

클라우드 환경에서 어플리케이션에 따른 자원 스케줄링 기법

김우중, 윤찬현
한국과학기술원 그리드미들웨어연구센터
한국과학기술원 전기 및 전자공학과
e-mail : {w.j.kim, chyoun}@kaist.ac.kr

A Study on Tools for Agent System Development

Woo-Joong Kim , Chan-Hyun Youn
GRID Middleware Research Center, KAIST
Dept. of Electrical Engineering, KAIST

요 약

많은 과학 분야에서 필수적인 요소인 High-performance computing 은 많은 컴퓨팅 자원을 필요로 한다. 이에 인터넷을 통해 사용자가 필요한 만큼의 컴퓨팅 자원을 제공받고 사용량만큼 비용을 지불할 수 있는 클라우드 컴퓨팅이 하나의 해결책으로 제시된다. 하지만 클라우드 컴퓨팅을 사용하기 얼마만큼의 자원이 필요할 것인가를 결정하는 문제가 있다.

본 논문에선 동적인 클라우드 서비스 환경에서 사용자가 실행하고자 하는 특정 어플리케이션에 대해 쓸모 없는 자원을 최대한으로 줄여 비용 대비 성능의 자원을 제공하는 브로커를 제안한다.

1. 서론

현대에 High-performance computing 는 많은 과학 분야에서 필수적인 요소가 되었다. 이로 인해 다양한 scientific application 이 개발 되었는데 이는 많은 컴퓨팅 자원을 필요로 한다. 이에 인터넷을 통해 사용자가 필요한 만큼의 컴퓨팅 자원을 제공받고 사용량만큼 비용을 지불할 수 있는 클라우드 컴퓨팅이 하나의 해결책으로 제시된다.

클라우드 컴퓨팅을 사용하기 앞서 문제가 되는 것이 특정 scientific application 을 수행하기 위해서 얼마만큼의 자원이 필요할 것인가를 결정하는 것이다. 어플리케이션이 어느 이상의 자원을 할당하면 성능이 포화되어 증가하지 않기 때문에 최대의 성능을 내면서 최소의 자원을 할당 할 수 있어야 한다. 클라우드 컴퓨팅 서비스 사용자는 직관적으로 어떤 사양(Core 개수, Memory 크기, Storage 크기 등)이 필요 할지 알기 힘들다. 더구나 클라우드 컴퓨팅 시스템에서 한 PM 내의 VM 간에는 interference 가 발생해 때에 따라 사양 대비 더 심각한 성능 저하를 보일 수 있다. 그렇기 때문에 이런 동적인 상황(i.e. interference)까지 고려해 사용자가 요청한 application 에 대해 자원을 스케줄링 할 수 있어야 한다.

본 논문에선 동적인 클라우드 서비스 환경에서 사용자가 실행하고자 하는 특정 어플리케이션에 대해 쓸모 없는 자원을 최대한으로 줄여 비용 대비 성능의

자원을 제공하는 브로커를 제안한다.

2. 관련 연구

특정 어플리케이션에 대한 자원 스케줄링을 하기 위해서 할당한 자원에 따른 성능 모델을 예측하는 것이 중요하다. 이에 관한 기존의 많은 연구들이 있었다.

[1]에선 기계 학습 모델 중 인공 신경망 모델(ANN: Artificial Neural Network)을 이용해 특정 가상머신에서 특정 어플리케이션이 동작할 때 성능을 예측하므로써 자원 스케줄링하는 방법을 제안한다. 하지만, 인공 신경망 모델을 이용해 높은 정확도의 예측을 얻기 위해선 높은 복잡도를 요구하기 때문에 예측 시간이 오래 걸리는 단점이 있다. 그리고 모델을 학습시키기 위해 많은 과거 정보가 필요하다. 이렇게 만든 성능 모델로 목표 성능(target performance)을 낼 수 있는 CPU 클럭(CPU clock), 메모리 크기(memory size)를 예측하지만 클라우드 서비스 환경에선 정해진 VM Type(small, medium, large)의 개수가 많지 않기 때문에 이러한 방법을 통한 스케줄링은 과분하다.

[2]에선 회귀 분석을 통해 수행시간을 예측하는 자원 스케줄링 하는 방법을 제안한다. 어플리케이션 종류에 따라 다른 입력 변수의 다른 모델을 사용하게 되어 통일성을 떨어뜨리고 복잡도를 증가시킨다.

