KIPS Trans. Softw. and Data Eng.
Vol.11, No.4 pp.163~168
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2022.11.4.163

A Resource Management Scheme Based on Live Migrations for Mobility Support in Edge-Based Fog Computing Environments

JongBeom Lim†

ABSTRACT

As cloud computing and the Internet of things are getting popular, the number of devices in the Internet of things computing environments is increasing. In addition, there exist various Internet-based applications, such as home automation and healthcare. In turn, existing studies explored the quality of service, such as downtime and reliability of tasks for Internet of things applications. To enhance the quality of service of Internet of things applications, cloud-fog computing (combining cloud computing and edge computing) can be used for offloading burdens from the central cloud server to edge servers. However, when devices inherit the mobility property, continuity and the quality of service of Internet of things applications can be reduced. In this paper, we propose a resource management scheme based on live migrations for mobility support in edge-based fog computing environments. The proposed resource management algorithm is based on the mobility direction and pace to predict the expected position, and migrates tasks to the target edge server. The performance results show that our proposed resource management algorithm improves the reliability of tasks and reduces downtime of services.

Keywords : Cloud-fog Computing, Resource Management, Internet of Things, Edge Computing

에지 기반 포그 컴퓨팅 환경에서 이동성 지원을 위한 라이브 마이그레이션 기반 자원 관리 기법

임 종 범†

요 약

클라우드 컴퓨팅과 사물인터넷의 대중화에 따라 사물인터넷 컴퓨팅 환경에 존재하는 인터넷 연결이 가능한 장치들의 수가 점차 증가하고 있다. 또한 스마트홈, 헬스케어 등 사물인터넷을 이용한 다양한 인터넷 응용이 많아짐에 따라 통신 지연 및 연산의 신뢰성과 같은 지표의 서비스품질과 관련된 연구들이 진행되고 있다. 사물인터넷 응용의 서비스품질 향상을 위해 중앙집중형 클라우드 서버에 연결하기 보다 장치와 가까이 존재하고 중앙집중형 클라우드 서버와의 오프로드(offload) 협업을 위해 에지 컴퓨팅(edge computing)이 결합된 클라우드-포그 컴퓨팅 환경이 주목을 받고 있다. 하지만 클라우드-포그 컴퓨팅 환경에서 장치들이 이동성을 특성을 가질 때 사물인터넷 응용 서비스의 연속성이 떨어지고 서비스품질 수준이 저하되는 문제점이 발생하고 있다. 이 논문에서는 에지 기반 포그 컴퓨팅 환경에서 이동성 지원을 위한 라이브 마이그레이션 기반 자원 관리 기법을 제안한다. 제안하는 자원 관리 알고리즘은 사용자의 이동성 방향과 속도를 기반으로 일정 시간 뒤의 위치를 예측하고 이를 기반으로 라이브 마이그레이션을 통해 사물인터넷 서비스 이주를 지원한다. 성능 평가를 통해 제안하는 자원 관리 알고리즘의 효용성을 측정하였으며, 성능 실험에서 정지 시간(downtime)과 서비스 작업의 신뢰성이 크게 향상됨을 보였다.

키워드 : 클라우드-포그 컴퓨팅, 자원 관리, 사물인터넷, 에지 컴퓨팅

1. 서 론

클라우드 컴퓨팅 환경에서 중앙집중형 클라우드 서버는 사물인터넷 장치로부터 물리적으로 먼 곳에 위치한다. 따라서 데이터 전송 시 지연이 자주 발생하고, 사물인터넷 서비스와 장치, 서비스 사용자의 수가 나날이 증가함에 따라 끊임없이 수집되고 생성되는 대용량의 데이터를 효율적으로 처리하기 어렵다[1-3]. 사물인터넷 및 산업사물인터넷 서비스의 급증과 클라우드 컴퓨팅의 대중화에 따라 사물인터넷 서비스 응용을 클라우크 컴퓨팅 기술을 이용하여 배포할 수 있는 클라우드-포그 컴퓨팅이 주목을 받고 있다[4-6].

