

# 에이전트 기반 모델링 및 시뮬레이션을 위한 컴퓨팅 인프라 기술 동향

Technical Trends of Computing Infrastructure for  
Agent Based Modeling & Simulation

정영우 (Y.W. Jung, jungyw@etri.re.kr)	클라우드컴퓨팅연구그룹 책임연구원
손석호 (S. Son, shsonkorea@etri.re.kr)	클라우드컴퓨팅연구그룹 선임연구원
오병택 (B.T. Oh, botoh@etri.re.kr)	클라우드컴퓨팅연구그룹 책임연구원
이규철 (G.C. Lee, gyuichel@etri.re.kr)	클라우드컴퓨팅연구그룹 책임연구원
배승조 (S.J. Bae, sbae@etri.re.kr)	클라우드컴퓨팅연구그룹 책임연구원
김병섭 (B.S. Kim, powerkim@etri.re.kr)	클라우드컴퓨팅연구그룹 책임연구원
강동재 (D.J. Kang, djkang@etri.re.kr)	클라우드컴퓨팅연구그룹 책임연구원/그룹장
정영준 (Y.J. Jung, jjing@etri.re.kr)	임베디드시스템연구그룹 책임연구원/그룹장

- I. 서론
- II. 대규모 ABMS 기술 동향
- III. 대규모 ABMS 컴퓨팅  
인프라 연구 동향
- IV. 클라우드 기반 ABMS  
시스템 국내 사례
- V. 결론

Agent-based modeling and simulation (ABMS) is a computational method for analyzing research targets through observations of agent-to-agent interactions, and can be applied to multidimensional policy experiments in various fields of social sciences to support policy and decision making. Recently, according to increasing complexity of society and the rapid growth of collected data, the need for high-speed processing is considered to be more important in this field. For this reason, in the ABMS research field, a scalable and large-scale computing infrastructure is becoming an essential element, and cloud computing has been considered a promising infrastructure of ABMS. This paper surveys the technology trends of ABMS tools, cloud computing-based modeling, and simulation studies, and forecasts the use of cloud-computing infrastructure for future modeling and simulation tools. Although fundamental studies are underway to apply and operate cloud computing in the areas of modeling and simulation, new and additional studies are required to devise an optimal cloud computing infrastructure to satisfy the needs of large-scale ABMS..

\* DOI: 10.22648/ETRI.2018.J.330512

\* 본 연구는 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행되었음[No.2018-0-00225, 과학적 정책 수립을 위한 도시행정 디지털트윈 핵심 기술 개발].



본 저작물은 공공누리 제4유형  
출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

## I. 서론

에이전트는 개별적이고 자율적인 개체로 현실과 유사한 가상 현실 객체로 정의되며, 하나 이상의 다른 에이전트들과 상호작용을 한다. 에이전트 기반 모델(Agent Based Model)은 에이전트와 에이전트 간 상호작용을 정의하여, 특정 현상을 시뮬레이션할 수 있게 해주는 계산 모델의 한 형태를 말한다[1], [2]. 이러한 에이전트 기반 모델은 복수의 에이전트들 간의 세밀한 행동과 상호작용을 분석하고 사회 현상을 분석할 수 있다[3]. 이와 같이, 에이전트 기반 모델링 및 시뮬레이션(ABMS: Agent-Based Modeling & Simulation)과 같은 마이크로 시뮬레이션 방법은 특히 다양한 사회과학 분야에서 다차원적 정책실험을 수행할 수 있으며, 이를 통해 필요한 정책수립 및 의사결정을 가능하게 한다.

여러 ABMS 도구는 데스크톱에서 슈퍼컴퓨터에 이르기까지 다양한 규모의 컴퓨팅 인프라에서 수행될 수 있지만, 사회가 보다 복잡해지고 수집되는 데이터의 기하급수적 증가에 따라 점차 대규모 컴퓨팅 인프라를 요구하고 있다. 이를 위하여 ABMS 연구자들은 기존 시뮬레이션 도구를 분산 컴퓨팅 클러스터에서 동작시킬 수 있도록 발전시켰으나, 고비용의 컴퓨팅 인프라 및 자원의 확장성이라는 한계에 직면하였다.

이러한 상황에서 대규모의 컴퓨팅 자원을 용도에 따른 다양한 성능으로 제공할 수 있는 클라우드 컴퓨팅 기술은 자원 운영의 유연성, 효율성, 확장성 등의 장점을 제공함으로써 대규모 분산 시뮬레이션을 위해 적합한 컴퓨팅 인프라로 인식되고 있다.

본고에서는 복잡한 사회현상 분석을 위한 대규모 ABMS 도구의 기술 동향 및 클라우드 컴퓨팅 기반의 대규모 고성능 시뮬레이션 기술 연구 동향을 살펴봄으로써 향후 ABMS를 위한 클라우드 컴퓨팅의 활용 전망을 분석하고자 한다.

## II. 대규모 ABMS 기술 동향

### 1. 기술 개요

지난 수십 년 간 개발된 다양한 ABMS 도구는 에이전트를 모델링하기 위한 특정 프로그래밍 방법을 제공하며, 일반성, 유용성, 수정 가능성, 확장성, 성능 등과 관련하여 서로 다른 특징을 가지고 있다[4]. 본 절에서는 사회과학 분야에 주로 적용되는 범용 ABMS 도구 및 대규모 병렬 분산 컴퓨팅 플랫폼을 지원하는 ABMS 도구를 살펴보고자 한다.

