가상화 어플리케이션 서비스에서 사용자 로그 파일을 기반으로한 Qos를 고려한 효율적인 프로비져닝 기법

유승환*, 김성천*
*서강대학교 컴퓨터공학과
e-mail : ossforever@sogang.ac.kr

Logfile Based Concerning Quality of Service, Cost Efficient Provisioning Method for Virtualized Application Service

Seung Hwan Yoo*, Sung Chun Kim*
*Dept of Computer Science & Engineering, Sogang University

요 약

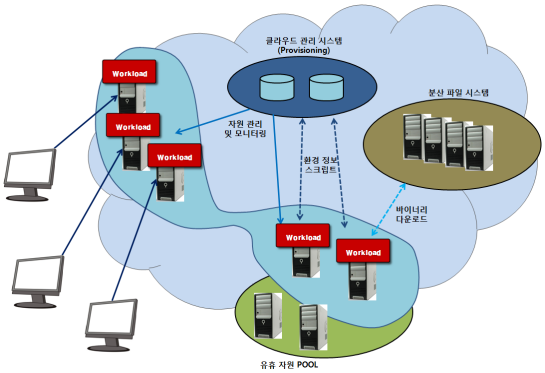
최근 클라우드 컴퓨팅에 관련된 기술이 각광을 받으면서, 기존에 인터넷을 통해 제공되던 다양한 서비스들이 클라우드 컴퓨팅 플랫폼 환경으로 이동하고 있다. 이를 통해 사용자들에게는 좀 더 편리하고 유연한 서비스를 제공하고 서비스 제공자들에게는 기존 관리 비용의 절감을 할 수 있게 되었고 이러한 변화는 새로운 컴퓨팅 환경 패러다임으로의 급속한 전환을 가져오게 되었다. 하지만 현재 폭발적인 수요의 증가로 인해, 기존의 자원 활용도의 극대화를 목적으로 고안된 자원 분배 기법들에 대한 여러 한계점이 나타나게 되었다. 본 논문에서는 이러한 문제점들을 해결하고자 클라우드 컴퓨팅 환경에서 사용자 요청 정보를 기반으로한 자원분배를 통해 서비스를 제공시 사용자 요구를 만족시키고 동시에 서비스 공급자에게는 비용 효율적인 프로비저닝(Provisioning)기법을 제안하고자 한다. 실험 결과 기존의 자원 활용도에 중점을 둔 기법보다 사용자 요청에 대한 응답 속도가 12.7% 향상되었으며, 컴퓨터 자원 유지 관련 비용면에서도 9.3%정도 절감 효과를 가져오는 것을 확인 할 수 있었다.

1. 서 론

최근 컴퓨팅 연구 분야에서 각광받고 있는 클라우드 컴퓨팅(Cloud Computing)이란 IT자원(서버, 저장장치, 네트워크, 응용 서비스 등 컴퓨팅에 필요한 모든 기능)을 직접 설치할 필요 없이 '원격으로 빌려 쓰는 서비스' 형태로 제공되는 새로운 컴퓨팅 패러다임을 말한다.

클라우드 컴퓨팅의 서비스 모델(Service Model)은 SaaS(Service as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service)로 구분한다. 본 논문에서는 이러한 클라우드 컴퓨팅 서비스를 어떻게(How) 효율적으로 사용자들에게 제공할 것인지에 대한 다양한 방법들 중에서 클라우드 컴퓨팅 환경에서 가상화 어플리케이션 서비스를 제공시 사용자 요구를 만족시키고 동시에 서비스 공급자에게는 비용 효율적인 프로비져닝(Provisioning)기법을 제안하고자 한다.

즉, 사용자에게 제공되는 '서비스의 수준(Quality of Service)을 보장'하면서, 전체 클라우드 시스템의 '운영비용(Cost)을 절감'하는 것을 핵심 목적으로 서비스를 제공하기 위해 효율적인 자원분배 기법에 대해 언급하고자 한다. [그림 1]은 제안 기법이 적용되는 클라우드 컴퓨팅 플랫폼에 대한 전체적인 구성도이다.