클라우드-포그 컴퓨팅 환경에서 스마트홈, 헬스케어 등 사물인터넷을 이용한 다양한 인터넷 응용이 많아짐에 따라 통신 지연 및 연산의 신뢰성과 같은 지표의 서비스품질과 관련된 연구들이 진행되고 있다[7-10]. 하지만 클라우드-포그 컴퓨팅 환경에서 장치들이 이동성을 특성을 가질 때 사물인터넷 응용 서비스의 연속성이 떨어지고 서비스품질 수준이 저하되는 문제점이 발생하고 있다[11-14].

이 논문에서는 에지 기반 포그 컴퓨팅 환경에서 이동성 지원을 위한 라이브 마이그레이션 기반 자원 관리 기법을 제안한다. 제안하는 자원 관리 알고리즘은 사용자의 이동 방향과

※ 이 논문은 2021학년도 평택대학교 학술연구비의 지원에 의하여 연구되었음.
† 종신회원 : 평택대학교 ICT융합학부 조교수
Manuscript Received : December 28, 2021
First Revision : January 26, 2022
Accepted : January 29, 2022
* Corresponding Author : JongBeom Lim(jblim@ptu.ac.kr)

속도를 기반으로 일정 시간 뒤의 위치를 예측하고 라이브 마이그레이션을 통해 사물인터넷 서비스 이주를 지원한다. 라이브 마이그레이션의 장점은 마이그레이션이 수행되는 것을 사용자가 인지하지 못하는 투명성을 제공함과 동시에 정지시간이 짧다는 것이다. 하지만 클라우드-포그 컴퓨팅 환경에서 콜드 마이그레이션과 컨테이너 마이그레이션 대한 연구가 진행되었고, 라이브 마이그레이션에 대한 연구는 미비하였다. 이에, 성능 실험에서는 제안 기법과 기존 마이그레이션 기법과의 성능 평가를 진행한다.

기존 연구와의 차별성은 다음과 같다. 제안하는 자원 관리 기술은 클라우드 서버와 말단 장치 사이에 서버를 위치시키는 에지 컴퓨팅 기술을 통해 데이터 연산과 스토리지, 네트워킹 관련 서비스의 연속성을 제공하는 연구 내용을 포함한다. 구체적으로, 제안하는 자원 관리 기법을 사용하여 클라우드와 신속히 연결될 수 있도록 하는 방법과 사용자 가까이에 위치하는 에지 서버 연결을 통해 실시간 처리와 서비스 이주, 이동성 지원을 제공한다. 또한 제안 연구는 다음과 같은 응용 분야 및 활용성을 갖는다.

- 반자동 및 자율주행 자동차에서 생성되는 데이터가 폭증하더라도 커넥티드 차량에 대한 데이터를 독립적으로 처리하여 주변 환경, 주행 상태, 주행 방향 등의 데이터를 실시간으로 처리할 수 있는 프레임워크를 제공한다.

- 클라우드 서비스를 사용하는 데이터에 대해 클라우드에서는 물론 종단점에서의 통신을 가능하게 하여, 다수의 사물인터넷 장치의 센서로부터 수집해야 하는 경우 제안하는 자원 관리 기술 및 알고리즘을 사용하여 데이터의 효율적인 처리를 가능하게 한다.

논문의 구성은 다음과 같다. 2장에서는 연구의 배경 및 관련 기술을 소개하고, 3장에서는 라이브 마이그레이션을 지원하는 이동성과 서비스 이주를 위한 자원 관리 알고리즘을 설명한다. 4장에서는 제안 연구의 성능 평가와 기존 연구와의 성능 비교를 통해 성능 특성을 분석하고, 5장에서는 논문의 결론을 맺는다.

2. 연구의 배경 및 동기

클라우드-포그 컴퓨팅 환경은 수많은 사물인터넷 장치들이 중앙집중형 클라우드 서버에 연결하는 구조가 아닌 장치와 가까운 에지 서버와 연결하여 서비스 및 데이터 통신을 지원하는 환경이다[15-17]. 클라우드-포그 컴퓨팅 구조에서는 클라우드 컴퓨팅 환경에서 제공하는 컴퓨팅 자원을 즉각적으로 대여하여 사용할 수 있고, 가상 머신 및 컨테이너 등의 가상화 기술을 이용할 수 있는 구조이다.