ABMS 도구는 사회과학, 자연과학, 환경, 야생생물, 헬스케어, 비즈니스, 재무, 운영 및 관리 등과 같이 다양한 분야에 활용될 수 있다. 본 고에서 다루고자 하는 사회과학 분야에 적용 가능한 ABMS 도구는 그간 수십여 종이 개발되었으며, 이 중에서 대규모 분산 병렬 컴퓨팅 클러스터 및 클라우드 컴퓨팅 환경에 적용 가능한 ABMS 도구도 다수 존재한다. 예로, FLAME[5], HLA\_Agent[6], HLA\_RePast[7], D-MASON[8], MAT-Sim[9], Pandora[10], PDES-MAS[11], Repast HPC[12], Swarm[13] 등이 해당한다.

### 2. 대규모 분산병렬 컴퓨팅 지원 ABMS 도구

ABMS에서 시뮬레이션의 복잡도가 높아질수록 많은 수의 에이전트가 필요하며, 대규모 에이전트의 연산과 에이전트 간 상호작용을 수행하기 위해서는 높은 수준의 컴퓨팅 자원이 요구된다. 이러한 이유로, ABMS 시스템 연구자들은 시뮬레이션 시스템을 분산병렬 컴퓨팅 클러스터에서 운영하기 위하여 시뮬레이션 엔진을 발전시켜왔다. 본 절에서는 분산병렬 컴퓨팅 환경을 대상으로 개발된 대표적인 분산 시뮬레이션 엔진인 Repast HPC, FLAME, D-MASON에 대해 상세히 살펴본다.

#### 가. Repast HPC

Repast는 시카고 대학에서 개발이 시작되어, 현재는

아르곤 국립 연구소에서 기능 확장과 유지보수를 수행하고 있다. 다양한 Repast 도구 중 범용적인 ABMS 개발 목적의 Repast Symphony와 고성능 시스템을 위한 Repast HPC의 2가지 도구가 현재 지속적으로 확장 및 유지보수 되고 있다.

Repast Symphony는 자바 기반으로 워크스테이션과 소규모 컴퓨팅 클러스터 상에서 사용하는 것을 목적으로 개발되었으며, Repast HPC는 C++ 기반 모델링 시스템으로, 컴퓨팅 클러스터와 슈퍼컴퓨터 등을 이용한 대규모 분산 컴퓨팅 플랫폼을 위한 차세대 ABMS 도구라고 할 수 있다[14]. 범용적인 시뮬레이션 도구인 Repast Symphony의 핵심 구성은 agent class, context & projection, discrete event scheduler, data set, model class 등으로 이루어진다.

Repast HPC는 분산 시뮬레이션 환경에 적용할 수 있도록 Repast Symphony의 핵심 구성요소들을 확장하였으며, 분산 병렬 처리를 위해 Repast Process, Parallel Scheduling, SharedContext, Parallel Projections, Synchronizing Projections, Data Collection, 그리고, Random Number Generation 등의 추가 컴포넌트를 제공한다.

Repast HPC도 Repast Symphony와 동일한 스케줄링 방법을 지원하지만, 시뮬레이션의 병렬 처리를 위해서 분산 이벤트 스케줄러를 동시에 제공하며, 모든 스케줄러는 동일 클럭(clock) 기반의 틱(tick) 수행을 통해 엄격하게 동기화된다[15].

#### 나. FLAME

FLAME은 영국 Sheffield 대학의 Simon Coakley의 주도로 개발되었으며 단일 시뮬레이션 환경에서 다양한 범위의 에이전트 모델을 지원하기 위한 시뮬레이터이다. 이후 영국 STFC RAL과 협력하여 MPI와 OpenMP를 이용한 병렬처리 기술이 적용된 HPC 시스템상에서

대규모 에이전트 시뮬레이션을 지원하도록 확장 개발되었으며, Sheffield 대학의 Paul Richmond에 의해 대규모 에이전트로 구성된 복잡계 시스템을 3D로 시뮬레이션하기 위해 CUDA 기반의 FLAME GPU로도 발전되었다. 또한, FLAME은 조직 배양이나 신호 전달 등의 의학과 생물학에 주로 사용되었으며, 경제 모형을 통한 금융 모델링을 연구하는 EURACE 프로젝트에 활용되기도 하였다[16].

FLAME은 템플릿 기반의 분산 시뮬레이션 코드 생성기로써, 데스크톱에서 슈퍼컴퓨터까지 다양한 컴퓨팅 시스템에서 실행될 수 있는 에이전트 기반의 응용 프로그램을 생성한다. 또한, 개발자들이 서로 호환되는 모델이나 소프트웨어 도구 개발에 사용할 수 있도록 공식 프레임워크 명세를 제공하고, 명세를 준수하여 개발된 모델은 기존 또는 새로운 시뮬레이션에 쉽게 통합될 수 있도록 지원한다. 또한, MPI 등의 병렬화 기술을 사용하면 대규모 멀티 프로세서 시뮬레이션을 개발할 수 있다[17].

FLAME의 병렬 수행을 위해서는 에이전트 간 통신 기술이 사용되며, 에이전트들의 메시지 핸들링 및 에이전트 간 효율적 상호작용이 가능하도록 지원한다[18].

Repast HPC와 D-Mason이 객체 개념의 에이전트를 정의하는 것과는 달리, FLAME은 X-Machine이라는 오토마타 개념의 에이전트를 사용한다[19]. 또한, Repast HPC가 동적 로드 밸런싱(Dynamic load balancing) 기술을 사용하는 반면, FLAME은 정적 로드 밸런싱(Static load balancing) 기술을 사용한다[19].

