

A Monitoring Scheme Based on Artificial Intelligence in Mobile Edge Cloud Computing Environments

JongBeom Lim[†] · HeeSeok Choi^{††} · HeonChang Yu^{†††}

ABSTRACT

One of the crucial issues in mobile edge cloud computing environments is to monitor mobile devices. Due to the inherit properties of mobile devices, they are prone to unstable behavior that leads to failures. In order to satisfy the service level agreement (SLA), the mobile edge cloud administrators should take appropriate measures through a monitoring scheme. In this paper, we propose a monitoring scheme of mobile devices based on artificial intelligence in mobile edge cloud computing environments. The proposed monitoring scheme is able to measure faults of mobile devices based on previous and current monitoring information. To this end, we adapt the hidden markov chain model, one of the artificial intelligence technologies, to monitor mobile devices. We validate our monitoring scheme based on the hidden markov chain model. The proposed monitoring scheme can also be used in general cloud computing environments to monitor virtual machines.

Keywords : Mobile Edge Cloud, Monitoring, Artificial Intelligence, Fault Tolerance

모바일 엣지 클라우드 환경에서 인공지능 기반 모니터링 기법

임종범[†] · 최희석^{††} · 유현창^{†††}

요약

모바일 엣지 클라우드 환경에서 중요하게 다루어야 할 사항 중 하나는 모바일 장치에 대한 모니터링이다. 모바일 장치는 장치의 특성상 불안정한 상태가 발생하여 결함이 발생할 수 있기 때문에 모바일 엣지 클라우드의 SLA (Service Level Agreement)를 만족시키기 위해서는 모바일 장치의 모니터링 기법을 통해 결함을 측정하여 이에 대한 조치를 수행하여야 한다. 이 논문에서는 모바일 엣지 클라우드 환경에서 인공지능 기반 모바일 장치 모니터링 기법을 제안한다. 제안하는 모니터링 기법은 모바일 장치에 대한 이전 모니터링 정보와 현재 모니터링 정보를 기반으로 모바일 장치의 결함 발생을 측정할 수 있도록 설계 되었다. 이를 위해 인공지능 기법 중 하나인 은닉 마르코프 체인 모델을 모바일 장치에 대한 모니터링 기법에 적용하였다. 실험 평가를 통해 제안하는 모니터링 기법에 대한 검증을 수행하였다. 제안하는 기법은 모바일 장치뿐만 아니라 일반적인 클라우드 환경에서의 가상 머신을 모니터링 하는 방법으로도 활용할 수 있도록 설계되었다.

키워드 : 모바일 엣지 클라우드, 모니터링, 인공지능, 결함 포용

1. 서론

최근 IT 산업의 주요 트렌드는 인공지능, 클라우드 컴퓨팅, IoT 등으로, 특히 클라우드의 성장과 IoT가 밀접하게 연관되어 헬스케어와 자율주행차 시장을 주도할 것으로 예상

된다[1, 2]. 인텔은 AutoMobility LA 2017에서 자율주행차를 포함하여 자동차가 8시간 마다 40TB를 생산 또는 소비하고 하루에 한 자동차가 4TB의 데이터를 생산해 낼 것이라고 발표하여 클라우드 컴퓨팅의 부하를 가중 시킬 것으로 예상하고 있다[3, 4].

MEC(Mobile Edge Cloud) 서비스는 모바일 디바이스와 주변 기지국 간의 근접성을 이용하여 작업의 일부분을 오프로드하여 모바일 디바이스의 활용성을 높이는 기술로, 헬스케어와 자율주행 차와 같은 실시간 서비스에 활용될 수 있어 각광 받고 있는 기술이다[5-8].

토요타, 인텔, 에릭슨 등 글로벌 기업들은 엣지 컴퓨팅 컨소시엄을 출범했으며, MS 및 아마존, HP는 IoT를 위한 엣지 클라우드 서버 및 소프트웨어를 공개하여 MEC 환경 구축에 박차를 가하고 있는 실정이다[9, 10].

