

멀티에이전트를 이용한 분산 서버 관리 방법에 관한 연구

박민우*, 주원균*, 최기석*, 김재수*,
*한국과학기술정보연구원
e-mail:{pminwoo,joo,choi,jaesoo}@kisti.re.kr

A Study of Distributed Server Management Using Multi-Agent System

Min-Woo Park*, Won-Kyun Joo*, Gi-Seok Choi*, Jae-Soo Kim*
*KISTI(Korea Institute of Science and Technology Informantion)

요 약

최근 클라우드 서비스를 위한 가상화 서버가 많아 졌다. 사용자 측면에서는 독립적인 환경의 자원을 할당받아 자율성이 보장된 서비스를 구현할 수 있다. 서버 관리적 측면에서 보면 복잡성이 늘어나게 된다. 이를 효율적으로 하기 위해 서버의 정보와 환경설정을 쉽고, 유연하게 할 수 있어야 한다. 전체적인 IT 자원관리 측면에서 독립적으로 운영되는 서버도 관리가 필요하다. 이러한 가상서버와 비가상 서버의 정보 수집, 정보 배포, 관리를 위해 방법으로 멀티에이전트 시스템을 이용하고 중앙집중식 관리를 도메인 단위로 관리하여 부하 분산을 하여 데이터를 처리 할 수 있도록 시스템을 설계한다.

1. 서론

최근 클라우드 컴퓨팅에 대한 관심이 늘어나고 있고 이를 활용하여 서비스 및 기술에 대한 연구가 늘어나고 있다. 이러한 클라우드 컴퓨팅을 위한 기술로 가상화 기술을 활용하고, 이용자의 요구에 따라 자원을 할당하여 서비스를 제공한다. 기업에서는 서비스의 제공 형태에 따라 크게 공용 클라우드 컴퓨팅과 개인 클라우드 컴퓨팅으로 나눈다. 자원공유, 탄력성, 비용을 줄이기 위해서 클라우드 컴퓨팅을 구축하는데 정보에 대한 통제권과 서비스의 형태에 맞추는 커스터마이징 할 수 있는 개인 클라우드 컴퓨팅을 적용을 선호한다.[1]

이러한 개인 클라우드 환경을 구축하기 위해서는 클라우드 서비스를 위한 관리 방향이 필요하다. 서비스를 위한 측면에서 클라우드 서비스 관리가 필요하고, 기존의 장비와 인프라에 대한 관리를 위한 관리 방법이 필요하다.

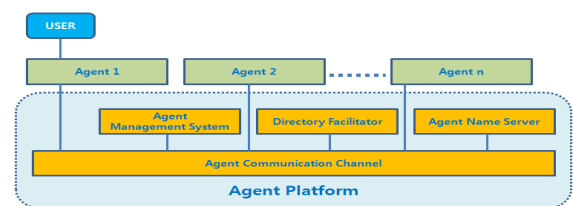
본 논문에서는 클라우드 환경을 구축하는데 있는 가상서버와 기존의 서버를 통합적으로 관리할 수 있는 방법을 멀티 에이전트를 이용하여 관리 방법을 제안하고자 한다. 분산되어 있는 서버의 정보 및 관리 요소를 수집하고, 서버를 직접적인 제어 할 수 있는 방법을 기술하고자 한다.

2. 멀티에이전트 시스템

에이전트는 사용자가 원하는 작업을 자동적으로 해결해 주는 독립적인 시스템 혹은 소프트웨어를 말한다. 에이전트의 특성은 사용자나 다른 에이전트의 직접적인 간섭 없이도 스스로 판단하는 자율성(Automomy), 스스로 추론

능력을 가지는 지능(intelligence), 표준언어나 프로토콜을 이용하여 다른 에이전트와 공통된 목적을 달성하는 협동성(cooperation), 하나의 에이전트로 처리 못하는 작업을 다른 에이전트의 도움을 받는 사교성(sociality), 환경변화에 대하여 반응할 수 있는 반응성(reactivity), 틀린 정보를 주고받지 않는 정확성(veracity), 반드시 목적을 달성하는 방향으로 작업을 수행한다는 이성적 행동(rationality)을 가지고 있다. [2]

알려진 멀티에이전트의 플랫폼은 FIPA(Foundation for Intelligent Physical Agent)이며 아래 (그림1)과 같은 구조를 가지고 있다.



