

효율적 자원 배치를 위한 이동객체의 최적 이동패턴 추출

조호성 · 남광우 · 장민석 · 이연식*

군산대학교

Optimal Moving Pattern Extraction of the Moving Object for Efficient Resource Allocation

Ho-Seong Cho · Kwang-Woo Nam · Min-Seok Jang · Yon-Sik Lee*

Kunsan National University

E-mail : hoseong4820@gmail.com / yslee{kwnam,msjang}@kunsan.ac.kr

요 약

본 논문은 Fog/Edge Computing(FEC) 환경에서 애플리케이션 서비스의 사용자 근접성을 지원하는 컴퓨팅 리소스의 최적 할당 및 지연시간 감소를 위한 이동에이전트 기반 오프로딩의 효율성을 제고하기 위한 선행연구로써, 이동객체들의 방대한 시공간 이동 이력데이터 집합으로부터 복합적인 시공간 제약을 적용한 최적 이동패턴 추출 시 수행시간 및 소요 메모리량을 효과적으로 감소시키는 알고리즘을 제안한다. 제안 알고리즘은 빈발도 기반의 최적경로 추출을 통하여 향후 FEC 환경에서 계산 오프로딩을 위한 컴퓨팅 리소스의 분배 및 배치에 유용하게 사용될 수 있다.

ABSTRACT

This paper is a prior study to improve the efficiency of offloading based on mobile agents to optimize allocation of computing resources and reduce latency that support user proximity of application services in a Fog/Edge Computing (FEC) environment. We propose an algorithm that effectively reduces the execution time and the amount of memory required when extracting optimal moving patterns from the vast set of spatio-temporal movement history data of moving objects. The proposed algorithm can be useful for the distribution and deployment of computing resources for computation offloading in future FEC environments through frequency-based optimal path extraction.

키워드

Fog/Edge 컴퓨팅, 계산 오프로딩, 자원 분배 및 배치, 이동객체, 최적 이동패턴

I. 서 론

사용자 중심의 차세대 컴퓨팅 서비스를 위하여 클라우드 서비스를 새로운 FEC (Fog/Edge Computing) 패러다임에서 제안되는 모바일 네트워크의 가장자리(edge cloud)로 이동시켜 사용자 장비와의 근접성을 확보하는 방법이 요구된다. 이러한 환경에서 응용 서비스를 요구하는 엣지 디바이스(이동객체)들의 시공간 속성 변화에 따른 동적 위치 변화 패턴을 예측하여 FEC 환경에서 컴퓨팅 리소스를 효율적으로 분배 및 배치하는 방법에 대한 연구가 필요하다.

본 논문은 FEC 환경에서 이동에이전트 기반 오프로딩 기술 개발의 선행연구로써, 임의의 시공간 환경에서 다수의 이동객체들의 이동경로들을 빈도수 기반으로 분석하여 최적 이동경로를 추출하는 방법을 제시함으로써, 네트워크 가장자리에 컴퓨팅 리소스들을 효율적으로 배치할 수 있는 방법을 제안하고자 한다.

II. 최적 이동패턴 추출 제안

2.1 이동객체 및 빈발 이동시퀀스

무선 센서네트워크의 엣지 디바이스에서 요구되는 응용은 시간의 흐름에 따라 이동하며 요구 자원

* corresponding author

과 환경이 지속적으로 변하는 특성을 가지는 시공간 이동객체이다. 본 논문에서는 이동객체의 위치가 연속적으로 변하는 이산적 시점에서 획득됨을 가정하고, 센서네트워크 상에서 요구되는 응용을 이동객체로 설정한다. 이동객체들의 이동패턴 추출을 위하여 이동 이력 데이터베이스를 이용하여 인접 공간영역 간의 최대 시간간격 하에서 최소지지도도를 만족하는 이동시퀀스를 빈발 이동시퀀스로 정의하고, 빈발도 적용 최적 이동패턴은 정해진 지점들 사이의 모든 빈발 이동시퀀스들을 탐색하여 최대지지도도를 가지는 패턴으로 정의하여 최적경로를 추출한다.

2.2 최적 이동패턴 추출

이동패턴 추출은 이동객체들의 이동 이력데이터로부터 이동시퀀스를 생성하기 위한 트랜잭션을 생성하는 과정과 최적 이동패턴 추출과정으로 구분한다.

1) 이동시퀀스 추출

이동패턴 추출을 위해서는 이동객체의 이동 공간 이력데이터에 시간간격 제약조건 적용 및 데이터 일반화를 통해 생성된 이동시퀀스를 사용한다. 다음 [그림 1]은 이동객체가 이동 가능한 시공간속성 네트워크상에서의 이동객체의 이동경로에 따른 물리적 요소를 나타낸다.

