

이종 Cloud Service Brokerage에 적용 가능한 실시간 모니터링 시스템 연구

김바울*, 조철용*, 오동휘*, 김명진*
*이노그리드 클라우드컴퓨팅연구센터

e-mail: baul · cheolyong · odonghwi · tough105@innogrid.com

Study of Real-time Monitoring System Applicable to Heterogeneous Cloud Service Brokerage

Baul Kim*, Cheolyong Jo*, Donghwi O*, Myungjin Kim*
*Cloud Computing R&D Center, Innogrid

요 약

최근 클라우드 기술이 확산되면서 여러 기업이 자신만의 클라우드를 다양하게 제공하고 있으며 세계 각지에 흩어져 있는 다양한 클라우드 공급자들과 여러 프라이빗 클라우드를 연결하여 하나의 서비스를 제공하는 클라우드 서비스 브로커리지(Cloud Services Brokerage) 기술이 등장하게 되었다. 하지만 현재의 CSB 기술은 가상 자원 관리, 가상 머신 라이프 사이클 관리에 초점이 맞추어져 있으며 CSB에서 제공되는 모니터링 요소는 사용자의 복잡한 요구를 만족하기에는 부족한 실정이다. 또한 추가 모니터링 도구를 사용하는 경우도 CSB의 데이터와의 일관성을 만족하지 않는 문제가 있다. 따라서 본 논문에서는 이종 CSB와 연동이 가능한 실시간 모니터링 시스템을 제안한다. 본 논문에서 제안하는 모니터링 기술은 Scalr 및 CompatibleOne을 사용하여 주기적으로 메타데이터 동기화를 진행하며 데이터 일관성을 충족시키고 다양한 모니터링 데이터를 실시간으로 수집하여 사용자에게 제공한다.

1. 서론

최근 클라우드 기술 확산에 따라 많은 공공기관들과 기업들이 클라우드 서비스를 구축하였으며 클라우드를 자원을 전문적으로 공급하는 클라우드 서비스 공급자(Cloud Service Provider)들이 나타나게 되었다. 또한 클라우드를 더 효과적으로 관리하기 위해 CSP의 자원을 중개하는 클라우드 서비스 브로커리지 기술(Cloud Service Brokerage)이 발전하게 되었다. CSB는 AWS, Azure와 같은 퍼블릭 클라우드나 OpenStack으로 구축한 프라이빗 클라우드 등 다양한 CSP들을 연결하여 이종 클라우드 간의 가상 머신 자원 및 관리하는 기술이다.

하지만 다양한 서비스가 혼재하는 상황에서 사용자의 예측 불가능 한 동적 자원 요구를 만족하는 것은 기존의 자원 중계를 전문적으로 하는 CSB에게는 어려운 일이다. 사용자의 자원 사용 예측과 관리를 위해서는 모니터링 데이터가 필수적이다. 하지만 기존의 CSB의 모니터링은 이종 클라우드 간 중계 기능에 비해 제공하는 정보가 많이 부족한 실정이다. CSB에서 제공하는 모니터링은 CSP에서 공급해주는 일부 데이터에 한정되고 있으며, 다른 모니터링 소프트웨어를 사용하는 경우도 하나의 메타데이터를 유지하는 CSB의 정보와 맞지 않을 가능성이 있어 데이터 일관성을 유지하는 못하는 위험성이 존재한다.

따라서 본 논문에서는 기존 CSB에서 부족한 모니터링 기능을 개선하기 위한 모니터링 시스템을 연구하였다. CSB에서 수집하기 어려운 데이터를 가상머신 내부의 Monitoring Agent를 이용하여 보다 상세한 정보를 수집하도록 하였으며 CSB의 메타 정보와 주기적으로 동기화하여 데이터 일관성을 유지하는 방법을 연구하였다.

