

# 모바일 클라우드 컴퓨팅 환경에서 오프로드 방법 연구

변휘림\*, 한석현\*, 박부광\*, 송은하\*\*, 정영식\*

\*동국대학교 멀티미디어공학과

\*\*원광대학교 교양교육대학

e-mail : hazzly@dongguk.edu

## A Survey on Offload Methods in Mobile Cloud Computing Environments

HwiRim Byun\*, Seok-Hyeon Han\*, Boo-Kwang Park\*, Eun-Ha Song\*\*, Young-Sik Jeong\*

\*Dept. of Multimedia Engineering, Dongguk University

\*\*Dept. of Liberal Arts, Wonkwang University

### 요약

모바일 디바이스와 어플리케이션의 발전으로 Mobile Cloud Computing(MCC)의 필요성이 대두되고 있다. MCC는 다수의 모바일 디바이스를 하나의 클라우드 환경으로 통합해 배터리 수명, 스토리지, 프로세싱 등의 성능 향상을 가져온다. 그러나 아직 MCC의 오프로드 처리 방식이 명확히 정의되어 있지 않은 상황이다. 본 논문은 MCC 환경에서 모바일 디바이스의 작업을 클라우드에 이관해 처리하는 오프로드를 서버의 역할에 따라 다섯 가지로 나누어 정의해 제안한다. 마지막으로 더 효율적인 오프로드 방법을 제안하여 미래 연구 방향에 대해 논의한다.

### 1. 서론

스마트폰, 태블릿 PC, 스마트워치 등과 같은 모바일 디바이스는 현시대에 가장 빠르고 편한 통신도구로서 자리잡고 있다. 그러나 모바일 디바이스의 이동성을 위해 작아진 사이즈로 인해 제한된 성능은 서비스 품질의 향상을 방해한다. 따라서 그림 1과 같이 다수의 모바일 디바이스로 하나의 네트워크를 구성하여 스토리지, 프로세서 등의 리소스를 공유함으로써 성능을 높일 수 있는 Mobile Cloud Computing(MCC)에 대한 연구가 활발하다. MCC는 저렴한 비용으로 모바일 디바이스가 기존보다 높은 성능을 가질 수 있어 차세대 컴퓨팅 인프라로 부각되었다. MCC의 관리 및 자원 효율성 측면은 다양한 서비스가 제공되고 있으나 아직 오프로드 방식에 관한 명확한 분류가 정의되지 않은 상황이다.

본 논문은 모바일 디바이스가 외부 플랫폼으로 데이터를 이관하여 처리하는 오프로드 방식에 따른 다섯 가지 분류를 정의해 제안한다. 2 절에서는 기존 연구의 오프로드 방식에 따른 다섯 가지 분류를 정의한다. 3 절은 앞의 내용을 바탕으로 이상적인 오프로드 방식을 제안한다. 4 절은 최종적으로 결론 및 향후 연구에 대한 목차로 구성되어 있다.

### 2. 오프로드 방식에 따른 분류

오프로드는 MCC에서 주체가 되는 모바일 디바이스가 외부 플랫폼에 특정 컴퓨팅 작업의 전송을 의미한다. 모바일 디바이스 자체의 리소스보다 클라우드를 통한 리소스 제공자를 활용하여 작업을 수행함으로써 기존 리소스의 한계를 극복한다. 이를 통해 더 강력한 가상의 디바이스 성능을 제공할 수 있다. 본 논문에서 제안하는 분류 방법은 그림 2와 같다.

