

# 단일 가상 머신-다중 작업 할당 기법 기반 고효율 클라우드 자원 브로커 시스템

김성환\*, 하윤기\*, 윤찬현\*

\* 한국과학기술원 전기 및 전자 공학과

e-mail : {s.h\_kim, milmgas, chyoun}@kaist.ac.kr

## Multiple Request per Single Virtual Machine Scheme based High Efficiency Cloud Resource Broker System

Seong-Hwan Kim\*, Yun-Gi Ha\*\*, Chan-Hyun Youn\*

\*Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology

### 요 약

비용대비 작업 처리의 효율을 위해서는 사용자들의 작업 요구사항에 적절한 자원을 선택하고 요구 작업을 적절한 할당된 자원에 스케줄링하는 플랫폼이 필수적이다. 또한 이러한 플랫폼은 사용자의 SLA 에 따라 작업 처리 기한 안에 요구 비용 이내로 작업을 처리할 수 있도록 결정을 내릴 수 있어야 하고 요구 작업량의 변화에 따라 즉각 대응을 하기 위하여 실시간적인 결정을 내릴 수 있어야 한다. 이러한 복잡한 결정 사항들을 최적 판단으로 대신 처리해주는 미들웨어로 클라우드 자원 브로커 시스템을 사용할 수 있다. 클라우드 자원 브로커 시스템은 작업 스케줄링과 자원 프로비저닝 등이 가격, 처리시간에 중요한 선택 및 수행을 한다. 기존의 많은 논문들에서의 작업 스케줄링은 다중 테넌트 정책의 클라우드가 제공하는 사용자들간의 가상 머신 독립에 초점을 두어 하나의 가상 머신이 하나의 작업에 한정되도록 처리하는 방식이었다. 이는 병렬화의 정도가 낮은 어플리케이션의 경우 시스템 활용률이 낮아 자원 활용율이 떨어진다. 이를 다수의 작업을 멀티 태스킹, 멀티 스레드의 방법으로 하나의 가상 머신에서 처리하도록 하여 스레드 레벨 병렬화의 이점을 이용해 자원 이용률을 높임으로 효율을 높이고자 한다.

### 1. 서론

클라우드 컴퓨팅은 컴퓨팅 자원의 효율적인 사용을 위해서 가상화 기술을 기반으로 하여 하나의 물리적인 머신을 다수의 논리적이고 독립적인 가상 머신들로 나누어 제공하는 플랫폼으로 컴퓨팅 자원을 이용하기 위한 초기 구입 비용과 서버 관리비용을 줄일 수 있고 다이나믹한 자원 요구에 대해서도 쉽게 대처할 수 있는 모델이다 [1]. 사용자들은 각자의 작업을 처리하기 위해 클라우드 서비스 프로바이더(CSP)들에게 CPU, Mem, Storage 의 정보를 기반으로 자원을 요청한다. 이러한 자원의 크기에 따라 다른 과금 정책이 적용되므로[2] 작업 처리에 걸리는 시간과 deadline 그리고 작업을 처리할 머신의 크기를 기반으로 선택하는 것이 중요하고 이는 비용에 큰 영향을 미친다. 또한 작업이 어떠한 가상 머신에 할당되는가라는 문제도 최적화 관점에서 고려하여야 한다. 사용자와 클라우드 서비스 프로바이더의 사이에서 서로의 요구사항과 정책을 중재하여 효율적인 작업 및 자원 관리를 할 수 있도록 클라우드 자원 브로커라는 미들웨어가 연구되고 있다. 이 논문에서는 자원 프로비저닝 알고리즘은 간단한 Two level threshold 정책을 기반으로 VM 의 타입, 개수를 변화시킨다고 두고 이중의 가상 자원들 사이에서 작업이 어떤 머신에 할당되어야 하

는지를 효율, 가격 대비 성능, 자원 이용률을 높이기 위한 결정을 내리는 기법에 대해서 설명하고자 한다. 그리고 실험을 통해 해당 기법의 우수성을 확인한다.