* 이 논문은 정부(과학기술정보통신부 및 교육부)의 재원으로 한국연구재단(차세대정보컴퓨팅기술개발사업 및 이공학개인지초연구지원사업)의 지원을 받아 수행된 연구임(No. 2017M3C4A7081955, NRF-2015R1D1A1A01061373).

† 중신회원 : 한국산업기술대학교 게임공학부 조교수

†† 준회원 : 고려대학교 컴퓨터학과 박사수료

††† 중신회원 : 고려대학교 컴퓨터학과 교수

Manuscript Received : December 11, 2017

First Revision : January 9, 2018

Accepted : January 28, 2018

* Corresponding Author : HeonChang Yu(yu@korea.ac.kr)

시스템의 고가용성은 99% 이상의 시스템 가용성을 의미하고 MEC 서비스는 실시간성 요구가 높아 고가용성을 보장해야 함에 따라 모바일 서비스를 위한 무정지 지향 엣지 클라우드 시스템 개발이 필요하다[11, 12].

MEC 환경에서 고가용성을 보장하지 않아 실시간성이 저하되는 경우에는 SLA를 위반하여 결함으로 판단되기 때문에 이에 대한 모니터링 기법을 포함한 측정 및 회복 기법에 대한 적용이 필요하다[13, 14].

이 논문에서는 모바일 엣지 클라우드 환경에서 결함 예측을 위한 인공지능 기반 모바일 장치 모니터링 기법을 제안한다. 제안하는 모니터링 기법은 모니터링 히스토리 정보를 기반으로 모바일 장치의 결함 발생을 예측할 수 있도록 설계되었다. 이를 위해 인공지능 기법 중 하나인 HMM(Hidden Markov Chain)을 모바일 장치에 대한 모니터링 기법에 적용하였다. HMM은 인공지능 분야 중 음성 인식, 필기 인식, 동작 인식, 품사 태깅, 악보에서 연주되는 부분을 찾는 작업, 부분 방진, 생물정보학에서 사용되는 기법 중 하나이다. 제안하는 모니터링 기법을 사용하여 MEC 관리자는 모바일 장치에 대한 결함을 측정하여 데이터 백업, 복제, 체크포인트 등 결함에 대한 조치를 취할 수 있다.

2. 관련 연구

일반적으로 시스템에서 발생하는 결함의 유형에는 크게 fail-stop [15, 16], omission [17, 18], Byzantine [19, 20] 결함 등으로 나뉜다. 또한 MEC에서는 모바일 장치의 이동성으로 인해 발생하는 성능 저하도 결함의 유형으로 분류하고 있다[21].

자율주행차와 헬스케어와 같은 서비스의 경우 모바일 장치 또는 노드의 결함이 발생한 경우 실시간성을 보장하지 못하여 서비스 제공에 영향이 미친다. 만약 이로 인해 서비스를 신뢰하지 못하는 상황이 발생한다면 SLA 위반으로 이어질 수 있다.

MEC의 특징을 고려한 위의 네 가지 결함에 대한 결함 포용 방법과 결함 감지, 예측 및 회피 방법에 대하여 ABFT (algorithm-based fault tolerance), 메시지 로깅(message logging), 체크포인트/재시작(checkpoint/restart), 중복(redundancy) 등이 연구되고 있는 실정이다.

하지만 MEC 환경의 모바일 디바이스에 대한 최소한의 오버헤드를 갖는 적절한 모니터링 기법에 대한 연구는 미비한 실정이며 모바일 장치에 대한 결함을 감지하고, 측정하기 위한 방법을 적용한 사례는 적다.

