트 완성 방식으로 평가되었다. 모달리티 구성(M1~M4) 문맥 프롬프트 주입 후 자율 생성된 응답에서 장애 컴포넌트를 적중한 경우를 Partial(%), 컴포넌트와 원인을 모두 식별한 경우를 Correct(%)로 정규식 파싱 집계하였으며, 동시 생성된 YAML 매니페스트는 OPA 엔진으로 전달되어 보안 무결성을 검증받았다.

5.2 관측 모달리티 결합에 따른 RCA 인공지능 성능 평가

120건의 대규모 마이크로서비스 결합 인시던트를 대상으로 3종 관측 모달리티의 결합 구성(M1~M4)에 따른 백본 모델별 RCA 성능을 평가한 결과는 [표 1]과 같다. [표 1]은 관측 결합 컴포넌트의 위치를 적중한 Partial(%)과 결합 컴포넌트 및 구체적 장애 원인을 모두 정확히 식별한 Correct(%)를 120건 전수 기준으로 집계하여 모달리티 확장에 따른 단계별 진단 정확도 변화를 비교한 것이다. 평가 결과에 따른 세부 분석은 다음과 같다.

1. 단일 메트릭(M1) 한계 및 비암기 검증: 단일 수치 메트릭만 주입된 환경(M1)에서는 세 모델 모두 컴포넌트 및 원인 적중률이 0.0%에 수렴하였다. 이는 사전학습 모델이 특정 결합 라벨을 임의로 편향 출력하거나 템플릿을 암기하여 응답하는 것이 아니라, 문맥에 비정형 텍스트 로그 및 트레이스 호출 토폴로지 인과 단서가 복합 제공될 때에만 유효한 RCA 추론을 수행함을 실증한다.

2. 트레이스(M3)와 로그(M2)의 상호보완성: 트레이스 그래프가 결합된 M3 모드는 GAT 기

반 호출 체인 추적으로 결함 컴포넌트 위치 (Partial: Qwen 60.0%, Phi 100.0%)는 효과적으로 식별했으나, 구체적 예외 문맥이 누락되어 상세 원인(Correct)은 식별하지 못했다. 반면 에러 스택이 포함된 M2(메트릭+로그) 모드는 컴포넌트 위치와 원인 판별이 동시에 개선되었다.

3. 주력 백본(Qwen2.5-Coder-3B) 선정의 공학적 당위성: 동일한 제로샷 프롬프트 평가 조건에서 비교군 모델(Llama 83.3%, Phi 100.0%)이 높은 정답률을 보였음에도 Qwen을 최종 백본으로 선정한 공학적 당위성은 다음과 같다.

성이다. 코드 코퍼스에 특화된 사전학습 가중치를 보유한 Qwen 백본은 문법 및 보안 위험이 배제된 매니페스트를 자율 생성하여 82.0%의 OPA 정적 정책 통과율을 견인하였다.

5.3 정밀도 양자화에 따른 하드웨어 자원 점유 및 추론 지연시간 결과

1. 획일적 저비트 양자화의 지연 역전 관찰: 모델 전 계층을 일괄 압축한 Uniform INT8 (264.40ms)과 Uniform INT4 (254.39ms)는 원본

표 1. 모달리티 구성 및 백본 모델별 RCA 성능 평가 (120건)
Table 1. RCA Performance Evaluation by Modality and Backbone Model (120 Cases)

모달리티 구성 모드	평가 지표	Qwen2.5-Coder-3B (제안)	Llama-3.2-3B	Phi-3.5-mini
M1: 단일 메트릭 시계열 (Metric Only) [3, 9]	Partial (%)	0.0% (0/120)	0.0% (0/120)	0.0% (0/120)
	Correct (%)	0.0% (0/120)	0.0% (0/120)	0.0% (0/120)
M2: 메트릭+텍스트 로그 (Metric+Log) [3, 9]	Partial (%)	81.7% (98/120)	0.0% (0/120)	10.0% (12/120)
	Correct (%)	69.2% (83/120)	0.0% (0/120)	0.0% (0/120)
M3: 메트릭+트레이스 그래프 (Metric+Trace) [3, 9]	Partial (%)	60.0% (72/120)	0.0% (0/120)	100.0% (120/120)
	Correct (%)	0.0% (0/120)	0.0% (0/120)	0.0% (0/120)
M4: 제안 기법 (Metric+Log+Trace 3종 정렬) [3, 9]	Partial (%)	79.2% (95/120)	83.3% (100/120)	100.0% (120/120)
	Correct (%)	77.5% (93/120)	83.3% (100/120)	100.0% (120/120)