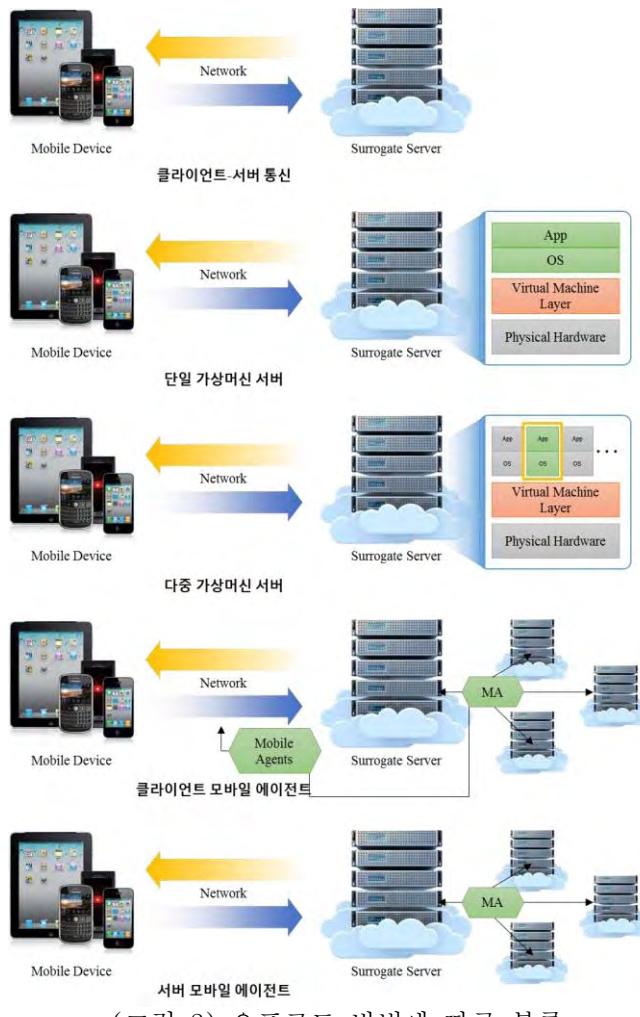


(그림 1) 오프로드를 통한 MCC

이 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2014R1A1A2053564). 또한 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학 ICT 연구센터육성 지원사업의 연구결과로 수행되었음" (IITP-2015-H8501-15-1014).

## 2-1. 클라이언트-서버 통신

모바일 디바이스가 네트워크를 통해 클라우드를 구성하고 있는 대리 서버 직접적으로 접속하는 방법이다. 클라이언트-서버의 직접적인 연결 관계를 뜻한다. 클라이언트와 서버가 공통적으로 탑재하고 있는 서비스에 대해서는 안정적이고 뛰어난 처리 능력을 가진다. 특히 서버에 동시에 연결되어 있는 클라이언트의 수가 적을수록 사용 가능한 리소스의 기대값이 높아진다. 그러나 단일 서버가 연결된 모든 모바일 디바이스에 리소스를 제공해야 하기 때문에 비용과 관리 측면에서 문제 가진다. 또한 이외의 서비스를 호출하기 위해서는 참여 디바이스에 필수적으로 서비스가 설치되어 있어야 한다. 이는 MCC의 이동성이라는 본질에 위배되는 문제를 가진다.



## 2-2. 가상머신 이관

서버의 물리적 하드웨어 계층 위에 OS 가상화를 통해 모바일 디바이스의 사용 환경을 구축한다. 클라이언트는 필요한 서비스를 OS에 직접 설치하여 사용할 수 있다. 이로 인해 서버는 클라이언트에 다양한 서비스 호출에 대응할 수 있게 된다. 서버간의 데이터 이관이 필요한 경우 시작 서버로부터 목적 서버에 가상머신의 메모리 이미지를 전송함으로써 OS 및 어플

리케이션의 무중단 서비스를 실현한다. OS를 사용하기 때문에 새로운 서비스를 제공하기 위해 추가적인 소스코드의 구성이 필요 없는 장점을 가진다. 또한 가상머신의 바운더리에 따른 보안의 우수성을 가진다. 그러나 VM의 통합시간만큼의 지연시간이 발생할 수 있다.

가상머신 이관은 클라이언트와 가상화 OS의 매칭 방식에 의해 두 가지로 분류될 수 있다. 가상화 OS와 클라이언트가 일대다수로 매칭되는 경우 단일 가상머신 서버, 일대일로 매칭되는 경우 다중 가상머신 서버로 분류한다.