[그림 1] 클라우드 컴퓨팅 환경에서 프로비져닝

본 논문의 구성은 다음과 같다. 1장에 클라우드 컴퓨팅에 대한 서론에 이어 2장에서 로그 파일에 기반한 경험적

(Empirical) 시스템 모델링에 기반한 자원 분배를 수행하는 프로비져닝 기법에 대해 언급하고 3장에서는 제안된 기법을 실제 클라우드 컴퓨팅 플랫폼에 적용한 실험 결과를 설명할 것이다. 마지막 4장에서는 이를 통한 결론과 향후 연구 방향에 대해서 간략하게 정리할 것이다.

2. 제안 기법

최근 클라우드 컴퓨팅 시스템이 대중화되면서 사용자 요구에 대한 빠른 응답 속도와 이와 동시에 비용 효율적인 관리 기법에 대한 요구가 대두되고 있다. 이를 위해 본 논문에서는 과거에 요청되었던 사용자 요구들에 대한 정보를 기반으로한 시스템 모델링을 통해 사용자에 대한 응답 시간을 단축시키고 서비스 제공자들에게는 운용비용을 절감할 수 있는 새로운 프로비져닝 기법을 제안한다.

이를 위하여 우선 각 가상머신(VM)의 설정 값을 Matrix로 수식 (1)과 같이 정의한다.

$$ConfigMAT=\begin{pmatrix} c_{1,1} \cdots c_{1,M} \\ \vdots \ \ddots \ \vdots \\ c_{N,1} \cdots c_{N,M} \end{pmatrix} \quad (1)$$

다음 과정으로, 자원 할당에 대한 기준을 정하기 위해 이러한 가상 머신의 설정 값을 일정 시간동안 수집하여 평균값을 구한다. 즉, 사용자 요청에 대한 필요한 컴퓨팅 자원에 대한 평균값을 가지고 추후에 들어오는 요청값에 과의 비교를 통해 자원 할당을 수행할 것이다.

사용자 요청에 대한 보다 빠른 응답을 위해 기존 요청된 평균값과의 차이를 계산하여 허용 범위(Threshold)를 넘어서지 않는 경우, 새로운 설정 값을 계산하여 가상 머신을 생성하여 할당하는 대신 유사한 값을 가진 이전에 사용되었던 가상 머신을 할당한다. 기존의 설정값과 새로운 요청에 대한 설정값의 차이를 계산하는 방식은 수식 (2)을 통해 유클리드 공간에서 거리를 계산하는 방식을 적용한다.

$$Distance(\overrightarrow{C_{present}}, \overrightarrow{C_{average}}) = \sqrt{\sum_{i=1}^{n} f(C_{prenet_i}, C_{average_i})} \quad (2)$$

또한, 수식 (3)에서 볼 수 있듯이, 새로운 요청값이 기존 값보다 작은 경우에는 문제가 없지만 반대인 경우, 자원 할당에 대한 세심한 고려가 필요하다. 이를 위해 가중치(α)를 통해 요청 빈도가 높거나 중요한 작업에 대한 사용자 서비스 수준을 보장하고자 한다.

$$f(C_{present_i}, C_{average_i}) = \begin{cases} (C_{present_i} - C_{average_i})^2 \\ \alpha * (C_{present_i} - C_{average_i})^2 \end{cases} \quad (3)$$

$$\begin{pmatrix} \text{if } C_{present_i} \le C_{average_i} \\ otherwise, \end{pmatrix}$$

[그림 2]은 제안하는 프로비져닝 기법이 자원 요구량이 허용치를 넘었을 경우나 유휴 자원이 일정량 이상 발생할 시에 동작되는 간략한 알고리즘에 대한 의사 코드(pseudo code)이다. 여기서 허용 가능한 범위의 설정은 사용자와 서비스 제공자 간에 합의한 SLA(Service Level Agreement)을 기준으로 정한다. 예를 들어, 활용빈도가 높은 값이나 중요한 작업에 대한 설정 값들을 위에 언급된 유클리드 공간에서 중요한 작업 공간으로 정의하여, 가중치 값과 허용 범위를 미세하게 설정하여 충분한 컴퓨팅 자원 할당을 받도록 하게 한다. 수식 (2)를 통해 계산된 값은 사용자 요청에 따른 추가로 필요하거나 잉여로 남은 클라우드 컴퓨팅 자원값으로 환산하여 일정한 주기로 자원 분배를 수행하게 된다. 또한, 새로운 요청값들은 자동적으로 시스템 설정값 평균값을 계산하는 데 사용되며, 주기적으로 특정 사용자 요청에 대한 시스템 설정 평균값을 갱신한다.