Fig. 1은 클라우드-포그 컴퓨팅의 구조를 보여준다. 기본적으로 사물인터넷 장치들은 서버와의 통신을 통하여 필요한 데이터들을 내려받아 서비스를 제공할 수 있다. 여기서 에지 컴퓨팅과 포그 컴퓨팅의 미묘한 차이가 있다. 에지 컴퓨팅은 데이터에 대한 연산을 지리적으로 가까운 에지 서버에서 수행하는 것을 말하며, 포그 컴퓨팅은 에지 서버와 클라우드 서버와의 중재자 역할을 수행한다. 다시 말해, 포그 컴퓨팅은 에지 컴퓨팅을 대체할 수 없으며, 반대로 에지 컴퓨팅은 포그 컴퓨팅 없이 여러 응용들을 수행할 수 있다.

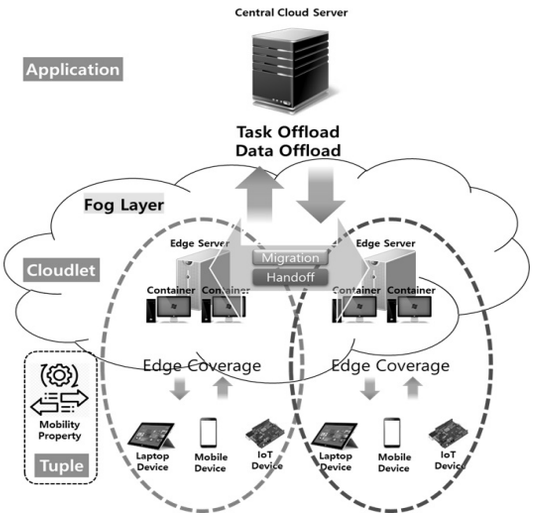


Fig. 1. Cloud-fog Computing Architecture

사물인터넷 장치들은 통신 지연 및 속도가 상대적으로 느린 중앙집중형 클라우드 서버에 직접 연결하는 것이 아니라 사물인터넷 장치 주변의 에지 서버와의 연결을 통해 저지연 및 실시간성의 서비스를 지원할 수 있다. 에지 서버는 사물인터넷 장치에서의 서비스 지원을 위해 필요한 서비스와 관련된 데이터 등은 오프로드 기능을 통해 중앙집중형 클라우드 서버로부터 사전에 내려받아야 한다.

하지만 에지 서버는 서비스 지원 및 연결에 대한 커버리지가 한정되어 있어 사물인터넷 장치가 에지 서버의 커버리지를 벗어나면 서비스를 제공하지 못하는 단점이 발생한다. 기존의 클라우드-포그 컴퓨팅 관련 연구에서는 사물인터넷 장치의 이동성을 고려하지 않은 환경 내에서의 자원 관리 기법들이 연구되었다.

따라서 제안 연구에서는 에지 기반 포그 컴퓨팅 환경에서 사물인터넷 장치들이 이동성의 특성을 가지는 경우, 서비스 품질 및 작업의 신뢰성 확보를 위한 라이브 마이그레이션 기술을 통해 서비스 이주를 지원하는 자원 관리 알고리즘을 제안한다. 제안하는 자원 관리 알고리즘은 사물인터넷 장치가 가지는 이동성의 특성(현재 위치, 이동 방향, 이동 속도)을 반영하여 일정 시간 뒤의 위치를 예측하고 기존 에지 서버와 자원 관리 알고리즘에 의해 선택된 다른 에지 서버와의 라이브 마이그레이션을 통해 적은 정지시간 및 작업의 신뢰성을 보장할 수 있다.

3. 제안하는 자원 관리 알고리즘

이 장에서는 에지 기반 포그 컴퓨팅 환경에서 이동성과 서비스 이주를 지원하는 자원 관리 알고리즘에 대해 설명한다. Fig. 2는 제안하는 자원 관리 알고리즘의 기본적인 개념을 보

여준다. Fig. 2(a)는 자신이 관리하는 에지 서버들을 주기적으로 모니터링하는 중앙집중형 클라우드 서버와 액세스 포인트(AP), 에지 서버 A와 B, 이동중인 사용자를 보여준다. 이 상황에서 사용자는 에지 서버 A에 연결되어 있으며, 사용자는 화살표 방향으로 이동중이다. 제안하는 알고리즘은 특정 시간 이후에 현재 연결중인 에지 서버와의 커버리지를 벗어난다고 판단하는 경우에 자원 관리 알고리즘이 작동되어 작업과 데이터에 대한 라이브 마이그레이션이 수행된다.