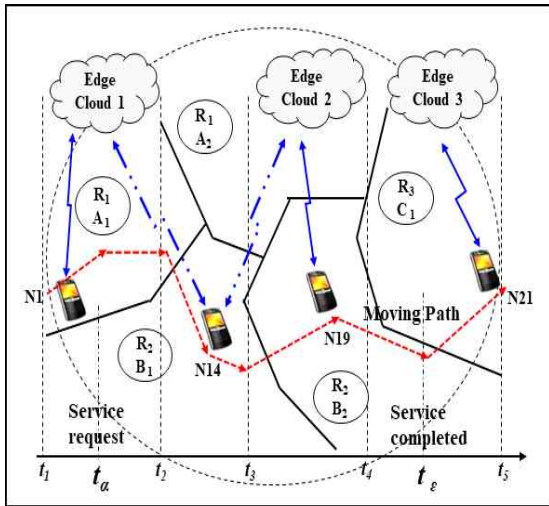


그림 1. 이동객체의 이동경로에 따른 물리적 요소

공간상의 특정 지점들 간의 최적 이동경로 추출을 위하여 상세 수준의 이동경로가 제공되어야하므로, 본 논문에서는 시공간 속성에서 최하위 수준으로 일반화된 각 이동객체들의 이동공간 노드들을 연결한 리스트 (예: [그림 1]에서 N1-N14-N19-N21) 이동시퀀스 집합으로 사용하며, 빈발 2-node 시퀀스 집합(예: N1-N14, N14-N19, N19-N21)으로부터 후보 시퀀스 집합들을 순차적으로 생성하고 지지도를 적

용하여 출발지부터 목적지까지의 빈발 시퀀스를 생성한다. 이 과정에서 후보 시퀀스들의 항목 차수가 점차 감소하므로 연산시간을 감소시킬 수 있으며, 시공간 속성으로 일반화된 데이터 집합을 사용하므로 패턴 지식화를 통하여 FEC 컴퓨팅 환경에서 요구되는 컴퓨팅 리소스 분배 및 배치에 적용이 가능하다.

2) 빈발도 기반 최적 이동패턴 추출 알고리즘

빈발도 기반 최적 이동패턴 추출 알고리즘은 최대지지도도를 적용한 2-node 시퀀스 생성모듈과 최적 이동패턴 추출모듈로 구성한다. 2-node 시퀀스 생성모듈은 최초 출발지(S)로부터 시작하는 2-node 시퀀스(<nodeS-nodeN>)들을 구한 후 각각에 대한 지지도를 산출하여 최대지지도도를 갖는 2-node 시퀀스를 추출한다. 추출된 2-node 시퀀스의 연결노드인 nodeN을 기준으로 주어진 공간 영역 내에서 목적지(D)를 포함하는 모든 부분 이동시퀀스들의 집합들에 대하여 nodeN으로부터 시작하는 2-node 시퀀스 생성과정을 반복 수행한다. 이 과정은 nodeN이 D이면 종료하고, S로부터 순차적으로 연결된 nodeN들의 리스트를 최적 이동패턴으로 결정하여 반환한다.

제안 알고리즘을 이용한 최적이동패턴 추출과정은 다음 [그림 2]와 같다.

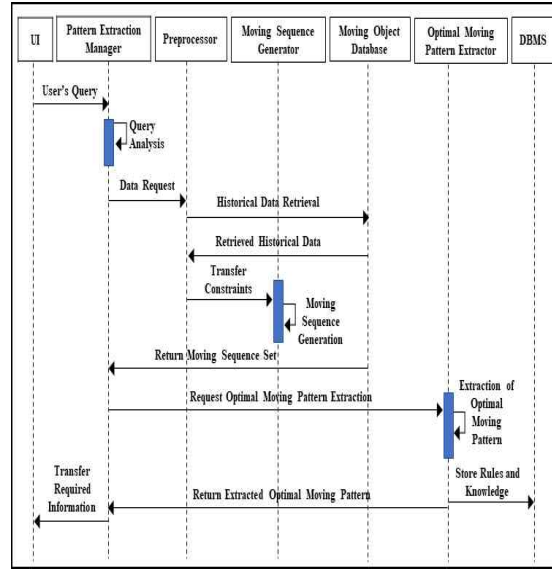


그림 2. 최적 이동패턴 추출 과정

본 논문에서 제안한 최적 이동패턴 추출방법은 사용자에게 의해 주어지는 시공간제약을 만족하는 빈발패턴을 추출하며, 이동에 따른 가중치(시간, 거리, 혼잡도 비용 등)를 적용하지 않고 빈발도를 기준으로 최적 이동패턴을 추출하므로 다른 알고리즘들(A*, Dijkstra 알고리즘)에 비하여 실행속도 및 메모리 소요량 측면에서 효율적이다. 이러한 특징을 FEC환경에서 엣지 디바이스의 응용서비스를 효율적

으로 실행하기 위하여 이동경로에 근접한 에지 클라우드 서버에 컴퓨팅 리소스를 분배하거나 새로운 에지 서버를 배치하는데 적용함으로써, 클라우드 컴퓨팅의 계산 지연시간 문제를 해결하기 위한 효율적 계산 오프로딩 환경 구축에 활용이 가능하다.