본 논문에서는 이종 클라우드 서비스 브로커리지(Cloud Service Brokerage)에 적용 가능한 실시간 모니터링 시스템을 제시한다. 2장에서는 CSB와 모니터링에 관련된 연구를 살펴보고, 3장에서는 본 논문에서 제시한 모니터링 아키텍처를 설명하고, 4장에서는 개발된 실시간 모니터링 시스템을 확인하고, 5장에서는 본 연구의 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

클라우드 서비스 브로커리지(Cloud Service Brokerage) 기술은 클라우드 서비스 공급자(Cloud Service Provider)들을 연결하여 서비스 이용자들에게 클라우드 자원을 중개하는 기술로 클라우드 관리자가 보다 쉽게 클라우드를 관리할 수 있게 만들어준다. CSB를 위한 대표적인 소프트웨

어로 Scalr와 CompatibleOne, RightScale 등이 있다. Scalr는 웹 기반 오픈소스 클라우드 컴퓨팅 관리 플랫폼으로 이종 CSP를 관리하여 클라우드 자원을 증계하다. 오픈소스 기반 소프트웨어로 상용 소프트웨어인 RightScale에 비해 더 적은 가격으로 비슷한 기능을 제공한다[1]. 기본적으로 제공하는 모니터링 항목은 일반 모니터링 소프트웨어보다 적고 수집하는 주기도 오래 걸린다. 또한 외부로 공개된 API가 적어 다른 시스템과 연동이 어렵다는 단점이 있다. CompatibleOne은 클라우드 자원 관리 및 증계뿐 아니라 CDMI 및 OCCI와 같은 개방형 표준과 CORDS 객체 기반 설명 모델을 지원한다[2,3]. 외부에 공개된 API가 있어 다른 시스템과 연동을 지원한다. RightScale은 클라우드 서비스 공급자들의 자원 관리를 지원하는 SaaS 기반 상용 솔루션으로 클라우드 전반에 대한 통합 API를 자체적으로 제공한다[4,5].

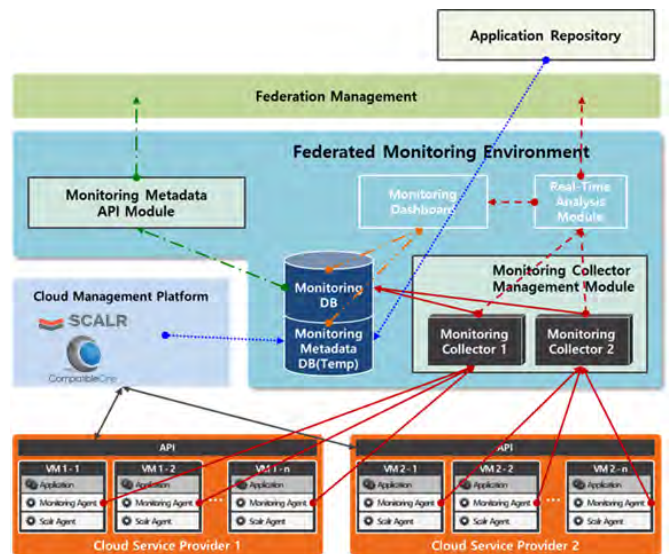
모니터링은 서버 관리자 혹은 클라우드 관리 시스템을 위해 서버의 중요한 정보를 실시간으로 수집하고 제공하는 기술이다. 대표적인 모니터링 소프트웨어로는 Ganglia, Nagios, Cacti, Zabbix 등이 있다. Ganglia는 대표적인 오픈소스 분산 모니터링 소프트웨어로 클러스터의 정보 수집에 특화되어 있다. 클라우드 환경에서도 가상머신, 서버들의 정보를 수집하는데 사용되며 전 세계적으로 많이 사용된다[6,7]. Nagios는 시스템, 네트워크 등 인프라 정보를 수집하여 제공하는 오픈소스 소프트웨어로 어플리케이션, 서버, 스위치뿐만 아니라 서비스에 대한 알람 서비스를 제공한다[8,9]. Cacti는 업계 표준인 RRDtool에 대한 응용프로그램으로 개발되었으며 오픈소스 웹 기반 모니터링 및 시각화 기능을 지원하여 CPU 부하 및 네트워크 트래픽 정보를 그래프로 변환하여 제공한다[10]. 또한 셸 스크립트를 통해 다양한 데이터를 수집하도록 손쉬운 개발 및 확장을 지원한다. Zabbix는 네트워크 및 어플리케이션을 위한 오픈소스 모니터링 솔루션으로 네트워크 서비스, 서버, 하드웨어 정보를 수집하고 추적하는 기능을 제공한다[11,12]. 다양한 모니터링 옵션을 제공하고 SMTP, HTTP를 이용하여 간단한 테스트를 실행할 수 있는 기능을 제공한다. 이러한 모니터링 시스템들은 CSB와는 독립적으로 동작하는 소프트웨어들이며 모니터링 관련 대부분의 논문들도 모니터링 시스템의 성능을 향상시키는데 초점을 맞추고 있다.