[22]에서는 전통적인 체크포인트링 기술의 오버헤드를 줄이기 위한 게으른 체크포인트 방법을 제안하였으며, 대부분의 결함은 비교적 짧은 시간 내에 다시 발생할 가능성이 있다는 실험 결과와 마지막 결함이 발생한 후에는 결함 발생률이 감소한다는 실험을 통해 결함 발생률이 낮은 기간 동안에는 체크포인트링 주기를 점차적으로 감소시켜 오버헤드를 줄이는 방법을 제공하였다. 하지만 체크포인트링 주기가 점차적으로 감소할수록 컴퓨팅 시간도 증가하게 되어 이때 결함

이 발생하면 롤백에 의해 더 많은 컴퓨팅을 다시 수행해야 하는 문제점이 있다.

[23]에서는 중복과 체크포인트링 기술을 동시에 적용하였을 경우에 대하여 100시간 이상의 다양한 응용들의 실행 시간을 줄이는 데 얼마나 효과적인지에 대한 연구와 최적의 복제 계수, 체크포인트 시간 간격을 구하기 위한 수학적 모델에 대한 연구를 진행하였다. 100시간 이상의 장시간 응용들을 대상으로 다양한 모델링을 통하여 실험을 진행하였지만 MEC 환경에서는 장시간 응용보다는 실시간 응용의 비중이 더 크기 때문에 MEC 환경에서는 적합한 방안은 아니며 MEC 응용에서의 실행시간을 줄이는 데에 얼마만큼의 효과가 있을지에 대한 고려는 부재하다.

3. 제안하는 모니터링 기법

MEC 환경에서는 높은 이동성을 가진 모바일 장치들에 대하여 엣지 클라우드 서버간 빈번한 서비스 이주가 발생하여 이에 대한 결함을 탐지하는 모니터링 기법이 필요하다.

서로 다른 엣지 클라우드 서버간의 작업 이주 시 데이터 뿐만 아니라, 모바일 디바이스의 상태정보와 서비스의 상태 정보를 전달해야 한다. MEC 환경에서 모바일 장치는 상대적으로 불안정하며, 다수가 엣지 클라우드 서버에 연결되어 있다. 엣지 클라우드 서버는 컴퓨팅 자원이 제한적이기 때문에 이주 과정에서 모바일 장치의 서비스가 제공되지 않는 순간이 발생할 수 있다[5]. 이러한 문제를 해결하기 위해서는 사용자 입장에서 실시간성이 보장되어야 하며 높은 이동성을 고려해야 하는 서비스의 QoS 보장을 위해 이주 시 생기는 결함에 대한 모니터링 연구가 필요하다.

MEC 환경에서의 결함의 기준은 서비스의 크기, 네트워크 속도, 데드라인을 고려하여 모니터링 기법을 설계할 수 있으며, 모니터링 정보를 기준으로 임계값을 초과할 경우 이를 결함으로 간주할 수 있다.

Fig. 1은 HMM 모델에 대한 그림으로 HMM에서는 은닉된 상태와 관찰 가능한 결과의 두 가지 요소로 이루어져 있다. 제안하는 모니터링 기법은 모니터링 정보를 HMM에서 관찰 가능한 상태로 간주하여 은닉된 상태에 대한 확률을 구해 결함을 측정하고 추후에 이에 대한 조치를 취할 수 있다. Table 1은 HMM에서 사용하는 기호를 나열하고 있다.

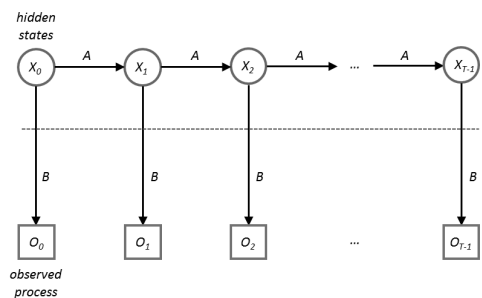


Fig. 1. The Hidden Markov Chain Model

Table 1. Notation used in the HMM

Notation	Meaning
T	The length of the observation sequence.
N	The number of states in the model.
M	The number of observation symbols.
Q	The distinct states of the Markov process, that is, $\{q_0, q_1, \dots, q_{N-1}\}$.
V	The set of possible observations, that is, $\{0, 1, \dots, M-1\}$.
A	The state transition probabilities.
B	The observation probability matrix.
π	The initial state distribution.
O	The observation sequence, that is, $\{O_0, O_1, \dots, O_{T-1}\}$.
X	The hidden state sequence, that is, $\{X_0, X_1, \dots, X_{T-1}\}$.