#### 다. D-MASON

D-MASON은 Salerno 대학에서 개발되었으며, MASON[20]의 분산형 버전으로 사용자가 기존 시뮬레이션 모델을 재개발할 필요 없이, 에이전트 수의 한계를 극복할 수 있는 도구를 제공한다. D-MASON은 에이전트 간 통신을 위하여 자바 메시지 서비스(ActiveMQ

JMS) 기술을 제공하며, 자바 언어를 사용하여 에이전트 모델을 구현하였다[19].

MASON은 본래 1,000개의 에이전트를 가진 모델에서도 실행 속도를 극대화한다는 목표를 가지고 설계되었으며, 개발자가 자바의 객체 지향 프로그래밍 패러다임을 활용하고, MVC(Model-View-Controller) 소프트웨어 디자인 패턴을 사용하여 복잡하고 구조화된 모델링을 할 수 있도록 고안하였다. 이러한 MASON이 단일 쓰레드 애플리케이션을 대상으로 하고 있는 반면, MASON의 분산 노드 버전인 D-MASON은 수백만 개의 에이전트가 포함된 모델을 분산 시뮬레이션함으로써 단일 쓰레드 버전에 비해 향상된 성능을 제공한다[21]. D-MASON 시스템은 마스터 노드와 워커 노드, 그리고 통신 서버라는 세 종류의 노드로 구성된다.

MASON의 에이전트 분할 방법이 가용한 워커 노드에 고정된 수의 에이전트들을 할당하는 반면, D-MASON은 공간 분할 방법을 채택하여 각 에이전트의 공간적 위치에 따라 영역을 구분하여 각 워커 노드에 할당함으로써 에이전트 간 통신에 따른 오버헤드를 최소화하고, 인접 지역 간 동기화와 지역 간 에이전트 마이그레이션을 지원한다.

### III. 대규모 ABMS 컴퓨팅 인프라 연구 동향

#### 1. ABMS 컴퓨팅 인프라 발전 방향

컴퓨팅 플랫폼은 분산 컴퓨팅, 병렬 컴퓨팅, 유틸리티 컴퓨팅, 그리드 컴퓨팅, 고성능 컴퓨팅, 클라우드 컴퓨팅 등으로 다양한 형태로 진화되어 왔다. 특히, 클라우드 컴퓨팅은 사용자에게 컴퓨팅 서비스 사용에 편의성 및 민첩성을 제공하며, 컴퓨팅 제공자에게는 자원 운영의 유연성, 효율성, 확장성을 제공하는 등 많은 장점으로 현대 컴퓨팅 패러다임의 주축이 되어 왔다.

한편, ABMS 연구 분야에서는 시뮬레이션 엔진을 새로운 컴퓨팅 패러다임에 적용하는 연구들이 진행되어

〈표 1〉 시뮬레이션 엔진의 발전 방향

단일 노드 시뮬레이션 엔진	분산 노드 시뮬레이션 엔진
MASON	D-MASON
FLAME	FLAME
Repast Symphony	Repast HPC

왔다. 기존의 시뮬레이션은 계산량이나 메모리 사용량이 높지 않았기 때문에 단일 컴퓨팅 노드에서 수행이 가능하였다. 하지만, 사회의 다양성이 커짐에 따라 시뮬레이션 복잡도는 지속적으로 상승하였으며, 단일 컴퓨팅 노드의 자원으로 시뮬레이션을 수행하는 데 있어서 한계상황에 직면하였다. 예를 들어, 시뮬레이션이 단일 컴퓨팅 노드에서 수행되는 경우, 메모리 부족으로 인해 시뮬레이션을 끝까지 수행할 수 없는 경우가 발생하고, 장시간 수행 중인 시뮬레이션이 오류로 인해서 중단되면 시뮬레이션 수행자들의 시간과 노력이 무의미하게 소비될 수도 있다.

특히, ABMS에서 의미 있는 결론을 도출하기 위해서는 대규모 에이전트와 에이전트 간 빈번한 상호작용을 포함하는 시뮬레이션이 필수적이며, 이를 위해서는 높은 수준의 CPU, 메모리 등의 컴퓨팅 자원이 요구된다. 이로 인해, 시뮬레이션 엔진 연구자들은 시뮬레이션을 분산 컴퓨팅 클러스터에서 수행할 수 있도록 발전시켰으며, 분산 시뮬레이션 엔진을 개발한 대표적인 사례는 〈표 1〉과 같다.

다만, 시뮬레이션 수행자 입장에서는 HPC와 같은 분산 컴퓨팅 환경은 관리가 어려운 환경일 뿐만 아니라, 구축 및 관리 비용이 높다[22]. 또한 대규모 에이전트 시뮬레이션 수행 중에 자원이 부족한 경우, 민첩하게 자원을 공급하기가 어려우며, 시뮬레이션에 필요한 다양한 소프트웨어 도구를 설치, 운용, 관리하는 데 발생하는 노력도 사용자에게 큰 부담이 될 수 있다[22].

#### 2. 클라우드 컴퓨팅 기반 ABMS 연구

클라우드 컴퓨팅은 사용자가 원하는 성능 및 규모의

컴퓨팅 클러스터를 매우 민첩하게 동적으로 구성할 수 있으며, 컴퓨팅 자원, 소프트웨어, 실행 환경 등을 서비스 형태로 제공할 수 있기 때문에 컴퓨팅 환경 구성에 익숙하지 않은 시뮬레이션 수행자들도 쉽게 필요한 인프라를 구성할 수 있다.