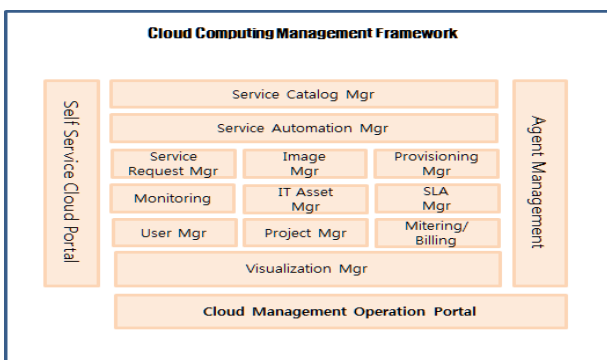
(그림 1) Structure of Multi Agent Platform

FIPA 모델에서 제안하는 에이전트 플랫폼은 ACC(Agent Communication Channel), ANS(Agent Name Server), DF(Directory Facilitator), AMS(Agent Management System)로 구성된다. ACC는 다수의 에이전트 사이에 메시지를 전달하며, ANS는 전역적인 에이전트 이름과 지역적인 전송 주소 사이의 매핑을 저장하며, DF는 에이전트의 기능과 그들이 제공하는 서비스를 저장하고, AMS는 에이전트의 생성, 삭제, 정지, 복구, 이동 등을 관리한다.[3][4]

3. Cloud Computing Management Framework

Cloud Computing Management Framework는 프라빗 클라우드 컴퓨팅환경에서 자원할당 및 관리를 중앙에서 관리를 한다. 가상화 관리 및 자원할당은 업체에서 제공하는 OpenAPI를 활용 한다. 클라우드는 기본적으로 가상화 기술 위에서 서비스가 제공하지만 검색, DB서비스와 같이 필요한 경우는 독립적인 서버에서 서비스를 제공할 수 있다. 이때 독립서버는 계정서비스와 용량을 기준으로 미터링과 빌링을 정한다. Cloud Computing Management Framework는 클라우드 서비스를 위한 VM 과 일반서버를 통합으로 관리할 수 있는 클라우드 환경의 중앙 관리 시스템 이다.

Cloud Computing Management Framework 구성은 아래 (그림2)와 같다.



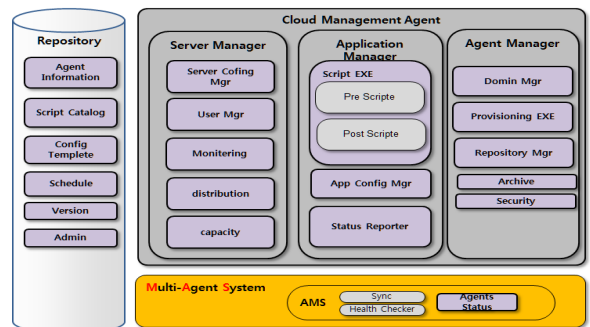
(그림 2) Cloud Computing Management Framework

- Self Service Cloud Potal : 사용자 클라우드 서비스 신청을 Potal 웹서비스
- Service Catalog Mgr : 클라우드 서비스 카탈로그 생성 및 관리
- Service Automation Mgr : 카탈로그에서 선택된 서비스를 실행
- Service Request Mgr : 서비스 요청 관리
- Image Mgr : 템플릿 및 OS 이미지 관리
- Provisioning Mgr : 사용자의 요구사항에 따라 자원을 할당, 회수 관리 자원
- Monitoring : 가상화 이벤트 및 모니터링
- IT Asset Mgr : 물리적 IT자산 관리와 정보서비스가 사용하는 논리적 자산관리
- SLA Mgr : 클라우드 사용에 따른 SLA 관리
- User Mgr : 클라우드 사용자 관리
- Project Mgr : 정보서비스의 기간, 목적, 용도에 정보 관리 및 자원회수를 위한 관리
- Mitering/Billing : 사용자의 자원사용 측정 및 과금 체계 관리
- Visualization Mgr : 인프라 자원(서버, 네트워크, 스토리지) 가상화 자원할당 및 관리
- Cloud Platform Manager Portal : 클라우드 운영자를 위한 웹서비스
- Agent Management : VM 및 Non-VM의 에이전트 Connection 및 Management

4. Cloud Management Agent

Cloud Management Agent는 중앙에 Cloud Management Frame에서 사용자에게 자원을 효율적으로 할당하고 관리를 위한 자원을 할당하기 위한 스크립트의 수행과 서버의 관리를 위한 데이터를 받아 관리를 한다. 이러한 각 서버에 분산된 데이터를 주고받기 위한 에이전트가 필요하고 이를 Cloud Management Agent가 수행하게 된다.

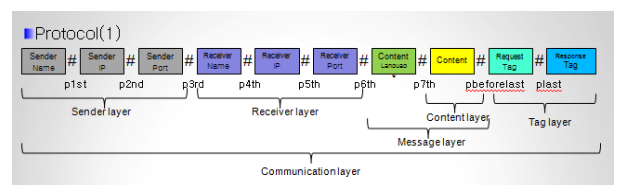
Cloud Management Agent는 아래 (그림3)과 같이 데이터 통신을 담당하는 Multi Agent System Platform위에 각각의 기능을 수행하는 3개의 Agent(Server Manager, Application Manager, Agent Manager)로 구성되고 기능은 아래와 같다.