### 2-2-1. 단일 가상머신 서버

서버가 하나의 가상화된 OS를 가지는 경우로 모든 클라이언트가 동일한 가상화 OS에 접속한다. 이 경우 클라이언트들은 서로의 작업에 관여할 수 있어 가상머신 바운더리에 의한 보안강화 효과를 기대할 수 없다. 그러나 여러 클라이언트가 동일 작업을 분산 처리함에 있어서 유연한 클라이언트의 추가 및 제거가 가능하다.

### 2-2-2. 다중 가상머신 서버

서버가 여러 개의 가상화된 OS를 가지고 모든 클라이언트는 개별적인 가상화 OS와 일대일 매칭된다. 각각의 가상머신은 독립되어 실행되기 때문에 클라이언트는 가상머신 바운더리에 의해 타 클라이언트의 작업에 접근할 수 없어 보안이 우수하다. 다중 가상머신 서버는 가상머신 오버레이 호환성 문제를 안고 있어 단일 서버가 많은 수의 클라이언트와 대응하기 어렵다.

## 2-3. 모바일 에이전트

서버에서 실행되는 독립적인 프로그램인 모바일 에이전트에 의해 클라이언트가 필요로 하는 리소스를 수집한다. 모바일 에이전트는 서버로부터 또 다른 연결되어 있는 서버에 전송된다. 서버는 자신의 리소스를 필요로 하는 에이전트를 인식하고 이를 제공한다. 실시간 동적 배치에 능하다. 또한 모바일 에이전트는 서버에서 독자적으로 실행되기 때문에 모바일 디바이스와 같이 네트워크 연결의 중단 가능성에 높은 환경에 적합하다. 그러나 서비스 자체에 대한 정보를 가지고 있는 에이전트가 많은 서버를 접속하기 때문에 보안에 취약성을 안고 있다.

모바일 에이전트는 클라이언트로부터 서버에 전송되어 프로세스를 수행하는 클라이언트 모바일 에이전트와 서버에서 자체적으로 수행되어 리소스를 필요로 하는 클라이언트에 즉각적인 응답을 하는 서버 모바일 에이전트로 분류할 수 있다.

### 2-3-1. 클라이언트 모바일 에이전트

스크립트 언어로 작성된 프로그램은 모바일 디바이스로부터 서버에 오프로드 되어 실행된다. 데이터가 전송되는 기준 통신과는 다르게 프로그램 자체가 전송된다. 서버는 클라이언트가 필요로 하는 리소스를

수집하여 제공하게 된다. 서버와 클라이언트가 동일한 서비스를 수행하는 어플리케이션을 가지고 있을 필요가 없어 클라이언트의 요청에 유연하게 대응할 수 있다.

### 2-3-2. 서버 모바일 에이전트

모바일 에이전트는 클라이언트가 요청하기 전에 서버 자체에서 리소스 수집을 수행한다. 클라이언트가 요청할 경우 미리 수집된 리소스를 바로 제공해줄 수 있다. 에이전트가 항상 작동하고 있기 때문에 동적 배치가 가능하지만 클라이언트가 요청한 리소스가 기존 서버에서 작동하던 모바일 에이전트와 다를 경우 리소스를 낭비하게 되는 단점이 있다.

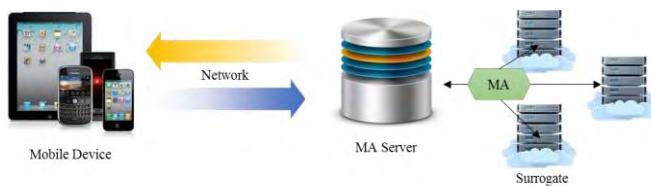
<표 1> MCC 환경에 따른 오프로드 방법의 비교

오프로드 종류	장점	단점
클라이언트-서버 통신	단일 서비스 처리에 가장 효과적임.	다양한 모바일 어플리케이션 환경에 유연한 대처가 어려움.
단일 가상머신 서버	단일 프로세스 협업에 유리함.	개인정보와 같은 데이터 처리에 이슈를 가짐.
다중 가상머신 서버	개인정보와 같은 데이터 처리에 유리함.	간헐적인 접속형태를 가진 모바일 클라우드에 비효율적임.
클라이언트 모바일 에이전트	불안정한 네트워크 환경에서도 사용 가능함.	에이전트를 전송을 위한 네트워크 데이터의 사용이 필요함.
서버 모바일 에이전트	빠른 처리속도를 확보할 수 있음.	서버가 제공하는 모바일 에이전트의 서비스에 한정됨.