최근 많은 연구자들이 대규모 분산병렬 ABMS를 위한 클라우드 컴퓨팅 인프라의 필요성을 주장하였으며, 시뮬레이션 엔진을 클라우드 컴퓨팅 환경에 접목하기 위한 기초적인 연구들을 수행하였다.

### 가. 클라우드 기반 ABMS 기초 연구

[22]는 Pandora 도구를 활용한 분산 ABMS를 위해 바르셀로나 슈퍼컴퓨터의 슈퍼컴퓨터 노드 9대로 구성된 환경과 퍼블릭 클라우드(Amazon EC2)의 가상시스템 인스턴스(m1.large, c1.4xlarge) 9대로 구성된 환경을 비교하여 성능을 평가하였다.

성능 평가를 위한 시뮬레이션은 컴퓨팅 중심 모델과 상호작용 중심 모델로 나누어 수행되었으며, 컴퓨팅 중심 모델은 100개의 에이전트가 에이전트 간 상호작용 없이 매 단계마다 Markov Decision Process를 수행하여 컴퓨팅 연산이 많이 발생하도록 하였다. 반면, 상호작용 중심 모델은 400개 에이전트 간의 상호작용 횟수를 20배 높이고, Markov Decision Process는 수행하지 않도록 설정하였다.

벤치마킹의 수행 결과, 컴퓨팅 중심 모델에서는 슈퍼컴과 퍼블릭 클라우드의 성능이 유사하였으며, 상호작용 중심 모델에서는 슈퍼컴의 성능이 더 높음을 보였다. 퍼블릭 클라우드에서의 시뮬레이션 성능 저하의 이유를 가상 시스템 인스턴스 간의 낮은 상호연결 속도로 분석하였다. 다만, 클라우드 컴퓨팅 자원의 비용이 저렴하기 때문에 벤치마킹에서 보인 수준의 성능 저하는 충분히 수용 가능한 것으로 결론지었다. 그러나, 에이전트 규모(100개 또는 400개)와 컴퓨팅 자원의 규모가 한정적이

었기 때문에 대규모 분산 시뮬레이션에서는 성능 저하가 더 크게 발현될 수도 있을 것으로 보인다.

[23]에서는 분산 ABMS 도구인 D-MASON을 클라우드에서 실행할 수 있도록, D-MASON as a Service 구조를 제안하였다. D-MASON as a Service 아키텍처는 클라우드 컴퓨팅 환경을 단순하게 Amazon EC2를 통해 구성하고, D-MASON의 시뮬레이션 워크로드를 Star-Cluster 플러그인 등의 클러스터링 도구를 활용하여 분산하였다. D-MASON as a Service 사용자는 클라우드 컴퓨팅 인프라에 대한 고민 없이 시뮬레이션을 서비스 형태로 수행할 수 있다. 이를 통해, 온프레미스 HPC 클러스터와 Amazon EC2 클라우드를 대상으로 시뮬레이션 성능을 비교 분석하였고, 그 결과 클라우드에서 보다 만족스러운 가격대 성능비를 확인하였다.

[23]과 유사한 연구로서, [24]에서는 분산 ABMS 도구인 FLAME을 클라우드에 접목하고 성능을 평가하기 위하여, 동일한 시뮬레이션을 64비트 윈도우 노트북, 듀얼 코어 노트북, HPC grid-serial, HPC grid-2core, Amazon EC2 등의 서로 다른 컴퓨팅 플랫폼에서 수행하였다. 그 결과 HPC grid와 Amazon EC2는 유사한 성능을 보였다. 비교적 단순한 시뮬레이션 수행이기 때문에 일반적인 성능 평가 결과로 판단하기에는 어려움이 있지만, HPC와 비교하여 클라우드의 성능 문제가 없다면, 클라우드 컴퓨팅은 동적 확장성, 자원 활용의 유연성 등의 관점에서 매우 활용 가치가 크다고 판단된다.

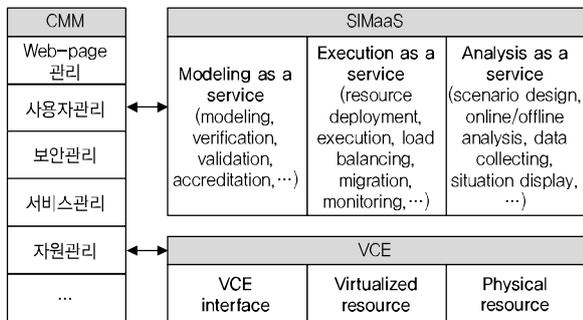
또한, [24]에서는 ABMaaS 구조를 제시하였는데, 클라우드에서 제공되는 가상 시스템에 시뮬레이션을 위한 에이전트 테스트를 동적으로 분배하고, 가상 시스템의 SLA에 맞는 에이전트 테스트를 할당하고, 자원 모니터링 및 장애극복 기능을 제공하는 등 클라우드의 장점을 일부 활용하고 있다.