• **Input:**

- Model $\lambda = (A, B, \pi)$
- Observations $O = (O_0, O_1, \dots, O_{T-1})$

1. **for** $i = 0, 1, \dots, N - 1$ **do**
 $a_0(i) = \pi b_i(O_0)$
end for
2. **for** $t = 0, 1, \dots, T - 1$ **do**
for $i = 0, 1, \dots, N - 1$ **do**
 $a_t(i) = \left(\sum_{j=0}^{N-1} a_{t-1}(j) a_{ji} \right) b_i(O_t)$
end for
end for

Fig. 2. The Forward Algorithm

• **Input:**

- Model $\lambda = (A, B, \pi)$
- Observations $O = (O_0, O_1, \dots, O_{T-1})$

1. **for** $i = 0, 1, \dots, N - 1$ **do**
 $\beta_{T-1}(i) = 1$
end for
2. **for** $t = T - 2, T - 3, \dots, 0$ **do**
for $i = 0, 1, \dots, N - 1$ **do**
 $b_t(i) = \left(\sum_{j=0}^{N-1} a_{ij} b_j(O_{t+1}) \right) \beta_{t+1}(j)$
end for
end for

Fig. 3. The Backward Algorithm

HMM을 모바일 장치의 모니터링 기법에 적용하기 위해 다음의 두 문제를 해결해야 한다.

- 모델 $\lambda = (A, B, \pi)$ 와 관찰 시퀀스 O 가 주어졌을 때, $P(O|\lambda)$ 를 결정하는 문제
- 모델 $\lambda = (A, B, \pi)$ 와 관찰 시퀀스 O 가 주어졌을 때, 마르코프 프로세스의 최적의 상태 구하는 문제

먼저 $P(O|\lambda)$ 를 구하기 위해 $X = (x_0, x_1, \dots, x_{T-1})$ 를 상태 시퀀스라고 할 때 B의 정의에 의해 다음을 구할 수 있다.

$$P(O|\lambda) = b_{x_0}(O_0)b_{x_1}(O_1) \cdots b_{x_{T-1}}(O_{T-1}) \quad (1)$$

또한 π 와 A의 정의에 의해 $P(X|\lambda)$ 는 다음과 같이 구할 수 있다.

$$P(X|\lambda) = \pi_{x_0} a_{x_0, x_1} a_{x_1, x_2} \cdots a_{x_{T-2}, x_{T-1}} \quad (2)$$

$P(O, X|\lambda) = P(O|X, \lambda)P(X|\lambda)$ 이므로 가능한 모든 상태 시퀀스로 다음을 구할 수 있다.

$$\begin{aligned} P(O|\lambda) &= \sum_X P(O, X|\lambda) \\ &= \sum_X P(O|X, \lambda)P(X|\lambda) \end{aligned} \quad (3)$$

$P(O|\lambda)$ 를 효율적으로 구하기 위해 Fig. 2와 같이 전향 알고리즘을 사용할 수 있다.

마르코프 프로세스의 최적의 상태 시퀀스를 구하기 위해 모든 경로 중에서 가장 높은 값을 다음을 고려하여 선택할 수 있다.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_{T-1} | x_t = q_i, \lambda) \quad (4)$$

$\beta_t(i)$ 를 구하는 후향 알고리즘은 Fig. 3과 같다. 이제 $t = 0, 1, \dots, T-1$ 와 $i = 0, 1, \dots, N-1$ 에 대하여 $\gamma_t(i)$ 를 다음과 같이 정의한다.

$$\gamma_t(i) = P(x_t = q_i | O, \lambda) \quad (5)$$

$a_t(i)$ 는 t 시간까지의 확률을 계산하고 $\beta_t(i)$ 는 t 시간 이후의 확률을 계산하므로 $\gamma_t(i)$ 는 다음과 같이 정리할 수 있다.

$$\gamma_t(i) = a_t(i)\beta_t(i)/P(O|\lambda) \quad (6)$$

따라서 t 시간에서의 최적의 상태는 $x_t = \max \gamma_t(i)$ 이다.