본 논문에서 제안하는 모니터링 시스템은 위와 같은 CSB들과 연계 가능하도록 아키텍처를 설계하였으며 기존 CSB의 부족한 기능인 모니터링을 보완하는 방법을 제시하였다. 또한 기존의 모니터링 소프트웨어처럼 독립적으로 동작할 뿐만 아니라 CSB와 연계하여 메타데이터 정보를 주기적으로 동기화하여 데이터 일관성을 충족하고 시스템 관리 및 운영의 편의성을 향상시켰다.

3. 본문

3.1 이종 CSB에 적용 가능한 실시간 모니터링 시스템 아키텍처

본 논문에서 제시하는 실시간 모니터링 시스템의 아키텍처는 그림 1과 같다. 하단에는 가상 머신을 제공하는 클라우드 서비스 공급자(Cloud Service Provider)들이 존재하며, 해당 CSP를 관리하는 클라우드 관리 플랫폼인 Scalr 또는 CompatibleOne에 연결되어 있다. 클라우드 관리 플랫폼과 모니터링 시스템과 연계하여 CSP 정보 등의 메타데이터 정보를 주기적으로 동기화 저장하는 Monitoring Metadata DB가 있다. 모니터링 정보 수집을 위해 각 CSP의 가상머신에 Monitoring Agent가 존재하고, 해당 정보를 수집하는 Monitoring Collector가 있다. 수집된 모니터링 데이터는 Monitoring DB에 저장된다. Real-Time Analysis Module는 실시간 모니터링 정보를 제공하고 Monitoring DashBoard를 통해 사용자에게 보여준다. 추가로 모니터링 서비스를 이용할 수 있게 Monitoring Metadata API Module을 제공하고 가상 머신에서 구동되는 어플리케이션 정보와 연계하기 위해 Application Repository를 이용한다.



(그림 1) 모니터링 시스템 아키텍처

3.2 메타데이터 동기화를 통한 데이터 일관성 유지

클라우드 관리 플랫폼과 데이터 일관성을 유지하기 위해 본 모니터링 시스템은 Scalr와 CompatibleOne의 API를 이용하여 메타데이터를 수집한다. 클라우드 관리 플랫폼은 스스로 클라우드 서비스 공급자(Cloud Service Provider)에 대한 정보를 가지고 있으며 전체 시스템 관리 및 운영 시 발생하는 CSP의 추가, 삭제에 따라 메타데이터 정보가 지속적으로 업데이트된다. 기존 모니터링 소프트웨어도 CSP의 정보가 필요하지만 모니터링 시스템 스스로 정보를 관리하기 때문에 만일 CSP 변동에 대한 정

보가 있으면 따로 정보를 업데이트해야하는 불편함이 존재한다. 따라서 클라우드 서비스 브로커리지(Cloud Service Brokerage) 기반의 클라우드 시스템에서는 CSP의 정보를 모두 가지고 있는 CSB에게 정보를 받는 것이 데이터 일관성을 유지하는 가장 좋은 방법이다.

본 연구에서는 클라우드 서비스 브로커리지를 위한 클라우드 관리 플랫폼으로 Scalr와 CompatibleOne을 선택하여 연구를 진행하였다. Scalr는 외부에서 사용할 수 있는 API가 제한적이어서 CSP 정보 수집이 불가능하지만 외부에서 데이터를 수집할 수 있도록 수정하고 Scalr가 관리하는 일부 데이터를 수집하였다. CompatibleOne은 외부에서 API를 통해 CPS 정보를 가져올 수 있도록 API를 제공하고 있어 본 모니터링 시스템에서는 해당 API를 이용하여 주기적으로 데이터를 수집하였다.

3.3 실시간 모니터링 데이터 수집 방법

본 연구에서 개발한 모니터링 시스템은 상세한 클라우드 데이터 수집을 위해 가상머신 내 Monitoring Agent를 설치하여 데이터를 수집한다. Monitoring Agent는 가상머신 생성 시 자동으로 설치되도록 Scalr와 CompatibleOne의 템플릿으로 등록하였다. Monitoring Agent는 가상머신 데이터를 수집하여 Monitoring Collector에게 보낸다. Monitoring Collector는 각 CSP 마다 자동으로 생성되며 해당 CSP의 정보를 수집한다. 수집된 데이터는 시계열 데이터베이스인 Monitoring DB에 실시간으로 저장된다.