MCC 환경은 네트워크의 불안정함을 충분히 고려한 오프로드 방식을 사용하여야 한다. 그러나 모바일 에이전트 모델을 제외하곤 이에 대한 고려가 되어있지 않으며, 모바일 에이전트 또한 많은 네트워크 대역폭을 사용해야 하는 단점을 가진다.

### 3. 제안하는 오프로드 모델

본 논문에서는 대용량 전송을 배제하고 간헐적으로 중단되는 네트워크 환경을 위한 새로운 오프로드 모델 Mobile Agent Oriented Server(MAOS)를 제안한다.



(그림 3) 제안하는 MAOS 모델

모바일 디바이스는 사용하고자 하는 서비스를 위한 모바일 에이전트를 선정한다. 선정한 모바일 에이전

트와 처리하고자 하는 데이터를 다양한 서비스에 대한 에이전트를 탑재하고 있는 모바일 에이전트 서버로 전송한다. 모바일 에이전트 서버는 클라이언트로부터 요청 받은 에이전트를 연결된 서버에 전송해 데이터를 처리한다. Software as a Service의 형태로 그림 3과 같이 나타낼 수 있다.

### 4. 결론 및 향후 연구

본 논문은 기존 연구들로부터 다섯 가지 오프로드 모델의 분류를 제안하였다. 또한 다섯 가지 분류에 대해 MCC 환경에 적용 시 어떠한 특징을 가질 수 있는지 분석해 보았다. 본 연구를 바탕으로 MCC 환경에 더욱 적합한 오프로드 모델 MAOS를 제시할 수 있었다. MAOS는 MCC 환경의 작은 네트워크 대역폭과 간헐적인 중단이 있어도 사용이 가능하다. 본 논문을 바탕으로 추후 MAOS 모델에 대한 연구를 지속해 가고자 한다.

### 참고문헌

- [1] Niroshinie Fernando, Seng W. Loke, Wenny Rahayu, "Mobile cloud computing: A survey," Future Generation Computer Systems, vol. 29, issue. 1, pp. 84-106, Jan. 2013.
- [2] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches," Wireless Communications and Mobile Computing, vol. 13, issue. 18, pp. 1587-1611, Dec. 2013.
- [3] K. Kumar, Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?", IEEE Comput., vol. 43, no. 4, pp. 51-56, Apr. 2010.
- [4] M. V. Barbera, S. Kosta, A. Mei, J. Stefa, "To offload or not to offload? The bandwidth and energy costs of mobile cloud computing," in Proceedings IEEE Infocom (INFOCOM '13), pp. 1285-1293, Turin, Italy, 14-19, Apr. 2013.
- [5] 박민균, "An Efficient Offloading Framework for Android Platform : 안드로이드 플랫폼을 위한 효율적인 오프로딩 프레임워크," 숭실대학교 석사학위논문, Feb. 2014.
- [6] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, Ashwin Patti, "CloneCloud: elastic execution between mobile device and cloud," Proceedings of the sixth conference on Computer systems (EuroSys '11), pp. 301-314, Salzburg, Austria, 10-13, Apr. 2011.
- [7] Karthik Kumar, Yung-Hsiang Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?", Computer, vol. 43, issue. 4, pp. 51-56, Apr. 2010.
- [8] Yuan Zhang, Hao Liu, Lei Jiao, Xiaoming Fu, "To offload or not to offload: An efficient code partition algorithm for mobile cloud computing," IEEE 1st International Conference on Cloud Networking (CLOUDNET '12), pp. 80-86, Paris, France, 28-30, Nov. 2012.