4. 실험 평가

이 장에서는 제안한 결합 측정 모니터링 기법의 성능을 알아보기 위하여 이벤트 기반 프로그래밍(event driven programming) 방식을 사용하여 성능 실험을 수행하고 결과를 분석한다. 실험에서 사용한 파라미터 값은 Table 2와 같다.

Table 2. Parameters for Experiments

Parameter	Value
A	$\begin{matrix} S U \\ \frac{S}{U} \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix} \end{matrix}$
B	$\begin{matrix} L M H \\ \frac{S}{U} \begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.6 & 0.3 & 0.1 \end{bmatrix} \end{matrix}$
π	[0.4 0.6]
O	(2, 1, 0, 1, 1) (2, 1, 0, 1, 1, 2, 1, 0)

파라미터 A는 $\begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$ 으로 이는 다음과 같은 의미이다. Stable 상태일 때 다음 단계에서 Stable 상태일 확률은 60%이고 Unstable 상태일 확률은 40%이다. Unstable 상태일 때 다음 단계에서 Stable 상태로 전이될 확률은 40%이고 Unstable 상태로 전이될 확률은 60%이다. 파라미터 A에서 각 행의 모든 값의 합은 1을 가진다.

파라미터 B는 $\begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.6 & 0.3 & 0.1 \end{bmatrix}$ 으로 이는 다음과 같은 의미이다. Stable 상태일 때 모바일 장치의 모니터링 정보가 L(low)이 나타나는 경우는 10%이고 M(medium)이 나타나는 경우는 30%이고 H(high)가 나타나는 경우는 60%이다. 또한 Unstable 상태에서 모바일 장치의 모니터링 정보가 L(low)이 나타나는 경우는 60%이고 M(medium)이 나타나는 경우는 30%이고 H(high)가 나타나는 경우는 10%이다. 파라미터 B도 파라미터 A와 마찬가지로 각 행의 모든 값의 합은 1을 가진다.

파라미터 π 는 [0.4 0.6]으로 모바일 장치의 로그 정보가 없을 경우 초기에 모바일 장치가 가질 상태에 대한 확률을 나타낸다. 다시 말해, 모바일 장치의 로그 정보가 없을 경우 시스템 초기에 모바일 장치가 Stable 상태일 확률은 40%이고 Unstable 상태일 확률은 60%이다.

파라미터 O는 두 가지 경우에 대하여 실험을 실시하였으며 모바일 장치의 모니터링 정보의 시퀀스 길이가 5인 경우와 8인 경우에 대하여 그 결과를 분석한다. 파라미터 O에서 0의 의미는 L을 뜻하며, 1은 M, 2는 H를 뜻한다. 모바일 장치의 특성을 고려하여 모바일 장치의 모니터링 정보가 L → H로 전이하거나 H → L로 전이하는 경우는 포함하지 않았다.

Table 3은 $O = (2, 1, 0, 1, 1)$ 에 대한 상태 시퀀스의 확률을 보여준다. 표에서 S는 Stable을 U는 Unstable을 의미한다. 예를 들어 UUSSS는 모바일 장치의 상태가 Unstable → Unstable → Stable → Stable → Stable 로 전이되는 것을 의미하며 이에 대한 확률은 약 0.1631159%이다. UUSSS는 $|O| = 5$ 인 경우에 대하여 나타날 수 있는 확률이 가장 적은 경우이다.

반대로 SSUUU는 모바일 장치의 상태가 Stable → Stable → Unstable → Unstable → Unstable 로 전이되는 것을 의미하며 이에 대한 확률은 약 11.3204478%이다.