<Proposed Provisioning Algorithm>

Input : |R|=current number of running resources;
|rem|: remaining tasks of all running resources;
|ari| = number of task arrivals for all resources in the previous interval;
|done| = number of all completed tasks in the previous measurement interval;
T = workload limit on each resource (number of tasks).

Output : |R'|: the required number of resources for the next interval

```
1 if |rem| > ( |R| * T)
2 if |done| < |ari|
3 get Rmore
4 |R'| = |R| + Rmore
5       else
6           |R'| = |R|
7
8       else if |rem| < ( |R| * T )
9           get Rless
10            |R'| = |R| − Rless
12
13      else
14          |R'| = |R|
```

[그림 2] 제안 프로비져닝 기법에 대한 의사 코드

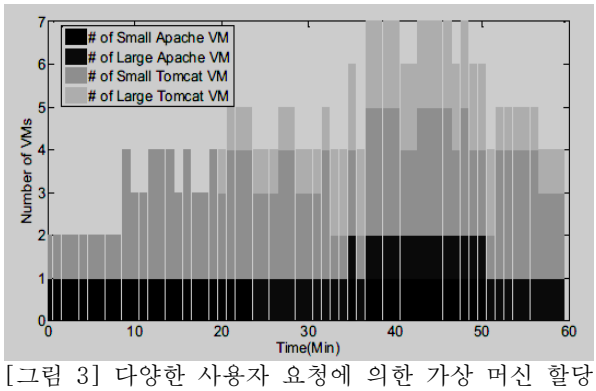
3. 실험 결과

실험은 클라우드 컴퓨팅 환경(ex> e-Bay)에서 발생하는 사용자 요청들을 발생시키는 RUBiS 벤치마크 도구를 통해서 수행하였다. <표 1>은 실험 환경에 대한 세부 설정에 대한 사항들을 정리한 것이다.

<표 1> RUBiS Benchmark 세부 설정

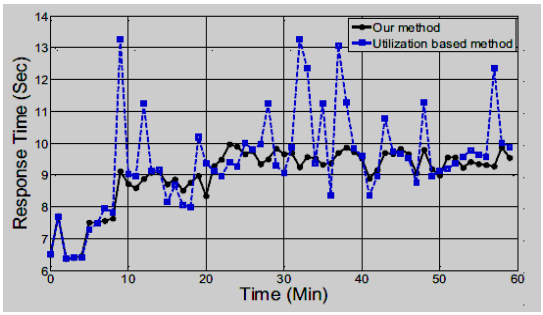
< Simulation Parameter >	
Web Server 환경 : Apache 2.0.55	
Application Server 환경 : tomcat 5.5	
Database Server 환경 : MySQL	
Hardware 환경	4Core CPU 8GB DDR3 RAM 4TB HDD 1Gbps LAN
Respond Time(SLA) : 9.5 sec	

[그림 3]는 1시간 동안 다양한 사용자 요청이 들어올 때, 그에 맞게 다양한 크기의 가상머신들이 할당되는 것을 보여주는 그래프이다. 실험결과 이를 통해 사용자 SLA를 만족시키는 비율이 87%에 근접하게 나타났다. 여기서 중요한 점은 SLA를 불만족시키는 자원할당의 경우이다. 이러한 문제를 극복하기 위해서 본 논문에서 제안하는 기법을 적용하여, 기존의 자원의 활용도를 극대화하고자는 기법과 실험을 통해 비교를 하였다.