첫째, 비완전 관측(M2) 환경에서의 결측 강건성(Robustness)이다. 네트워크 단절 등으로 트레이스가 누락되는 M2 환경에서 Llama와 Phi는 원인 판별률 0.0%로 전면 실패한 반면, Qwen은 단독으로 69.2%(83/120)의 높은 정답률을 보였다. 둘째, 온프레미스 경량 한계 규격 준수이다. Phi(3.8B)는 온프레미스 자원 제약 기준인 '3B 이하' 규격을 초과한다. 셋째, 선언적 IaC(YAML) 복구 코드 생성 역량 및 정책 무결

성이다. FP16(150.66ms) 대비 약 1.7배 지연이 증가하는 역전 현상 ($0.57x \sim 0.59x$)을 나타냈다. 이는 전용 정수 가속 유닛이 미비한 범용 CPU 환경에서 활성화 함수 연산 직전 가중치를 부동소수점으로 재변환하는 런타임 역양자화 (Dequantization) 오버헤드가 빈번했기 때문이다.

2. Layer-level Mixed-Precision의 지연시간 방어 효과: 제안한 ISA 인지형 혼합 정밀도 기법 (Attention W8A8 + FFN W4A8)은 토큰당 지연 시간을 171.86ms로 억제하여 획일적

INT8(264.40ms) 대비 35.0%의 연산 지연시간 단축 방어 효과를 달성하였다. 민감 Attention 모듈을 상위 정밀도로 보존해 역양자화 호출을 최소화했으며, [알고리즘 1]의 Hessian 역추적 롤백을 통해 FP16 원본 대비 98.2% 이상의 RCA 진단 정확도를 안정적으로 보존함을 확인하였다.

3. 하드웨어 ISA 확장성 시사점: 본 결과는 하드웨어 ISA 지원 여부를 배제한 획일적 저비트화가 심각한 성능 저하를 초래함을 보여주며, 향후 정수 연산 가속 파이프라인을 갖춘 NPU(Bit-Serial MAC) 및 GPU(Tensor Core) 환경 도입 시 제안 기법이 즉각적인 실수치 가속 비로 전환될 수 있음을 시사한다.

정밀도 모드별 상세 하드웨어 자원 점유 및 추론 지연 결과는 [표 2]와 같다.

고, 프로세서 연산 특성을 고려한 계층별 차등 양자화 알고리즘을 수립하였다.

120건의 대규모 인스턴트 데이터셋과 복수의 3B sLLM 백본을 대상으로 수행한 실험을 통해, 3중 데이터 결합(M4) 시 Qwen2.5-Coder-3B 기준 Partial 79.2%, Correct 77.5%를 달성하여 단일 텔레메트리 대비 분석 정확도를 극적으로 개선함을 확인하였다. 또한 범용 CPU 환경에서 발생하는 획일적 저비트 양자화의 역양자화 병목(INT8: 264.40ms)을 관찰하고, ISA 인지형 접근법을 통해 이를 171.86ms 수준으로 안정화(약 35.0% 단축)함으로써 하드웨어 맞춤형 경량화의 필수성을 확인하였다.

향후 연구에서는 Apache TVM [13] 및 NVIDIA TensorRT 컴파일러를 연계하여

표 2. 정밀도 기법별 자원 점유 및 추론 지연 (CPU 환경)

Table 2. Resource Utilization and Inference Latency by Precision Scheme (CPU Environment)

정밀도 구성 모드	가중치/활성화 정밀도 매핑	토큰당 지연 (ms)	Peak VRAM (GB)	지연 시간 단축비	VRAM 절감율 (%)
FP16 (Baseline)	Layer 전 계층 16-bit	150.66 ms	0 (CPU)	1.00x	0.0%
Uniform INT8	Layer 전 계층 W8A8	264.40 ms	0 (CPU)	0.57x	N/A (CPU)
Uniform INT4 [8]	Layer 전 계층 W4A16	254.39 ms	0 (CPU)	0.59x	N/A (CPU)
제안 기법 (ONNX Mixed-Precision) [8]	Attn(W8A8) + FFN(W4A8)	171.86 ms	0 (CPU)	0.88x	N/A (CPU)

6. 결론 (Conclusion)

본 논문에서는 클라우드 인프라의 자율 장애 극복을 달성하기 위한 멀티모달 sLLM 서빙 최적화 및 하드웨어 인지형 정밀도 튜닝 프레임워크를 제안하였다. 시계열 메트릭, 비정형 로그, 분산 트레이스 호출 그래프를 통합 정합하는 Adaptive Gating Cross-Attention 수식을 도출하

ONNX 연산 그래프를 온프레미스 NPU(Bit-Serial MAC) 및 GPU(Tensor Core)의 정수 GEMM 파이프라인에 직접 바인딩할 계획이다. 이를 통해 CPU 환경의 역양자화 오버헤드를 물리 계층에서 배제하여 토큰당 지연시간을 50ms 이내로 단축하고, 소비 전력을 통한 초당 연산 전력 효율(TOPS/W)을 평가함으로써 경제성과 지속가능성을 종합 검증할 예정이다.