[25]는 클라우드 기반의 PDES 구조를 제안하고, 기존의 PDES 엔진을 클라우드 인프라에 효율적으로 접목하

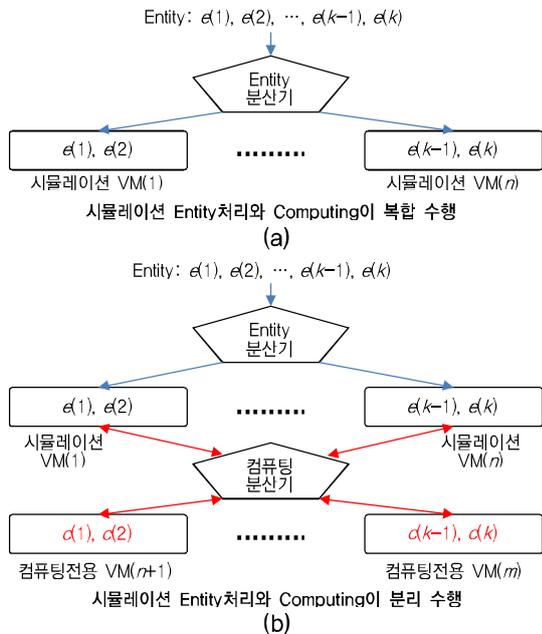
기 위한 새로운 PDES 엔진으로 CSim(Cloud-based computer Simulation)을 제안하였다.

CSim 구조에서 클라우드 관리 모듈(CMM)은 웹페이지, 사용자, 보안, 서비스, 자원 등의 관리 모듈로 구성되어 있으며, 가상 컴퓨팅 환경(VCE)은 클라우드 인프라 자원들로 구성되어 있다(그림 1) 참조]. SIMaaS는 모델링 및 시뮬레이션을 위한 워크플로우에 해당하는 세부 기능들을 서비스 형태로 제공한다[25].

CSim은 기존의 PDES 엔진을 클라우드 인프라에 효율적으로 접목하기 위하여 1) 시뮬레이션 실행 방식 2)



(그림 1) CSim 구조



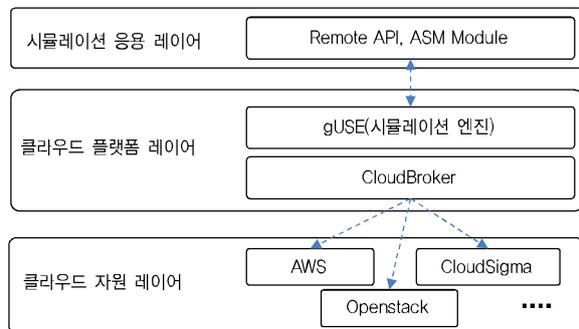
(그림 2) CSim의 시뮬레이션 실행 방식: (a) 전통적 모델, (b) 제안된 모델

시뮬레이션 모델 구성 3) 자원의 수평적 확장성 지원 4) 스케줄링 및 자원 할당 방식 등을 위한 새로운 방법을 반영하였다. 특히, 기존 시뮬레이션 수행 방식이 엔티티(에이전트) 객체와 엔티티가 수행하는 계산이 분리되지 않고 단일 노드에서 이루어지는 반면, 제안된 모델에서는 엔티티 객체를 위한 노드와 엔티티의 계산 수행을 위한 노드를 분리하여 구성할 수 있다(그림 2) 참조]. 또한, 수행되는 계산량에 맞게 자원 규모를 높이거나 낮추는 등의 자원의 유연한 확장을 지원한다[26].

마지막으로 시뮬레이션 워크로드의 스케줄링 방식에 따라 성능의 차이가 발생하는 것을 실험을 통해 확인하였고, 이는 향후 더욱 효과적인 스케줄링 방식이 연구되어야 함을 시사하였다.

#### 나. ABMS를 위한 멀티 클라우드 활용

[26]은 클라우드 기반 시뮬레이션 서비스 플랫폼인 CloudSME(Cloud-based Simulation platform for Manufacturing and Engineering)를 제안하였다(그림 3) 참조]. CloudSME도 클라우드 기반의 MSaaS를 제공하며, 이종의 멀티 클라우드 컴퓨팅 플랫폼을 중개할 수 있는 CloudBroker를 통해 Amazon EC2, Openstack, Eucalyptus, OpenNebula, IBM SmartCloud 등의 다양한 클라우드 플랫폼에서 시뮬레이션을 수행할 수 있는 특징을 가지고 있다. CloudSME는 다양한 시뮬레이션 엔진을 사용할 수 있는 프레임워크를 제공하며, [26]에



(그림 3) CloudSME 시뮬레이션 플랫폼

서는 단일 ABMS 도구인 Repast Symphony가 클라우드에서 효과적으로 동작하는 것을 확인하였다.

[27]은 다양한 클라우드의 배치 모델(퍼블릭 클라우드, 인터클라우드, 프라이빗 클라우드, 파트너 클라우드 등)에 따라 MSaaS가 클라우드에 배치 및 연합되는 방식을 다음과 같이 4가지로 분류하였다.

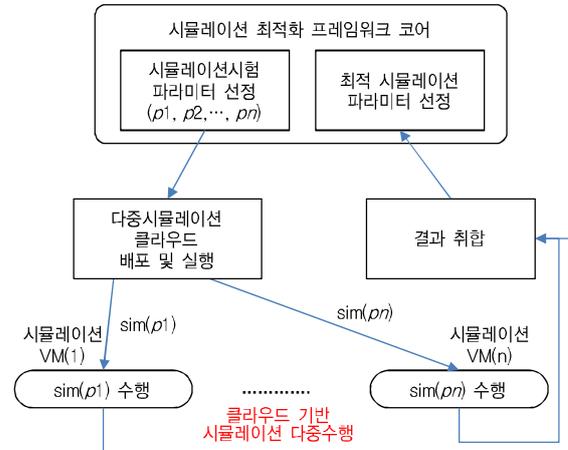
- 독립적인 어플리케이션들이 동일 데이터센터에 위치하여 상호 연동되는 시뮬레이션 연합.
- MSaaS 형태로 구축된 서비스들이 동일 데이터센터 내에서 상호 연동되는 연합.
- 독립적인 어플리케이션들이 서로 다른 데이터센터에 위치하여 상호 연동되는 시뮬레이션 연합.
- MSaaS 형태로 구축된 서비스들이 서로 다른 데이터센터 내에서 상호 연동되는 연합.