### 2. 본론

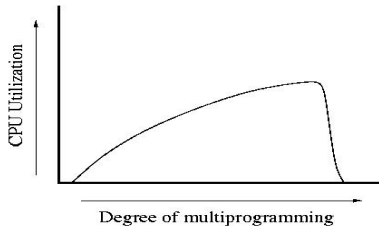
#### 2.1 클라우드 리소스 브로커 시스템

제안하는 클라우드 리소스 브로커 시스템은 클라우드 API 와 연동하여 자원을 요청하는 Cloud Node Adaptor, 워크로드에 따라 적절한 리소스의 개수를 유지하도록 하는 Resource Provisioning Manager, 작업을 어떠한 가상 머신에서 구동해야 할 것인지를 선택하는 VM Allocation Manager 등으로 구성되어 있으며 이번 논문에서는 VM Allocation Manager 의 기능 중 일부를 중점으로 다룬다.

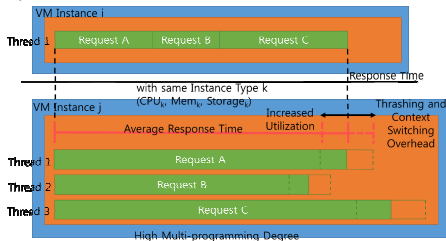
#### 2.2 다중 프로그래밍 정도와 자원 이용률

현재의 CPU 는 내부의 레지스터에 비해 아주 느린 처리속도를 가지는 I/O, Memory 가 수 없이 많은 stall 들 발생시킨다. 하나의 스레드가 존재한다고 가정하는 경우에 한 스레드가 I/O 를 기다리는 동안 CPU 는 idle 상태로 기다리고 있어 자원 사용률이 떨어지고 작업 효율이 떨어지며 결국은 전체 작업을 처리하는데 들어가는 비용은 증가하는 것이다. 반면 두 스레드가 번갈아 가며 I/O, CPU 를 요구할 경우 CPU 사용과 I/O 사용이 오버랩 될 확률이 높아져 이용률을 높

일 수 있다. 이는 멀티 코어 시스템에서 다중프로그래밍의 정도가 높아질수록 스레드 레벨 병렬성이 커져 자원 이용률이 증가함을 뜻한다. 하지만 특정 정도의 다중프로그래밍 정도를 넘어설 경우 스래싱에 의해 자원 이용률이 급격히 떨어지는데 이는 컨텍스트의 잦은 교체에서 발생하는 빈번한 캐시 미스, page fault 때문이라 볼 수 있다. 아래의 왼쪽 그림에서 그러한 경향성을 볼 수 있으며 우리는 이러한 스래싱을 겪지 않도록 다중 프로그래밍 정도를 제한해야 할 필요가 있다. 오른쪽 그림은 같은 작업을 직렬로 실행한 경우와 병렬로 실행한 경우 이용률의 이점으로 병렬의 경우가 좀 더 단시간에 작업을 처리함을 볼 수 있다. 이 때 주의할 점은 평균 응답시간은 직렬일 경우가 더욱 좋다는 것이다.



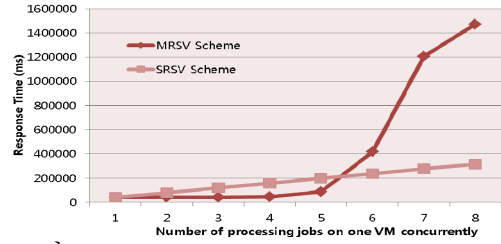
(그림 1) 다중프로그래밍 정도와 자원 이용률 [3]