UUSSS는 $|O| = 5$ 인 경우에 대하여 나타날 수 있는 가장 높은 확률이다.

Table 3. Probabilities of State Sequences (The Sequence of the Maximum Probability is SSUUU)

Sequence	Probability
SSSSS	0.014360028
SSSSU	0.015185222
SSSUS	0.019021767
SSSUU	0.020114845
SSUSS	0.080816903
SSUSU	0.085461016
SSUUS	0.107052732
SSUUU	0.113204478
SUSSS	0.013864855
SUSSU	0.014661594
SUSUS	0.018365844
SUSUU	0.019421230
SUUSS	0.078030113
SUUSU	0.082514084
SUUUS	0.103361259
SUUUU	0.109300875
USSSS	0.001689415
USSSU	0.001786497
USSUS	0.002237855
USSUU	0.002366452
USUSS	0.009507871
USUSU	0.010054237
USUUS	0.012594439
USUUU	0.013318174
UUSSS	0.001631159
UUSSU	0.001724893
UUSUS	0.002160688
UUSUU	0.002284851
UUUSS	0.009180013
UUUSU	0.009707539
UUUUS	0.012160148
UUUUU	0.012858927

Table 4. HMM Probabilities (the sequence will be SSUUSSUU)

Sequence number	Probability of stable (S)	Probability of unstable (U)
0	0.8947576066	0.1052423934
1	0.5090529425	0.4909470575
2	0.1523176942	0.8476823058
3	0.4439562979	0.5560437021
4	0.5562550549	0.4437449451
5	0.8485699873	0.1514300127
6	0.4997196054	0.5002803946
7	0.1499719605	0.8500280395

Table 4는 $O = (2, 1, 0, 1, 1, 2, 1, 0)$ 에 대한 HMM의 확률을 보여준다. 표의 시퀀스 0에서 Stable 상태일 확률은 약 89.47576066% 임을 나타내며, Unstable 상태일 확률은 약 10.52423934% 임을 나타낸다. 따라서 시퀀스가 진행됨에 따라 Stable 상태가 될 확률과 Unstable 상태가 될 확률 중에서 높은 것을 선택하여 상태 정보를 유추할 수 있다.

$O = (2, 1, 0, 1, 1, 2, 1, 0)$ 에 대한 가장 높은 확률을 가지는 시퀀스는 SSUUSSUU로 이의 확률은 약 4.30941946%이다. 반대로 가장 낮은 확률을 가지는 시퀀스는 UUUSSUUU로 이의 확률은 약 0.00175953%이다.

제안하는 결함 측정 모니터링 기법을 실제 응용에 적용하기 위해서는 다음과 같은 사항들을 추가적으로 고려할 수 있다. 먼저 모바일 장치에 대한 충분한 로그 정보를 분석하여 실험 평가에서 사용한 파라미터들에 대한 사전 정보를 가지고 있어야 한다. 예를 들어 파라미터 A, B, π 에 대하여 전체 로그 정보를 파악하여 모바일 장치에서 일어나는 상태전이 정보(Stable \rightarrow Unstable 또는 Unstable \rightarrow Stable)와 모바일 장치의 모니터링 정보(L(low), M(medium), H(high))를 알고 있어야 한다.

다음으로 실험 평가에서 사용한 파라미터에 대한 분류를 더 세분화하여 적용하는 것을 고려할 수 있다. 예를 들어 파라미터 A 의 경우 Stable 상태와 Unstable 상태 두 가지로만 나누는 것이 아니라 Stable, Semi-stable, Unstable 상태 세 가지로도 나누어 고려할 수 있다. 또한 관리자가 원하는 경우 네 가지 이상의 상태로도 나누어 모바일 장치에 대한 모니터링을 수행할 수 있다. 파라미터 B 에 대해서도 L, M, H의 세 가지 수준이 아니라 더 정밀한 수준으로 나누어 모바일 장치에 대한 모니터링을 수행할 수 있을 것이다.