이와 같이, [27]은 클라우드의 배치 모델에 따라 분산 시뮬레이션 엔진의 구조 변화가 필요하며, 이를 위한 클라우드 기반의 시뮬레이션 모델 및 상호 운용방식의 연구 필요성을 강조하였다.

#### 다. 시뮬레이션 최적화를 위한 클라우드 자원 확장성 활용

시뮬레이션에 사용되는 다양한 파라미터들은 현실을 반영할 수 있도록 설정되어야 하기 때문에 시뮬레이션 수행 결과를 보고 파라미터가 현실을 반영하는지를 판단하여야 한다. 만약 그 결과가 현실을 반영하지 않는다면 파라미터를 변경하여야 하며, 이 과정을 시뮬레이션 최적화를 위한 파라미터 공간 탐색(Parameter space exploration, Parameter sweep)이라고 한다.

[28]은 시뮬레이션 최적화의 자동화 및 병렬화를 위해서 클라우드 컴퓨팅의 자원 확장성을 활용하였으며, 제한한 시뮬레이션 최적화를 위한 프레임워크 코어는 시뮬레이션을 위한 다양한 파라미터 세트를 생성하고, 각



(그림 4) 시뮬레이션 최적화 프레임워크

파라미터 세트를 사용하는 복수의 시뮬레이션을 동시 수행한다. 이때, 클라우드는 필요한 시뮬레이션 횟수에 맞게 가상시스템을 동적으로 준비하고, 각 시뮬레이션은 가상시스템에 할당되어 병렬적으로 수행된다. 모든 시뮬레이션의 결과들은 빅데이터 플랫폼을 통해서 취합되며, 각 파라미터 그룹을 평가하여 시뮬레이션 최적화를 결정한다(그림 4 참조). 시뮬레이션 최적화 프레임워크는 다양한 시뮬레이션 엔진을 지원하며, 단일 노드 시뮬레이션 엔진인 NetLogo[29]나 MASON 등을 사례로 활용하였다.

#### IV. 클라우드 기반 ABMS 시스템 국내 사례

본 장에서는 클라우드 기반 ABMS 시스템의 국내 사례를 통해 향후 클라우드 기반 ABMS의 주요 기술 및 전망을 살펴본다.

한국전자통신연구원에서는 2018년부터 미래 스마트 시티 구현의 정책 기초 일환으로, 저출산·고령화에 따른 지역 소멸 등의 도시문제 해결을 위해 도시가 직면한 다양한 문제를 분석하고 미래변화 모습의 이해를 통한 근원적인 해결을 위한 정책 수립을 지원하고자 클라우드 기반의 '과학적 정책 수립을 위한 도시행정 디지털트윈 핵심 기술 개발' 사업을 진행 중에 있다.

사람의 직관에 의한 정책 수립은 한계가 있어 ICT 기술의 접목이 필요하며, 초지능·초연결 사회 가속화로 도시의 공간적, 사회적 구성요인 간 상호작용이 복잡해짐에 따라, 기존의 개별적인 도시 정책 수립 방법론은 한계에 이르러 도시 경제사회의 객체 간 유기적 상호작용을 종합적으로 고찰할 수 있는 기술 확보가 필수적이다. 더욱이, 기존 도시는 교통, 환경, 에너지 등 개별 인프라 중심의 운영 시스템을 구축해 왔지만 복잡해지는 사회 구조에 따라 향후에는 도시의 구성요소, 영향요인, 사회현상 등 유기적 상호작용을 기반으로 미시적·포괄적 정책 수립 지원 기술 개발이 필요하다.

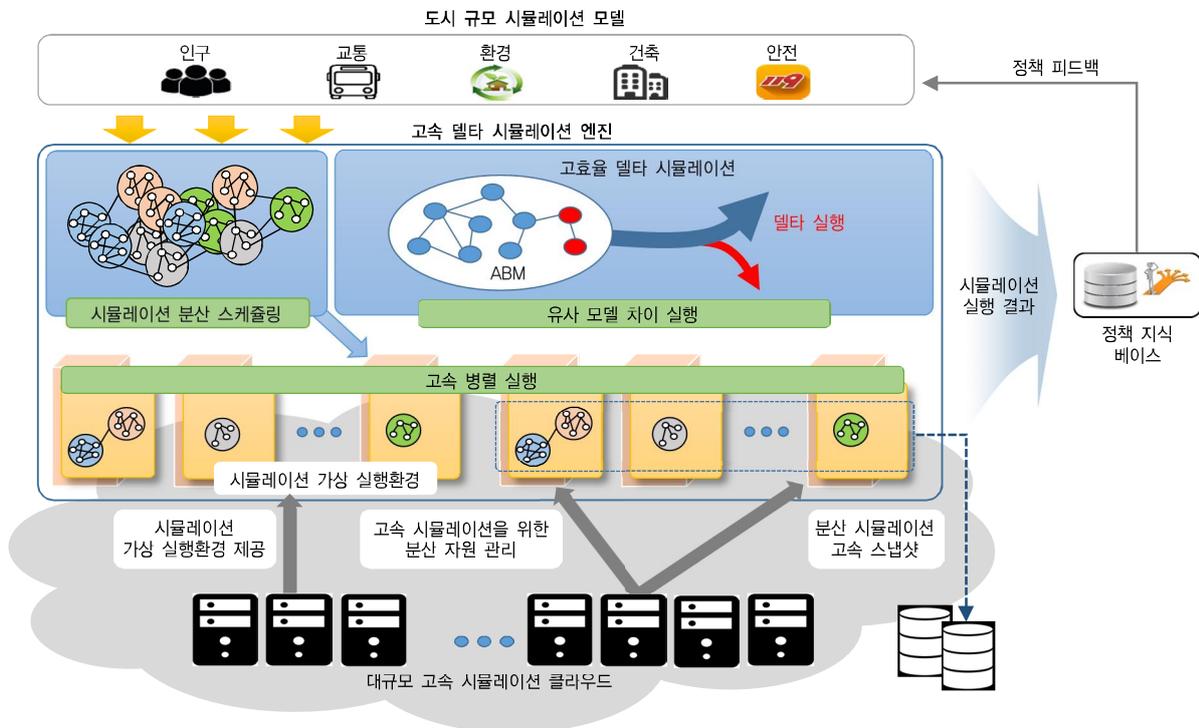
관련하여, 상기 사업은 현실세계를 모사한 가상세계에서 문제의 현상과 핵심 요인을 분석하여 근거 기반과 학적 정책수립을 지원하는 「도시행정 디지털트윈」 핵심 기술 개발을 목표로 하고 있으며, 다양한 도시 사회 문제의 해결책을 찾기 위해 데이터 기반 도시 모형 기술