Fig. 2(b)는 제안하는 자원 관리 알고리즘이 수행된 이후의 상황을 도식적으로 보여준다. 중앙집중형 클라우드 서버에서 사용자의 현재 위치, 이동 방향, 이동 속도에 대한 파라미터를 받아온 후 특정 시간 이후의 사용자의 위치를 예측한다. 이렇게 예측된 사용자의 위치를 기반으로 예측된 위치에서 가장 가까운 다른 에지 서버를 탐색 후 기존 에지 서버에서 다른 에지 서버로 라이브 마이그레이션이 수행된다. 하지만 예측된 지점에서 서비스가 가능한 에지 서버를 찾지 못하는 경우도 발생할 수 있다. 이런 경우에는 예측된 지점 또는 예측된 지점 근처에 새로운 에지 서버를 프로비저닝하여 새롭게 프로비저닝 된 에지 서버로 라이브 마이그레이션을 수행한다.

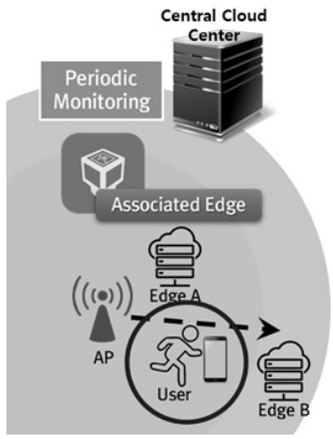
Fig. 3은 제안하는 자원 관리 알고리즘을 보여준다. 알고리즘의 입력값은 User_i, FogTask_ij, CloudServer, EdgeSet. Map_before 이고, 출력값은 Map_after 이다. 여기서 FogTask_ij는 User_i의 j번째 작업을 의미하고, EdgeSet는 모든 에지 서버, 가상 머신, 컨테이너를 포함하는 자료구조이다. Map_before와 Map_after는 FogTask_ij가 어떤 에지 서버에 매핑되어 있는지를 나타내는 자료구조이다.

초기화 과정으로 User_i의 현재 위치, 이동 속도, 이동 방향, 알고리즘 정책, 임계값, 초(pred_sec)와 미터(pred_meter)에 대한 파라미터 값을 가져온다. 여기서 pred_sec와 pred_meter는 각각 몇 초 후에 몇 미터 만큼의 거리를 예측할 것인가에 대한 전역 시스템 파라미터 값이다.

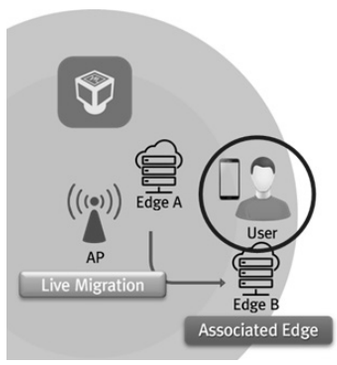
알고리즘의 의사코드는 다음과 같다. 먼저 prologue() 함수를 통해 사용자의 작업이 라이브 마이그레이션이 필요한지 여부를 판단한다. 만약 라이브 마이그레이션이 필요하다고 판단되면 migrate() 함수를 통해 라이브 마이그레이션이 수행된다. prologue() 함수의 반환 값은 bool_migration이고, migrate() 함수의 반환 값은 Map_after이다. 각 함수의 자세한 설명은 다음과 같다.

함수 prologue() 에서는 check_condition() 함수에 현재 위치, 초와 미터에 대한 파라미터를 입력값으로 설정하여 라이브 마이그레이션에 대한 상태 정보(cond)를 가져온다. 이 상태 정보 값과 정책(policy) 파라미터에 따라 라이브 마이그레이션이 필요한 정도(degree_migration)를 계산한다. 여기서 degree_migration의 값은 0과 1사이의 값은 갖는다. 이 값이 1에 가까울수록 라이브 마이그레이션이 필요하다는 의미를 갖는다.