본 논문에선 비교적 낮은 복잡도와 좀 더 직관적인

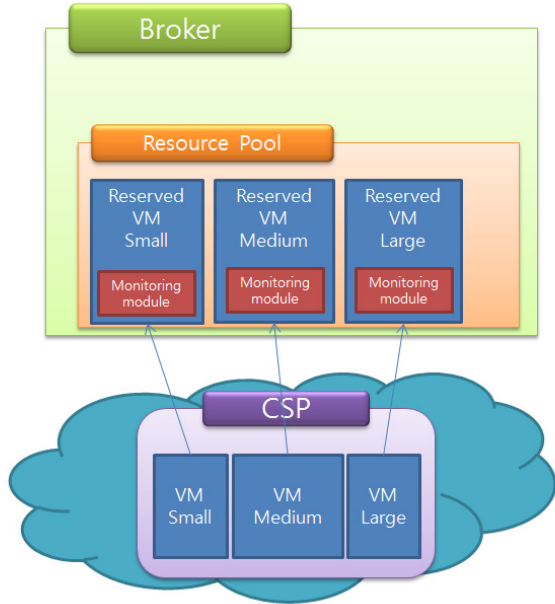
알고리즘을 통한 자원 스케줄링을 제시한다.

$$x_t = \theta_0 + \theta_1 x_{t-1} + \theta_2 x_{t-2} + \dots + \theta_n x_{t-n}$$

$$y_t = \theta_0 + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \dots + \theta_n y_{t-n}$$

$$z_t = \theta_0 + \theta_1 z_{t-1} + \theta_2 z_{t-2} + \dots + \theta_n z_{t-n}$$

3. 제안하는 브로커 구조



(그림 1) 제안하는 브로커 구조

브로커는 사용자의 요청량을 예측해 그에 알맞는 Reserved VM 을 미리 클라우드 서비스 제공자로부터 생성해 리소스 풀(resource pool)을 형성한다. Reserved VM 은 Amazon EC2 서비스에서 on-demand VM, Spot VM 과 함께 제공하는 서비스로 장기간 VM 을 계약하고 싶을 경우 좀 더 싼 가격에 VM 을 생성할 수 있다. 브로커의 리소스 풀 내에 VM 들은 백그라운드 프로세스로 모니터링 모듈이 실행된다. 모니터링 모듈은 [3]collectd 를 이용하고 이를 통해 VM 의 실시간 자원 사용량을 측정할 수 있다. CPU 사용량, 메모리 사용량 등의 모든 자원의 사용량을 얻을 수 있다.

4. 제안하는 브로커의 알고리즘

특정 어플리케이션이 요청되었을 때 가장 먼저 VM type 중 가장 큰 large 타입에서 실행 시켜보면 해당 어플리케이션의 resource utilization profile 을 알 수 있다. 기본적인 방법은 측정된 resource utilization profile 만큼만 자원을 할당하면 된다는 것이다. 가장 처음에는 이와 같은 한번의 pre-running 후에 얻은 resource utilization profile 을 포용하는 최소 VM Type 을 할당한다. 하지만 가상화 환경에선 VM 간의 interference 가 존재하기 때문에 이후에도 이와 같은 utilization 을 보인다는 보장이 없다. 그래서 이후 요청 시 utilization 을 예측하기 위해 선형 회귀 방법으로 접근한다. 현재로부터 정해진 window 크기 만큼의 과거 utilization 데이터를 이용해 미래의 utilization 을 측정한다.

x,y,z 는 각각 CPU core 개수, Memory 크기, Storage Size 이다. Window 크기는 n 이며 각 자원에 대해 n 개의 θ 값을 과거 정보를 통해 예측한다.

5. 결론

본 논문에선 현대 과학에서 많이 사용되고 있는 Scientific Application 을 클라우드 서비스 환경에 사용하고자 할 때 자원을 스케줄링하는 단순하고 직관적인 브로커 구조 및 알고리즘을 제안했다.

Acknowledgement

이 논문은 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단-클라우드 Collaboration 기술 사업과 BK21 사업의 지원을 받아 수행된 연구임(No. 2012-0006425)

참고문헌

- [1] S. Kundu, R. Rangaswami, A. Gulati, M. Zhao, and K. Dutta. Modeling Virtualized Applications using Machine Learning Techniques. In VEE '12.
- [2] Shuichi Shimizu, Raju Rangaswami, Hector A. Duran-Limon, and Manuel Corona-Perez. "Platform-independent modeling and prediction of application resource usage characteristics". The Journal of Systems & Software, vol. 82, pp. 2117-2127, Dec. 2009.
- [3] collectd www.collectd.org/