3.4 실시간 모니터링 데이터 테이블

모니터링 시스템이 수집하는 데이터는 가상 머신의 CPU, Memory, Disk, Network 사용량 등의 자원 정보와 가상 머신의 이름, 운영체제 정보 등과 같은 식별 정보를 수집한다. 본 연구에서 개발한 모니터링 시스템이 수집하는 정보는 표 1과 같다.

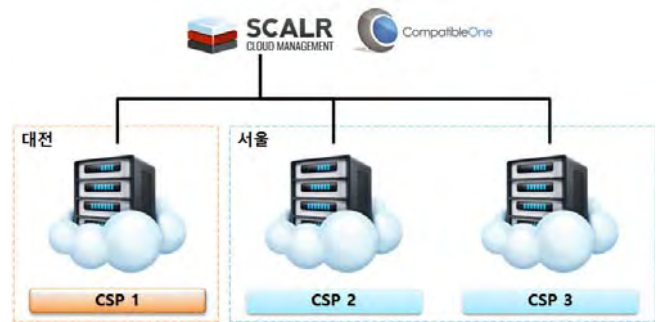
<표 1> 모니터링 아이템 리스트

Name	Unit	Description
bytes_in	byte/sec	Number of bytes read from all non-loopback interfaces
bytes_out	byte/sec	Number of bytes written to all non-loopback interfaces
bytes_read	byte/sec	Number of reads between buffers and block devices
bytes_written	byte/sec	Number of writes between buffers and block devices
cpu_idle	Percentage(%)	Percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request
cpu_intr	Percentage(%)	Percentage of IRQ (Interrupt ReQuest)
cpu_system	Percentage(%)	Percentage of CPU cycles spent in non-user mode
cpu_user	Percentage(%)	Percentage of CPU cycles spent in user mode
cpu_utilization	Percentage(%)	Percentage of CPU utilization
disk_free	Percentage(%)	Total free space on the fullest local disk partition

disk_total	Percentage(%)	Total capacity on the fullest local disk partition
disk_utilization	Percentage(%)	Percentage of disk utilization
host_name	Host Name	The name of the host machine
load_fifteen	Load Avg. (15 min.)	Reported system load, averaged over fifteen minutes
load_five	Load Avg. (5 min.)	Reported system load, averaged over five minutes
load_one	Load Avg. (1 min.)	Reported system load, averaged over one minute
mem_buffers	byte/sec	Amount of memory allocated to system buffers
mem_cached	byte/sec	Amount of memory allocated to cached data
mem_free	byte/sec	Amount of available memory
mem_shared	byte/sec	Amount of memory occupied by processes
mem_total	byte/sec	Total amount of physical memory
mem_used	byte/sec	Total memory usage
mem_utilization	Percentage(%)	Percentage of memory utilization
os_name	OS name	The name of the operating system
proc_run	Run/Total	Total number of running processes
proc_total	Run/Total	Total number of processes
swap_free	Percentage(%)	Amount of free swap space
swap_total	Percentage(%)	Amount of total swap space

4. 연구 결과

본 논문에서 개발한 실시간 모니터링 시스템의 구축 구조는 그림 2와 같고 상세한 정보는 표 2와 같다. 대전에 1개, 서울에 2개의 오픈스택 CSP를 구축하였고 Scalr와 CompatibleOne으로 모든 CSP들을 연결하여 관리하는 구조이다. 모니터링 시스템은 Scalr와 CompatibleOne이 동작하는 물리노드와 같은 로컬 네트워크로 연결되어 있다.



(그림 2) 연구 시스템 구성도

<표 2> 연구 시스템 상세 정보

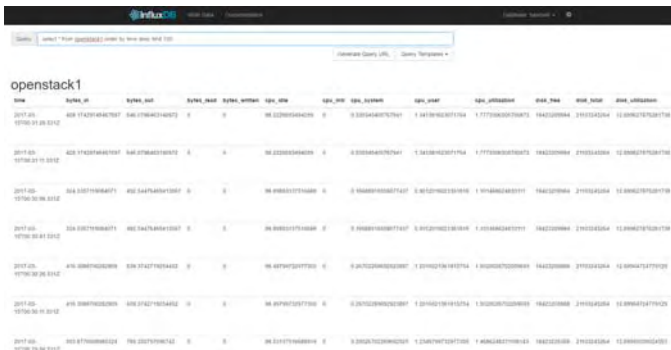
	Location	Description
Manager	서울	Two Scalr nodes - Node 1 (24Core, 48GB) - Node 2 (24Core, 48GB)
CSP 1	대전	3 OpenStack-based nodes - Node 1 (8Core, 32GB) - Node 2 (8Core, 32GB) - Node 3 (8Core, 32GB)
CSP 2	서울	3 OpenStack-based nodes - Node 1 (32Core, 96GB) - Node 2 (8Core, 32GB) - Node 3 (8Core, 32GB)
CSP 3	서울	1 OpenStack-based nodes - Node 1 (8Core, 32GB)