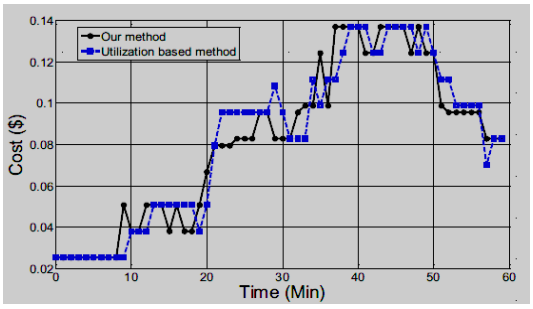


[그림 3] 다양한 사용자 요청에 의한 가상 머신 할당

[그림 4]에서 볼 수 있듯이, 제안 기법을 통한 클라우드 컴퓨팅 자원 할당을 할 경우 보다 안정성 있는 응답시간을 얻을수 있음을 알 수 있다. 또한, 기존 기법에 비해 전체적인 평균 응답시간 역시 12.7% 빨라졌음을 알 수 있다. [그림 5]는 사용자 요청에 의한 가상 머신 할당을 하기 위한 비용 발생에 대한 실험 결과이다. 그래프에서 볼 수 있듯이 전체적으로 총 발생 비용의 경우, 9.3%정도의 절감효과를 가져옴을 알 수 있었다. 다만, 새로운 요청 값이 기존 평균 설정 값과의 차이가 현격하게 발생할 때 기존 기법보다 순간적으로 비용이 더 발생하는 사례가 발견되었다.



[그림 4] 다양한 사용자 요청에 의한 응답시간 비교



[그림 5] 다양한 사용자 요청에 의한 발생비용 비교

4. 결 론

기존 클라우드 컴퓨팅 플랫폼 환경에서 자원 할당하는 프로비져닝 기법들은 자원의 활용도(Utilization)을 극대화하는데에 중점을 두고 연구되어져 왔다. 하지만 최근의 클라우드 컴퓨팅을 활용한 서비스들은 사용자들에게 신속한 서비스를 제공하야 할 필요성이 점차 커지고 있다. 때문에 본 논문에서는 사용자들에게 객관적인 수치화 된 신속한 서비스를 제공하기 위해 SLA를 통한 서비스 Qos를 고려한 프로비져닝 기법을 고안하고자 하였다. 추후 연구로는 사용자 요청이 종류가 좀 더 다양하지고 비정형적인 특징을 가지게 될 때, 발생하는 문제점들을 해결하기 위한 Qos를 보장하면서 동시에 비용 효율적인 클라우드 자원 분배 기법에 대해서 연구를 진행하고자 한다.

ACKNOWLEDGMENTS

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012R1A1A2009558)

참고문헌

[1] W. E. Walsh, G. Tesauro, and J. O. Kephart, "Utility functions in autonomic systems", Proceedings of the First IEEE International Conference on Autonomic Computing, New York, NY, USA, May 17-18, 2004.

[2] B. Urgaonkar, P. Shenoy, and A. Chandra, "Agile dynamic allocation of multi-tier Internet application", ACM Trans. on Autonomous and Adaptive Systems, 2008, vol. 3, pp. 1-39.

[3] D. Ardagna, M. Trubian, and L. Zhang, "SLA based profit optimization in multi-tier systems", Proceedings of the 4th IEEE International Symposium on Network Computing and Applications, Cambridge, Massachusetts, USA, July 27-29, 2005.

[4] P. Barham, B. Dragovic, and K. Fraser, et al, "Xen and the art of virtualization", Proceedings of the nineteenth ACM symposium on Operating systems principles, NY, USA, 2003.

[5] RUBBoS: Bulletin board benchmark. http://jmob.objectweb.org/rubbos.html

[6] S. Malkowski, M. Hedwig, D. Jayasinghe, C. Pu, and D. Neumann: CloudXplor: A tool for configuration planning in clouds based on empirical data. SAC '10.

[7] S. Malkowski, M. Hedwig, and C. Pu: xperimental evaluation of N-tier systems: Observation and analysis of multi-bottlenecks. IISWC '09.

[8] D. Feitelson: Workload modeling for computer systems performance evaluation. http://www.cs.huji.ac.il/~feit/wlmod/ ,2011.

[9] M. Hedwig, S. Malkowski, and D. Neumann: Taming energy costs of large enterprise systems through adaptive provisioning. ICIS '09.

[10] M. Hedwig, S. Malkowski, et al.: Towards autonomic cost-aware allocation of cloud resources. ICIS '10.