정책에 사용되는 파라미터 값은 총 세가지(latency, cloudlet, AP)이다. 정책 latency는 사용자와 클라우드렛 사이의 통신 지연에 기반한 정책이고, cloudlet은 사용자와 클라우드렛 사이



(a) A user is Getting Out AP's Coverage



(b) After User Movement

Fig. 2. Basic Idea of the Proposed Algorithm

의 거리, AP는 사용자와 액세스포인트 사이의 거리에 기반한 정책이다. degree_migration값에 따라 이 값이 threshold보다 크면 라이브 마이그레이션이 수행되고 그렇지 않으면 라이브 마이그레이션을 수행하지 않는다.

함수 migrate() 에서는 cond, pace, direction값을 기반으로 pred_sec 초 후의 예상되는 사용자의 위치 값을 가져온다. 다음으로 find_edge() 함수를 통해 예측된 위치에서 pred_meter 반경 내에 존재하는 접속 가능한 클라우드렛을 검색한다. 이 단계에서 접속 가능한 클라우드렛을 찾았다면 정책에 따라 라이브 마이그레이션을 수행할 에지 서버를 선택하고, 만약 접속 가능한 클라우드렛을 찾지 못했다면 예측된 위치를 기반으로 새로운 클라우드렛을 프로비저닝한다.

이렇게 선택된 타겟 에지 서버와 기존에 FogTask_ij가 존재했던 에지 서버와의 라이브 마이그레이션을 수행한다. 라이브 마이그레이션의 수행 과정은 클라우드 컴퓨팅에서 제공하는 가상화 기술에 기반하며 제안하는 자원 관리 알고리즘에서는 컨테이너 라이브 마이그레이션을 수행하는 구조이다. 라이브 마이그레이션 이후에는 소스 에지 서버와 타겟 에지 서버와의 핸드오프를 수행하고, 라이브 마이그레이션이 완료된 이후에는 FogTask_ij에 대하여 타겟 에지 서버를 CloudServer에 등록하는 과정이 수행된다. 마지막으로 FogTask_ij와 새로운 매핑 정보인 Map_after를 반환한다.

```
• Input
  - User_i
  - FogTask_ij (User_i's j-th fog task)
  - CloudServer
  - EdgeSet (Edge_i, VM_j, Container_k)
  - Map_before → (FogTask_ij, Edge_i)
• Output
  - Map_after → (FogTask_ij, new_Edge_i)
• Initialization
  - loc ← get_user_location (User_i);
  - pace ← get_user_pace (User_i);
  - direction ← get_user_direction (User_i);
  - policy ← get_parameter (CloudServer, Param_1);
  - threshold ← get_parameter (CloudServer, Para_2);
  - pred_sec ← get_parameter (CloudServer, Para_3);
  - pred_meter ← get_parameter (CloudServer, Para_4);
• Pseudocode
bool_migration ← call prologue (Map_before);
if (bool_migration) then
  Map_after ← call migrate (Map_before);
  return Map_after;
else
  return null;
end if
function prologue (Map_before)
  cond ← check_condition (loc, pred_sec, pred_meter);
  degree_migration ← calc_degree (cond, policy);
  if (degree_migration 〉 threshold) then
    bool_migration ← true;
  else
    bool_migration ← false;
  end if
  return bool_migration;
end function
function migrate (Map_before)
  expected_position ← predict (cond, pace, direction);
  near_edge ← find_edge (expected_position, EdgeSet);
  if near_edge then
    new_edge ← near_edge;
  else
    new_edge ← provision (expected_position, cloudlet);
  end if
  transfer_data (Map_before, new_edge);
  handoff_sync (Map_before, new_edge);
  associate_edge (FogTask_ij, new_edge, CloudServer);
  return Map_after → (FogTask_ij, new_edge);
end function
```

Fig. 3. The Proposed Resource Management Algorithm

4. 실험 평가

이 장에서는 제안한 자원 관리 알고리즘의 성능을 알아보기 위하여 수행한 성능 실험 결과를 기술한다. 기존의 연구 내용과의 성능 비교를 위해 베이스라인 알고리즘 두 가지를 구현하고 성능 실험을 수행하였다. 첫 번째 베이스라인 알고리즘은 클라우드-포그 컴퓨팅 환경에서 콜드 가상 머신 마이그레이션을 하는 것이고, 다른 하나는 콜드 컨테이너 마이그레이션을 수행하는 것이다.