(그림3) Architecture of Cloud Management Agent

- Agent Manager : 분산된 환경에서 도메인으로 관리하고, Provisioning 수행, Repository DB를 상호 백업 및 관리
- Server Manager : 계정서비스, Monitoring, 배포, 성능 감시
- Application Manager : Application의 Configure 및 계정 서비스를 통해 자원할당을 수행

Agent는 확장 가능한 시스템으로 설계를 하고 FIPA에 권고하는 Agent Framework Architecture를 따라 설계한다. 에이전트는 Extend Agent Programming Interface에 데이터를 수집하고 관리할 수 있는 자율성을 가진다. AMS는 Agent의 상태를 자동으로 확인하고 관리한다. 상호 데이터 전송을 위해서 (그림4)과 같이 메시지 포맷을 가지고 도메인 내에서 메시지를 전달한다. 메시지 전송은 Archive(Compress, Decompress), Security (Encryption, Decryption) Utility 모듈을 통해 네트워크 부하를 최소화하고, 한국의 표준 알고리즘인 SEED를 사용하여 보안을 고려한다.



(그림4) Message Format

사용자의 Cloud Management Agent는 많은 에이전트의 관리로 인한 부하를 분산하기 위해 자율적으로 도메인 형성하고 하나의 Agent가 서버 역할을 수행한다. 각자의 Agent의 자신의 서버에 관련된 리파지토리 DB를 가지고 있어 독립적으로 기능을 수행한다. 중앙 집중식 관리로 인한 DB 부하를 최소화 시킨다. Domain Server Agent는 도메인내의 Agent의 리파지토리 DB를 가지고 있다. Domain Server Agent 리파지토리 정보는 Computing Management framework내에 있는 Agent Management의 리파지토리 DB에 저장하고 관리를 한다. 도메인 Server Agent의 소멸 시 도메인내의 Agent에 서버 기능을 이관하고 중앙 서버에 이관사실을 통보한다.

사용자가 자원할당을 요청 시 지원서비스의 형태에 따라 서버가 결정된다. VM에서는 해당 서비스를 위한 템플릿을 통해 서비스를 위한 환경을 제공한다. 그러나 일반서버의 경우는 Agent를 통해 서버 계정을 만들고 서비스를 위한 Application에 기능을 수행할 수 있는 서비스를 제공한다. 이러한 서비스 제공을 위해 중앙에서 만들어진 스크립트를 실행하여 서비스를 생성한다. 스크립트는 Cloud Management Operation Portal 에서 스크립트는 입력하고 관리한다. 전송된 해당 Script catalog는 Agent를 통해 Repository에 저장되고 관리된다.

Server Manger Agent는 서버에서 필요한 계정서비스, 모니터링, 배포, 용량 체크, 서버환경 파일을 관리한다. 일반서버와 가상서버에 빌링을 위해 정보 수집을 하게 되고 필요한 정보 데이터는 도메인 서버에 보낸다. 각각의 도메인 서버는 중앙 서버에 데이터를 Multi AgentSystem을 통해 전송한다. 서버의 계정 서비스를 위해 디렉토리를 생성하고 필요한 환경파일을 설정해 준다.

사용자는 클라우드 요청에 대한 신속성을 위한 자동화와 많은 가상서버 및 일반서버를 관리를 위해 Agent를 통해 실행과 정보수집이 이루어진다. Agent는 요청에 따라 VM보다 일반서버에서 복잡한 과정을 수행하고 자원을 할당 한다. 많은 VM과 일반서버의 자원할당과 관리를 위한 많은 데이터를 중앙 서버에서 중앙서버로 집중하지 않고 도메인 단위로 관리하여 부하를 분산시킨다. 기존의 서버의 정보를 수집하는 단일 Agent는 수집 및 관리를 위한 새로운 기능을 추가하기 위해 Agent를 전체를 수정해야 한다. 또한 에이전트의 배포 및 관리기능이 용이하지 않다. 또한 멀티에이전트 기반으로 설계되어 새로운 기능의 Agent를 배포하여 확장성이 용이하고, 불필요한 기능의 에이전트 관리 시스템(AMS)를 통해 비활성화 시켜 Agent의 자원 사용을 줄일수 있다.

5. 결론

Cloud Computing Management는 사용자의 요구에 신속하게 자원을 할당할 수 있어야 한다. 클라우드 환경에서 직접적으로 이용할 수 없는 일반서버나 미들웨어를 Agent를 통해 관리하는 방안을 제시하였다. 장점은 Cloud Management 환경

에 IT자산과 서비스를 통합적으로 관리하고, 서비스를 신속하게 제공할 수 있는 점이다. Agent는 라이트하고, Repository DB를 각자 가지고 있고 도메인간 데이터를 전달하여 부하 분산을 한다. 또한 Multi Agent 기반으로 향후 필요한 데이터의 수집 및 관리를 위한 기능을 추가만으로 확장성에 용이하다.

참고문헌

- [1] 상상 그 이상의 파워 - 클라우드 서경구 ITWORLD CIO (2012)
- [2] Michael Wooldridge, An Introduction to MultiAgent Systems, Wiley, 2002
- [3] MultiAgent Systems , <http://www.multiagent.com/>
- [4] FIPA(Foundation for Intelligent Physical Agents , <http://www.fipa.org/>