이전의 모니터링 정보(N 개의 상태 시퀀스)를 기반으로 현재 모바일 장치의 L, M, 또는 H의 상태에 따라 모바일 장치에서 결함이 발생할 것인지 또는 발생하지 않을 것인지에 대한 충분한 정보를 수집할 수 있으며, 이 정보를 활용하여 모바일 장치의 데이터 백업, 복제, 체크포인트, 합의 알고리즘 등의 결함에 대한 조치를 빠르게 수행할 수 있다. 또한 제안하는 결함 측정 모니터링 기법은 모바일 장치 뿐만 아니라 일반적인 클라우드 환경에서의 가상 머신(virtual machine)을 모니터링 하는 방법으로도 활용할 수 있다.

5. 결론

이 논문에서는 MEC 환경에서 인공지능 기반 모바일 장치 모니터링 기법을 제안하였다. 제안하는 모니터링 기법은 인공지능 기법 중 하나인 HMM을 적용하여 모니터링 기법을 기반으로 모바일 장치의 상태 정보를 측정하는 알고리즘을 제시하고 이를 실험 평가를 통해 검증하였다. 제안한 모니터링 기법을 적용하여 MEC 환경에서 발생할 수 있는 SLA 위반을 줄이고 결함에 대한 조치를 신속하게 대처할 수 있으며, 이를 통해 사용자의 만족도를 높일 수 있다. 수

집하는 모니터링 정보는 서비스의 크기, 네트워크 속도, 데드라인 등을 고려하여 설정할 수 있으며, 관리자의 정책에 따라 측정 정보를 세분화하여 적용할 수 있다. 실험 결과 A, B, π, O 의 파라미터 값에 대하여 모바일 장치의 상태를 확률값에 따라 Stable 또는 Unstable로 구별할 수 있는 것을 확인하였다. 향후 연구로는 재귀함수를 사용하는 알고리즘의 효율성을 증대시키기 위하여 동적 프로그래밍(dynamic programming) 방식을 적용하여 알고리즘을 개선하고 데이터 백업, 복제, 체크포인트 등 클라우드 서비스 또는 응용에 통합하여 적용하는 것이다.

References

- [1] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks," *IEEE Wireless Communications*, Vol.22, No.1, pp. 96-102, 2015.
- [2] H. Abid, L. T. T. Phuong, J. Wang, S. Lee, and S. Qaisar, "V-Cloud: vehicular cyber-physical systems and cloud computing," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Barcelona, Spain, 2011, pp.1-5.
- [3] A. B. T. Sherif, K. Rabieh, M. M. E. A. Mahmoud, and X. Liang, "Privacy-Preserving Ride Sharing Scheme for Autonomous Vehicles in Big Data Era," *IEEE Internet of Things Journal*, Vol.4, No.2, pp.611-618, 2017.
- [4] A. Daniel, K. Subburathinam, A. Paul, N. Rajkumar, and S. Rho, "Big autonomous vehicular data classifications: Towards procuring intelligence in ITS," *Vehicular Communications*, Vol. 9, pp.306-312, 2017.
- [5] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. d. Foy, and Y. Zhang, "Mobile Edge Cloud System: Architectures, Challenges, and Approaches," *IEEE Systems Journal*, Vol. PP, Issue 99, pp.1-14, 2017.
- [6] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," *IEEE/ACM Transactions on Networking*, Vol.24, No.5, pp. 2795-2808, 2016.
- [7] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: Integrating cloudlets and Mobile Edge Computing," in *2016 23rd International Conference on Telecommunications (ICT)*, pp.1-5, 2016.
- [8] S. N. Shirazi, A. Gougliadis, A. Farshad, and D. Hutchison, "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective," *IEEE Journal on Selected Areas in Communications*, Vol.35, No.11, pp.2586-2595, 2017.

[9] L. Józwiak, "Advanced mobile and wearable systems," *Microprocessors and Microsystems*, Vol.50, pp.202-221, 2017.