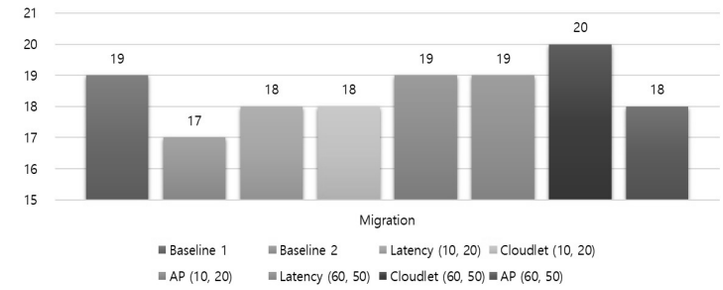
클라우드-포그 컴퓨팅 환경에서 사용자의 이동성에 대한 성능 실험을 위해 SUMO((simulation of urban mobility) 데이터셋[18,19]을 사용하였으며, 마이그레이션의 수, 정지 시간, 실패한 작업의 수, 전체 경로 중 마주친 클라우드렛의 수를 측정하였다. 이 중 전체적인 성능에 영향을 끼치는 요소는 정지 시간과 작업의 신뢰성의 척도가 되는 실패한 작업의 수이다. 클라우드-포그 환경에서 정지 시간이 길면 길수록 사용자의 서비스를 제공하지 못하는 시간이 증가함에 따라 가용성 측면에서 불이익을 받는다. 실패한 작업의 수는 사용자에 대한 작업이 실패함에 따라 작업의 신뢰성이 저하되어 전체적인 성능이 저하된다.

Fig. 4는 베이스라인 알고리즘과 제안한 알고리즘에 대한 성능 결과를 보여준다. 그림에서 Latency, Cloudlet, AP는 3장에서 설명한 정책을 나타내고, 괄호안의 두 숫자는 각각 pred_sec와 pred_meter를 의미한다. Fig. 4(a)에서 전체 마이그레이션의 수는 Baseline 2가 가장 적고, Cloudlet(60, 50)이 가장 많게 나타났다. 여덟 가지 비교군의 평균값은 18.5번이다. Fig. 4(b)는 마이그레이션 중 발생한 정지 시간을 보여준다. 베이스라인 1과 2는 각각 약 97초, 59초의 정지 시간을 가진 반면, 제안 알고리즘의 여섯 항목에 대해서는 20초 미만의 정지 시간을 갖는 것을 보여준다. 정지 시간에 대하여 베이스라인 알고리즘과 제안한 알고리즘의 평균을 계산하였을 때 제안한 알고리즘은 베이스라인 알고리즘에 비해 약 24% 정도의 정지 시간을 갖는 것으로 나타났다.

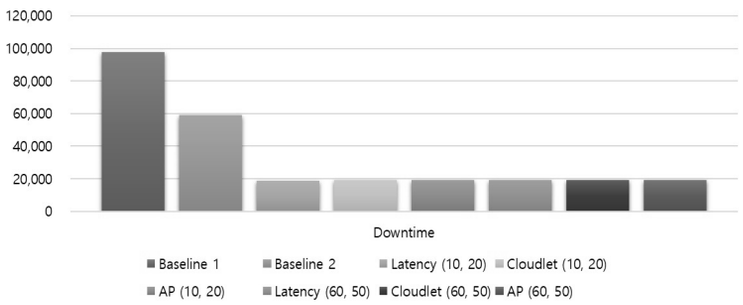
Fig. 4(c)는 수행을 완료하지 못하고 실패한 작업에 대하여 백분율과 여덟 개의 알고리즘에 대한 누적 그래프를 보여준다. 베이스라인 알고리즘 두 가지는 각각 13.30%, 12,49%의 작업 실패율을 보인 반면, 제안한 알고리즘은 평균 0.04%의 작업 실패율을 가지는 것으로 나타났다. 전체 작업의 개수가 약 1,200만개 정도인 것을 감안하면 차이가 크다는 것을 알 수 있다. 이러한 결과 차이가 나오는 것은 마이그레이션 중 발생한 정지 시간에 큰 영향이 있다.

Fig. 4(d)는 사용자의 전체 경로 중 마주친 클라우드렛의 개수를 보여준다. 여덟 가지 알고리즘의 평균값은 42.75개로 알고리즘 사이에 큰 차이는 없는 것으로 나타났다. 이 실험의 결과로 알 수 있는 점은 사용자와 클라우드렛 사이의 통신 지연을 기반으로 한 정책(latency)이 비교적 마주친 클라우드렛의 수가 많게 나왔고, 사용자와 클라우드렛 사이를 기반으로 한 정책(cloudlet)이 비교적 마주친 클라우드렛의 수가 적게 나왔다는 것이다.