을 개발하고 시뮬레이션을 통한 최적 정책 조합 및 탐색 핵심 기술을 개발하고 있다.

도시행정 디지털 트윈은 방대한 분석 객체 및 지역을 대상으로 하기 때문에 복잡한 대규모 시뮬레이션을 위한 확장가능한 고성능 클라우드 컴퓨팅 인프라를 채용하고 있으며, 사용 편의성 및 시스템 공유의 유연성 또한 반드시 확보되어야 한다.

도시행정 디지털트윈 시스템을 위한 클라우드 컴퓨팅 인프라의 주요 기능 및 특징은 다음과 같다(그림 5) 참조.

- 성능: 대규모 고속 시뮬레이션 가상 실행환경 관리
  - 시뮬레이션에 특화된 고속 가상 실행 환경 제공
  - 대규모 가상 실행 환경 라이프사이클 관리
- 확장성: 대규모 고속 시뮬레이션을 위한 분산 자원 관리



(그림 5) 클라우드 기반 고속 시뮬레이션 개념도

[출처] ETRI, "과학적 정책 수립을 위한 도시행정 디지털트윈 핵심 기술 개발", 사업계획서, 2018.

- 대규모 가상 실행 환경 자원 모니터링 및 로깅
- 대규모 시뮬레이션 가상실행 환경 동적자원 관리
- 가용성: 대규모 고속 시뮬레이션 가용성 지원
  - 가상 실행 환경 고속 스냅샷
  - 가상 실행 환경 마이그레이션

클라우드 기반 고속 시뮬레이션 시스템은 우리 사회가 직면한 미래변화 모습을 탐색하고, 복잡하고 다양한 문제의 해결책을 제시하여, 대규모의 사회적 문제를 통합적으로 해결할 수 있는 수단을 제공함으로써 ICT 기술을 통한 사회적 역할 강화, 국민행복 증진, 삶의 질 향상등을 도모할 수 있을 것으로 기대된다.

또한, 도시 내 상황을 실시간 습득·분석할 수 있도록 통합 플랫폼을 구축하고 각종 도시문제에 대한 의사결정 지원 시스템을 구축하여 일반시민, 기업 등이 참여하는 고부가가치 신산업 생태계 조성에 기여하게 될 것이다.

## V. 결론

사회·경제 현상을 분석 및 예측하기 위한 마이크로 시뮬레이션 방법으로서 ABMS가 유용한 기술로 대두되고 있으며, 동시에 대규모 에이전트와 에이전트 간 상호작용의 효율적인 고속 처리, 분석을 위한 컴퓨팅 인프라로서 클라우드 컴퓨팅이 중요성이 점차 증가하고 있다.

본 고에서는 복잡한 사회현상을 분석하기 위해 사용될 수 있는 에이전트 기반 모델링 및 시뮬레이션 도구의 기술 동향 및 클라우드 컴퓨팅 기반의 적용 연구들을 살펴보고, 향후 모델링 및 시뮬레이션 분야에 클라우드 컴퓨팅의 활용 전망을 분석하였다.

현재는 모델링 및 시뮬레이션 분야에 클라우드 컴퓨팅을 적용, 운영하기 위한 다양한 기초 연구들이 진행 중이며 클라우드의 다양한 특징 및 기능들을 최적화하여 활용하는 데는 많은 이슈가 남아있는 상황이다. 따라서, 향후 대규모 에이전트 기반 모델링 및 시뮬레이션을

위한 최적의 클라우드 컴퓨팅 환경을 위한 다수의 연구가 필요할 것으로 전망된다.