본 시스템에서 설치된 클라우드 관리 플랫폼 중의 하나인 Scalr에서 수집한 데이터는 그림 3과 같으며 Scalr가 제공하는 모니터링 웹 UI에서 대전과 서울에 구축된 CSP의 물리 자산 정보를 확인할 수 있었다.



(그림 3) Scalr에서 수집되는 정보

본 연구에서 개발한 모니터링 시스템으로 수집되어 저장된 데이터는 그림 4에서 보이는 것과 같이 InfluxDB가 제공하는 웹 UI를 통해 확인할 수 있었다. 해당 데이터를 검토한 결과 기존의 CSB에서 제공하는 모니터링 정보보다 더 많은 정보를 실시간으로 제공하는 것을 확인하였다.



(그림 4) Monitoring DB에 수집된 정보

5. 결론

본 논문에서는 클라우드 서비스 브로커리지(Cloud Service Brokerage)와 연계할 수 있는 실시간 클라우드 모니터링 시스템을 개발하였다. 기존 CSB의 부족한 모니터링 기능을 보완하여 독립적인 모니터링 소프트웨어 사용 시 발생하는 데이터 일관성 문제를 해소하였다. 또한 모니터링 시스템이 CSB의 메타데이터 정보와 자동으로 동기화를 진행하여 시스템 관리자의 불필요한 추가 작업을 방지할 수 있었다.

본 논문에서는 Scalr와 CompatibleOne과 연계한 모니터링 시스템을 보여줌으로 다양한 CSB에 적용 가능한 모니터링 시스템의 가능성을 보여주었다. 하지만 다양한 CSB 플랫폼과의 연동 부분을 다양하게 실험하고 평가하는 것은 부족하였다. 추후 연구에서는 본 실시간 모니터링 시스템을 개선하여 다양한 CSB 플랫폼과 연동을 보여주고 동작 및 성능을 평가할 예정이다.

Acknowledgment

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0115-16-0001, 모바일 사용자 및 어플리케이션을 위한 지역 간 클라우드 인프라 연동]

참고문헌

- [1] <https://en.wikipedia.org/wiki/Scalr>
- [2] Yangui, S., Marshall, I. J., Laisne, J. P., & Tata, S. (2014). CompatibleOne: The open source cloud broker. *Journal of Grid Computing*, 12(1), 93-109.
- [3] Carpentier, J., Gelas, J. P., Lefevre, L., Morel, M., Mornard, O., & Laisne, J. P. (2012, November). Compatibleone: Designing an energy efficient open source cloud broker. In *Cloud and Green Computing (CGC), 2012 Second International Conference on* (pp. 199-205). IEEE.
- [4] Clark, T. (2010). Quantifying the benefits of the rightscale cloud management platform. Fact point group whitepaper, funded by Rightscale.
- [5] <https://en.wikipedia.org/wiki/RightScale>
- [6] Massie, M. L., Chun, B. N., & Culler, D. E. (2004). The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7), 817-840.
- [7] <http://ganglia.sourceforge.net/>
- [8] Imamagic, E., & Dobrenic, D. (2007, June). Grid infrastructure monitoring system based on nagios. In *Proceedings of the 2007 workshop on Grid monitoring* (pp. 23-28). ACM.
- [9] Barth, W. (2008). Nagios: System and network monitoring. No Starch Press.
- [10] Haiyan, L. Y. L. J. Z. (2008). The application of cacti in the campus network traffic monitoring [j]. *Computer & Telecommunication*, 4, 004.
- [11] Mescheryakov, S. V., & Shchemelinin, D. A. (2014). Analytical Overview of Zabbix International Conference 2013. *Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление*, (1 (188)), 91-98.
- [12] Olups, R. (2010). Zabbix 1.8 network monitoring. Packt Publishing Ltd.