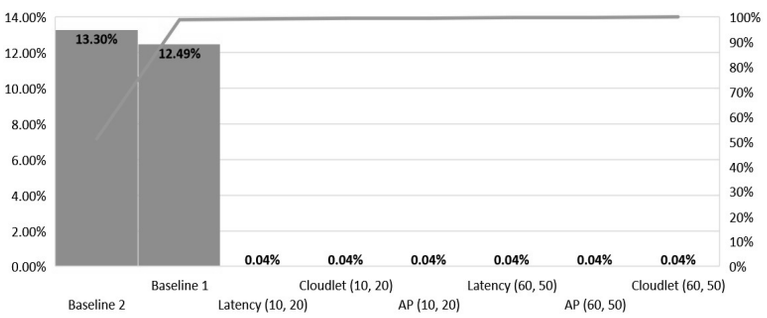
같은 정책이라고 할지라도 정책의 파라미터에 따라 결과가 다르게 나온 것도 확인하였다. 파라미터 pred_sec와 pred_meter의 값을 각각 (10, 20), (60, 50)으로 하였을 때의 전체적인 성능 차이가 많다고는 할 수 없지만, (60, 50)의 값을 주었을 때 전체 마이그레이션의 수가 비교적 많으며, 마주친 클라우드렛의 수 편차가 비교적 크다는 것을 알 수 있다. 따라서 클라우드-포그 환경에서 마이그레이션 횟수의 중요도와 클라우드렛의 분포도에 따라 환경에 적합한 파라미터를 적용하는 것이 가능함을 보였다.



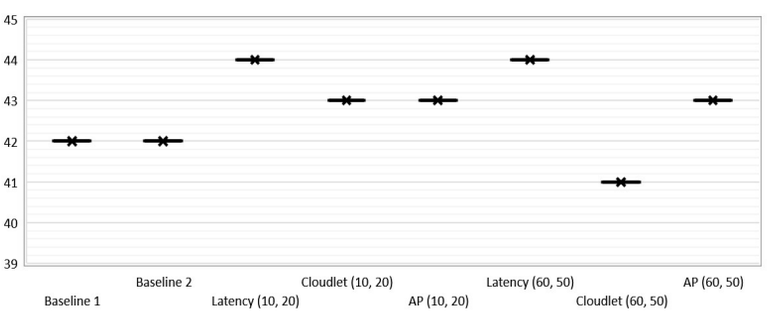
(a) The Number of Migrations



(c) Downtime (Milliseconds)



(c) The Number of Lost Tasks and Its Cumulative Distribution



(d) The Number of Encountered Cloudlets

Fig. 4. Performance Results

5. 결 론

이 논문에서는 에지 기반 포그 컴퓨팅 환경에서 라이브 마이그레이션을 이용하여 이동성과 서비스 이주를 지원하는 자원 관리 알고리즘을 제안하였다. 제안한 알고리즘은 사용자의 이동성을 지원하여 특정 시간 이후의 사용자의 위치를 예측하고 라이브 마이그레이션을 수행함으로써 정지 시간을 줄이고 작업의 신뢰성을 높였다. 또한, 통신 지연, 클라우드렛 사이의 거리, 액세스포인트와의 거리를 기반으로 다양한 정책을 적용하는 것을 가능하게 하며, 에지 기반 포그 컴퓨팅 환경의 특성 및 고유 정보에 따라 적합한 정책이 적용될 수 있음을 보였다. 성능 비교를 통해 기존 베이스라인 알고리즘과 비교했을 때 제안 알고리즘의 정지 시간은 약 24%를 가지며, 실패한 작업의 수는 약 0.53% 수준이다.

References

[1] N. Zhang, C. Zhang, and D. Wu, "Construction of a smart management system for physical health based on IoT and cloud computing with big data," *Computer Communications*, Vol.179, pp.183-194, 2021.

[2] W. Fang, F. Xue, Y. Ding, N. Xiong, and V. C. M. Leung, "EdgeKE: An on-demand deep learning IoT system for cognitive big data on industrial edge devices," *IEEE Transactions on Industrial Informatics*, Vol.17, No.9, pp.6144-6152, 2021.