[10] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and Privacy in Fog Computing: Challenges," *IEEE Access*, Vol.5, pp.19293-19304, 2017.

[11] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-Enabled Tactile Internet," *IEEE Journal on Selected Areas in Communications*, Vol.34, No.3, pp.460-473, 2016.

[12] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine*, Vol.55, No.4, pp.54-61, 2017.

[13] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things," *IEEE Consumer Electronics Magazine*, Vol.5, No.4, pp.84-91, 2016.

[14] P. S. Khodashenas et al., "Service provisioning and pricing methods in a multi-tenant cloud enabled RAN," in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp.1-6, 2016.

[15] R. D. Schlichting and F. B. Schneider, "Fail-stop processors: an approach to designing fault-tolerant computing systems," *ACM Trans. Comput. Syst.*, vol. 1, no. 3, pp. 222-238, 1983.

[16] D. Hakkarinen, P. Wu, and Z. Chen, "Fail-Stop Failure Algorithm-Based Fault Tolerance for Cholesky Decomposition," *IEEE Transactions on Parallel and Distributed Systems*, Vol.26, No.5, pp.1323-1335, 2015.

[17] T. Hong-Yi and S. Kai-Yeung, "On the message and time complexity of protocols for reliable broadcasts/multicasts in networks with omission failures," *IEEE Journal on Selected Areas in Communications*, Vol.13, No.7, pp.1296-1308, 1995.

[18] H. Moniz, N. F. Neves, M. Correia, and P. Verissimo, "Randomization Can Be a Healer: Consensus with Dynamic Omission Failures," in *Distributed Computing: 23rd International Symposium, DISC 2009, Elche, Spain, September 23-25, 2009. Proceedings*, I. Keidar, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.63-77, 2009.

[19] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Program. Lang. Syst.*, Vol. 4, No.3, pp.382-401, 1982.

[20] J. Lim, K.-S. Chung, H. Lee, K. Yim, and H. Yu, "Byzantine-resilient dual gossip membership management in clouds," *Soft Computing*, pp.1-12, 2017.

[21] L. Minh, S. Zheng, Y. W. Kwon, and E. Tilevich, "Reliable and efficient mobile edge computing in highly dynamic and volatile environments," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, pp.113-120, 2017.

[22] D. Tiwari, S. Gupta, and S. S. Vazhkudai, "Lazy Checkpointing: Exploiting Temporal Locality in Failures to Mitigate Checkpointing Overheads on Extreme-Scale Systems," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp.25-36, 2014.

[23] J. Elliott, K. Kharbas, D. Fiala, F. Mueller, K. Ferreira, and C. Engelmann, "Combining Partial Redundancy and Checkpointing for HPC," in *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pp.615-626, 2012.

임종범



<http://orcid.org/0000-0001-8954-2903>

e-mail : jblim@kpu.ac.kr

2009년 백석대학교 정보통신학부(학사)

2011년 고려대학교 컴퓨터교육과(석사)

2014년 고려대학교 컴퓨터교육과(박사)

2014년~2016년 동국대학교

IT융합교육센터 초빙교수

2017년~현 재 한국산업기술대학교 게임공학부 조교수

관심분야 : Distributed & Cloud Computing

최희석



<http://orcid.org/0000-0002-5340-1989>

e-mail : hsrangken@korea.ac.kr

2012년 백석대학교 정보통신학부(학사)

2012년~현 재 고려대학교 정보대학

컴퓨터학과 박사수료

관심분야 : Energy-aware Computing,

Cloud Computing, Server

Consolidation

유현창



<http://orcid.org/0000-0003-2216-595X>

e-mail : yuhc@korea.ac.kr

1989년 고려대학교 전산과학과(학사)

1991년 고려대학교 컴퓨터학과(석사)

1994년 고려대학교 컴퓨터학과(박사)

1995년~1998년 서경대학교 컴퓨터공학과

조교수

1998년~현 재 고려대학교 컴퓨터학과 교수

관심분야 : Distributed Systems, Cloud Computing