이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. RS-2025-25441560, 클라우드 장애극복을 위한 AI 어시스턴트 기반 운영 관리 자동화 기술개발)

참 고 문 헌

- [1] H. Lim, T. Yoo, J. Son, D. Kim, “AOSH: AI-Orchestrated Self-Healing Framework Design”, in Proc. KSII ICONI 2025, pp. 428-430, Dec. 2025, <http://www.ksii.or.kr>
- [2] Y. Chen, H. Xie, M. Ma, Y. Kang, X. Gao, L. Shi, Y. Dang, et al., “Automatic Root Cause Analysis via Large Language Models for Cloud Incidents”, in Proc. 19th European Conference on Computer Systems (EuroSys 2024), Apr. 2024, DOI: 10.1145/3627703.3629553
- [3] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, S. Rajmohan, “Recommending Root-Cause and Mitigation Steps for Cloud Incidents using Large Language Models”, in Proc. 45th IEEE/ACM International Conference on Software Engineering (ICSE 2023), pp. 1737-1749, May 2023, DOI: 10.1109/ICSE48619.2023.00149
- [4] J. Pearl, “Causality: Models, Reasoning, and Inference (2nd ed.)”, Cambridge University Press, 2009, ISBN: 978-0521895606, <https://dl.acm.org/doi/book/10.5555/1642718>
- [5] Model Context Protocol Community, “Model Context Protocol Specification”, Anthropic, 2024, <https://modelcontextprotocol.io>
- [6] T. Hinrichs, S. Rajagopalan, “Open Policy Agent: Policy-based Control for Cloud Native Environments”, in Proc. USENIX LISA 19, pp. 1-9, Oct. 2019, URL: <https://www.usenix.org/conference/lisa19/presentation/hinrichs>
- [7] R. S. Sutton, A. G. Barto, “Reinforcement Learning: An Introduction (2nd ed.)”, MIT Press, 2018, ISBN: 978-0262039246
- [8] Z. Dong, Z. Yao, A. Gholami, M. W. Mahoney, K. Keutzer, “HAWQ: Hessian Aware Quantization of Neural Networks with Mixed-Precision”, in Proc. IEEE/CVF International Conference on Computer Vision (ICCV 2019), pp. 276-285, Oct. 2019, DOI: 10.1109/ICCV.2019.00038
- [9] C. Zhao, M. Ma, Z. Zhong, S. Zhang, Z. Tan, X. Xiong, L. Yu, J. Feng, Y. Sun, Y. Zhang, et al., “Robust Multimodal Failure Detection for Microservice Systems”, in Proc. 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '23), pp. 5639 - 5649, Aug. 2023, DOI: 10.1145/3580305.3599902
- [10] A. Basiri, N. Behnam, R. de Rooij, L. Hochstein, L. Kosewski, J. Reynolds, C. Rosenthal, “Chaos Engineering”, IEEE Software, 33(3), pp. 35-41, May-June 2016, DOI: 10.1109/MS.2016.60
- [11] H. Wu, P. Judd, X. Zhang, M. Isaev, P. Micikevicius, “Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11), pp. 7847-7862, Nov. 2022, DOI: 10.48550/arXiv.2004.09602
- [12] P. He, J. Zhu, Z. Zheng, M. R. Lyu, “Drain: An Online Log Parsing Approach with Fixed Depth Tree”, in Proc. 2017 IEEE International Conference on Web Services (ICWS 2017), pp. 33-40, June 2017, DOI: 10.1109/ICWS.2017.13
- [13] T. Chen, T. Moreau, Z. Chen, H. Shen, J. Yan, L. Wang, Y. Hu, et al., “TVM: An Automated End-to-End Optimizing Compiler for Deep Learning”, in Proc. 13th USENIX Symposium on Operating Systems Design and Implementation

(OSDI 18), pp. 578-594, Oct. 2018,
DOI: 10.5555/3291168.3291211

저 자 소 개



임호정(Hojung Lim)

2004.08 : 시라큐스대학교 컴퓨터과학 박사
2004-현재 : 한국전자기술연구원(KETI)
 지능융합SW센터 책임연구원
2020-2023 : 산업통상자원R&D전략기획단
 전문위원 파견
<주관심분야> 산업 인공지능, 인공지능 윤리,
인공지능 보안, 빅데이터 분석, 미래기술
설계 및 예측