[3] G. Aceto, V. Persico, and A. Pescapé, "Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0," *Journal of Industrial Information Integration*, Vol.18, pp.100129, 2020.

[4] R. Zhu, S. Li, P. Wang, Y. Tan, and J. Yuan, "Gradual migration of co-existing fixed/flexible optical networks for cloud-fog computing," IEEE Access, Vol.8, pp.50637-50647, 2020.

[5] J. Feng, L. T. Yang, R. Zhang, W. Qiang, and J. Chen, "Privacy preserving high-order bi-lanczos in cloud-fog computing for industrial applications," *IEEE Transactions on Industrial Informatics*, pp.1-1, 2020.

[6] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective task scheduling in cloud-fog computing using goal programming approach," *Cluster Computing*, Vol.25, No.1, pp.141-165, 2021.

[7] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," *International Journal of Communication Systems*, Vol.33, No.8, pp.e4340, 2020.

[8] F. Murtaza, A. Akhunzada, S. U. Islam, J. Boudjadar, and R. Buyya, "QoS-aware service provisioning in fog computing," *Journal of Network and Computer Applications*, Vol.165, pp.102674, 2020.

[9] J. C. Guevara and N. L. S. da Fonseca, "Task scheduling in cloud-fog computing systems," *Peer-to-Peer Networking and Applications*, Vol.14, No.2, pp.962-977, 2021.

[10] S. K. Mani and I. Meenakshisundaram, "Improving quality-of-service in fog computing through efficient resource allocation," *Computational Intelligence*, Vol.36, No.4, pp.1527-1547, 2020.

[11] D. Gonçalves, C. Puliafito, E. Mingozzi, O. Rana, L. Bittencourt, and E. Madeira, "Dynamic network slicing in fog computing for mobile users in MobFogSim," In *Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, Leicester, UK, 7-10 pp.237-246, Dec. 2020.

[12] R. M. Abdelmoneem, A. Benslimane, and E. Shaaban, "Mobility-aware task scheduling in cloud-Fog IoT-based healthcare architectures," *Computer Networks*, Vol.179, pp.107348, 2020.

[13] J. P. Martin, A. Kandasamy, and K. Chandrasekaran, "Mobility aware autonomic approach for the migration of application modules in fog computing environment," *Journal of Ambient Intelligence and Humanized Computing*, Vol.11, No.11, pp.5259-5278, 2020.

[14] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-Enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, Vol.69, No.5, pp.5481-5493, 2020.

[15] V. Porkodi et al., "Resource provisioning for cyber–physical–social system in cloud-fog-edge computing using optimal flower pollination algorithm," *IEEE Access*, Vol.8, pp.105311-105319, 2020.

[16] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research," *Internet of Things*, Vol.12, pp.100273, 2020.

[17] Y. Kalyani and R. Collier, "A systematic survey on the role of cloud, fog, and edge computing combination in smart agriculture," *Sensors*, Vol.21, No.17, pp.5922, 2021.

[18] A. Oliveira and T. Vazao, "Generating synthetic datasets for mobile wireless networks with SUMO," in *Proceedings of the 19th ACM International Symposium on Mobility Management and Wireless Access*, Alicante, Spain, pp.33-42, 2021.

[19] J. J. Gonzalez-Delicado, J. Gozalvez, J. Mena-Oreja, M. Sepulcre, and B. Coll-Perales, "Alicante-murcia freeway scenario: A high-accuracy and large-scale traffic simulation scenario generated using a novel traffic demand calibration method in SUMO," *IEEE Access*, Vol.9, pp.154423-154434, 2021.

임 종 범

https://orcid.org/0000-0001-8954-2903
e-mail : jblim@ptu.ac.kr
2009년 백석대학교 정보통신학부(학사)
2011년 고려대학교 컴퓨터교육과(석사)
2014년 고려대학교 컴퓨터교육과(박사)
2015년 ~ 2017년 동국대학교 IT융합교육센터 초빙교수
2017년 ~ 2021년 한국공학대학교 게임공학부 조교수
2021년 ~ 현 재 평택대학교 ICT융합학부 조교수
관심분야 : Distributed & Cloud Computing, Resource Management, Artificial Intelligence Application