III. 실험 및 평가

제안 알고리즘의 성능을 A^* (구간범위: 5~10Km) 및 Dijkstra 알고리즘을 대상으로 최적 이동패턴 추출의 효율성과 정확성 기준의 비교실험 결과를 통하여 평가한다. 최적 이동패턴 추출 시스템은 Windows x64 OS에서 Eclipse IDE와 JDK 8, Oracle 10g 버전 데이터베이스를 기반으로 구현하였고, 실험 데이터는 서울시 도로 네트워크상에서 x_y 좌표로 임의의 노드(교차, 분기 및 목적지)들을 설정하여 측정된 택시 운행기록을 이동객체 ID, 노드의 x_y 좌표 값, 유효시간'으로 표시하여 사용하였고, 이동객체 1,000개에 대한 10시간의 위치데이터 중 일부를 사용하였다.

동일한 조건에서 출발지로부터 목적지를 임의로 변경하여 약 150~500회의 경로탐색을 위한 연산시간, 접근 노드 수 및 경로추출의 정확도를 기준으로 평균값을 산출하여 비교한 결과의 요약은 다음과 같다.(상세 결과는 향후 논문지에 발표 예정임)

- 실행시간: 제안 알고리즘은 동일구간에서 경험적 가중치를 적용한 A^* 알고리즘에 비하여 연산시간이 무시할 정도로 조금 길지만, 동심원 내에 포함된 모든 노드들을 대상으로 탐색을 수행하는 Dijkstra 알고리즘과는 큰 차이로 우수한 성능을 보였다.

- 접근노드 수: 제안 알고리즘은 빈발도가 작은 접근 대상 노드들의 수가 매 단계에서 줄어들기 때문에 다른 방법에 비하여 가장 적은 수의 노드 접근 성능을 보였다.

- 추출 정확도: 모든 탐색 대상노드들을 탐색하는 Dijkstra 알고리즘의 정확도를 100%로 설정하여 비교하면, A^* 알고리즘이 경험적 가중치에 의한 직진성에 의해 탐색 범위가 작으므로 최적경로 추출 확률이 낮아짐을 알 수 있고, 제안 알고리즘은 짧은 연산시간을 고려하면 최적경로 추출의 정확도는 다른 알고리즘들에 비해 우수한 정확도를 보였다.

IV. 결 론

본 논문은 FEC 환경에서 컴퓨팅 리소스의 최적 할당 및 지연시간 감소를 위한 계산 오프로딩의 효율성을 제고하기 위한 선행이며, 이동객체들의 시공

간 이동 이력데이터 집합으로부터 시공간 제약을 적용한 최적 이동패턴 추출 알고리즘을 제안하였다. 빈발도가 높은 패턴을 추출하여 최적경로를 탐색하는 제안 알고리즘은 A^* 및 Dijkstra 알고리즘들과 비교하여 상대적으로 빠른 연산시간과 적은 메모리를 사용하여 최적경로를 추출하는 장점이 있으며, 경험적 가중치를 적용한 A^* 알고리즘과의 비교 실험 결과를 통하여 이동경로 상의 사용자 선호도 가중치를 빈발도와 동시에 적용함으로써 최적경로의 정확도를 보다 향상시킬 수 있음을 도출하였다.

제안 알고리즘은 이동객체의 이동경로나 시공간적 동적 특성을 적용한 응용 서비스에 적용이 가능하므로, 이는 향후 FEC 환경에서 계산 오프로딩을 위한 컴퓨팅 리소스의 분배 및 배치에 유용하게 사용될 수 있다.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1 047768) and a grant (21R1TD-C161698-01) from Regional Innovation Technology Development Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

References

- [1] R. Buyya, S. Srirama (ED), Fog and Edge Computing: Principles and Paradigms, and Applications, 1st ed, John Wiley & Sons Inc., 2019.
- [2] Mach, P., Becvar, Z., "Mobile edge computing: A Survey on Architecture and Computation Offloading," IEEE Communication Survey Tutorial, 19, pp. 1628-1656, 2017.
- [3] Josilo, S., Dan, G., "Selfish Decentralized Computation Offloading for Mobile Cloud Computing in Dense Wireless Networks," IEEE Trans. Mobile Computing, 18, pp. 207-220, 2018.
- [4] C. You, K. Huang, H. Chae, B. Kim, "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," IEEE Trans. on Wireless Communications, 16(3), pp. 1397-1411, 2017.
- [5] Chen, M., Liang, B., Dong, M., "Multi-user Multi-task Offloading and Resource Allocation in Mobile Cloud Systems," IEEE Trans. Wireless Communication, 17, pp. 6790-6805, 2018.

- [6] Alameddine, H., et. al., “Dynamic Task Offloading and Scheduling for Low-latency IoT Services in Multi-access Edge Computing,” *IEEE Journal Sel. Areas Communication*, 37, pp. 668-682, 2019.
- [7] Al-khafajiy, M., et. al., “Improving Fog Computing Performance via Fog-2-Fog Collaboration,” *Future Generation Computing System*, 100, pp. 266-280, 2019.
- [8] Y. Lee, “Distributed Control Framework based on Mobile Agent Middleware,” *Journal of The KSCI*, 25(12), pp. 195-202, 2020.