## 약어 정리

ABMS	Agent-Based Modeling & Simulation
ABaaS	ABM as a Service
CUDA	Compute Unified Device Architecture
D-MASON	Distributed MASON
FLAME	Flexible Large-scale Agent Modeling Environment
GPU	Graphics Processing Unit
HLA	High Level Architecture
HPC	High-Performance Computing
JMS	Java Message Service
MASON	Multi-Agent Simulator Of Neighborhoods or Networks
MATSim	Multi-Agent Transport Simulation Toolkit
MPI	Message Passing Interface
MSaaS	Modelling and Simulation as a Service
OpenMP	Open Multi-Processing
PDES	Parallel Discrete Event Simulation
PDES-MAS	Modeling & Simulation PDES of Multi-Agent Systems
RAL	Rutherford Appleton Lab
SLA	Service Level Agreement

## 참고문헌

- [1] 윤봉규, 이원재, “ABM 개론: Netlogo를 활용한 자연, 사회, 공학 복잡계 모델” 국방대학교 국가안전보장문제연구소, 2017, p. 13.
- [2] 정상욱, 고동우, 손동현, “마케팅에서 Agent Based Modeling의 활용,” 글로벌경영연구, 제28권 제2호, 2016.8, p. 1.
- [3] R. Axelrod and L. Tesfatstion, “A Guide for Newcomers to Agent Based Modeling in the Social Science,” *Handbook Comput. Economics*, vol. 2, 2006, pp. 3-5.
- [4] S. Abar et al., “Agent Based Modelling and Simulation Tools: a Review of the State-of-Art Software,” *Comput. Sci. Rev.*, vol. 24, 2017, pp. 13-33.
- [5] FLAME, <http://flame.ac.uk/>
- [6] HLA\_AGENT, [http://www.agents.cs.nott.ac.uk/simulation/hla\\_agent/](http://www.agents.cs.nott.ac.uk/simulation/hla_agent/)

- [7] HLA\_RePast, <http://www.cs.bham.ac.uk/research/projects/hlarepast/>
- [8] D-MASON, <https://sites.google.com/site/distributedmason/>
- [9] MATSim, <http://matsim.org/>
- [10] Pandora, <http://xrubio.github.io/pandora/>
- [11] PDES-MAS, <http://www.cs.bham.ac.uk/research/projects/pdesmas/>
- [12] Repast for High Performance Computing, [https://repast.github.io/repast\\_hpc.html](https://repast.github.io/repast_hpc.html)
- [13] SWARM, [http://www.swarm.org/wiki/Swarm\\_main\\_page](http://www.swarm.org/wiki/Swarm_main_page)
- [14] "Self-Study Guide for Repast," <http://www2.econ.iastate.edu/tesfatsi/repastsg.htm>
- [15] C. Guo and W. Xiong, "Parallel Agent-Based Simulation of Complex System Based on Repast HPC," *Int. Symp. Instrument. Measurement, Sensor Netw. Autom. (IMSNA)*, Toronto, Canada, Dec. 23-24, 2013, pp. 808-812.
- [16] 인창원 외, "인구동태 마이크로 시뮬레이션 기술동향," 전자통신동향분석, 제29권 제4호, 2014, pp. 11-20.
- [17] R. Allan, "Survey of Agent Based Modelling and Simulation Tools," 2008, <http://www.grids.ac.uk/Complex/ABMS/ABMS.html>
- [18] C.M. Pérez and E. César, "Tutorial: Agent-Based Simulations Using FLAME," *Social Simulation Conf. (SSC)*, Barcelona, Spain, Sept. 1-5, 2014.
- [19] A. Rousset, B. Herrmann, C. Lang, and L. Philippe, "A Survey on Parallel and Distributed Multi-Agent Systems," in *Workshop Parallel Distrib. Agent-Based Simulations*, Springer, 2014, pp. 371-382.
- [20] MASON, <http://cs.gmu.edu/~eclab/projects/mason/>
- [21] M. Paolucci and L. Vicidomini, "A Distributed Simulation of Roost-Based Selection for Altruistic Behavior in Vampire Bats," in *Euro-Par 2013 Workshops*, LNCS 8374, 2013, pp. 575-584.
- [22] P. Wittek and X. Rubio-Campillo, "Scalable Agent-Based Modelling with Cloud HPC Resources for Social Simulations," *IEEE Int. Conf. Cloud Comput. Technol. Sci. Proc.*, Taipei, Taiwan, Dec. 3-6, 2012, pp. 355-362.
- [23] M. Carillo et al., "D-Mason on the Cloud: an Experience with Amazon Web Services," *Euro-Par 2016: Parallel Processing Workshops*, Springer, Aug. 2016, pp. 322-333.
- [24] M. Kiran et al., "Agent-Based Modelling as a Service on Amazon EC2: Opportunities and Challenges," *IEEE/ACM Int. Conf. Utility Cloud Comput. (UCC)*, Limassol, Cyprus, 2015, pp. 251-255.
- [25] X. Liu et al., "Cloud-Based Computer Simulation: towards Planting Existing Simulation Software into the Cloud," *Simulation Modelling Practice and Theory*, vol. 26, 2012, pp. 135-150.
- [26] S.J.E. Taylor et al., "A Tutorial on Cloud Computing for Agent-Based Modeling & Simulation with Repast," in *Proc. Winter Simulation Conf. (WSC '14)*, Savannah, GA, USA, Dec. 7-10, 2014, pp.192-206.
- [27] E. Cayirci, "Modeling and Simulation as a Cloud Service: A Survey," *Winter Simulations Conf. (WSC)*, Washington, DC, USA, Dec. 8-11, 2013, pp. 389-400.
- [28] M. Carillo et al., "Distributed Simulation Optimization and Parameter Exploration Framework for the Cloud," *Simulation Modelling Practice Theory*, vol. 83, 2018, pp. 108-123.
- [29] NetLogo, <https://ccl.northwestern.edu/netlogo/>