(그림 2) 다중 프로그래밍 정도에 따른 수행시간 비교

**2.3 MRSV 기법을 이용한 가상 머신 할당 기법**

기존의 가상 머신 할당 기법은 각각의 가상 머신의 독립성을 얻기 위해 하나의 가상 머신에서는 하나의 작업만 처리하도록 하는(SRSV, Single Request per Single VM) 기법을 주로 이용하였다. 현재 컴퓨터 공학에서 작업의 병렬화를 위해 지속적인 연구를 하고 있으나 현재 시점에서는 프로그램의 복잡도에 따라 난이도가 높아지고 한계가 존재하므로 SRSV의 경우에는 효율성이 떨어질 것임을 생각할 수 있다. 따라서 다중 스레드, 다중 프로세스를 이용해 추가적인 병렬화를 통해 자원 이용률을 높이려는 노력이 MRSV 기법(Multi Request per Single VM)이다. 앞에서 언급했듯이 일반적으로 SRSV가 MRSV 보다는 응답시간이 짧아 고성능 처리나 실시간 처리와 같은 특정 영역에서는 SRSV가 유리하지만 일반적인 작업에서는 MRSV가 유리하다. 작업의 크기에 따라 가상 머신의 크기를 정해주는 모듈을 통해 해당 작업이 처리되기에 적합한 가상 머신 그룹이 선택된다. 각 가상 머신에서 해당 작업을 처리하기에 가장 이득이 되는 수를 찾기 위해 실제 응용의 테스트를 진행하는데 그림 3에서와 같이 MRSV가 SRSV에 비해 최대의 이점을 가지는 6개 가상 머신의 최대의 작업 동시 처리수로 선택할 경우 응답시간이 deadline을 넘지 않는 선에서 최대의 이득을 가진다. 가상 머신 그룹 중 특정 가상 머신 선택은 정책에 따라 load balancing이 될 수 있고 반대로

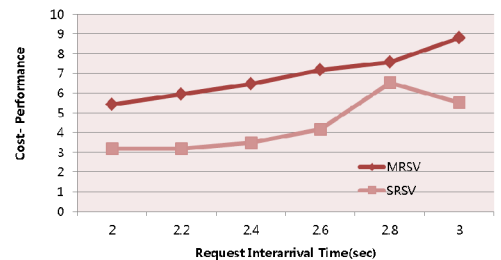


(그림 3) Empirical measurement with Test Set

가장 많은 작업을 처리하는 가상 머신에 할당하여 최대한 사용되는 가상 머신의 개수를 줄이는 방법도 있다.

**2.4 실험 및 결과**

실험은 4개의 컴퓨팅 노드(HP, Xeon E5620 2.8G, Core8, Mem 16GB, HDD 1TB 환경)들을 오픈스택 기반으로 구축한 환경에서 진행하였다. 브로커에는 약물/화학 분자의 특성을 분석해주는 과학 응용 모듈을 구동하고 가상의 클라이언트에서는 랜덤한 크기의 작업들을 interarrival time을 바꾸어 가며 일정 시간 동안 브로커에 요청하였다. 이후 반응시간, 가상 머신 비용 등을 측정하여 SRSV와 MRSV를 비교할 다양한 특성을 그래프화 하였다. 이 때 특히 ‘처리 시간에 대비 레하는 성능’을 ‘가상 머신 사용 비용’으로 나눈 ‘비용 대비 성능’을 비교해 보았을 때 MRSV가 더 좋은 특성을 나타냄을 확인할 수 있고 이는 MRSV가 시스템을 좀 더 효율적으로 사용하는 것을 나타낸다.



(그림 4) 실험 결과

**2.5 결론**

본 논문에서는 MRSV 기법을 이용하여 고효율을 가지는 클라우드 자원 브로커를 설계하였다. 다중 프로그래밍 정도의 증가에 따른 시스템 자원 이용률 변화의 이론을 기반으로 스래싱이 발생하지 않는 범위에서 각 가상 머신이 처리해야 하는 최적의 작업 할당 수를 결정하였고 두 가지 가상 머신 선택 정책을 보였으며 실험을 통해 MRSV 기법이 가지는 효율성을 보였다.

**Acknowledgement**

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

**참고문헌**

[1] Michael Armbrust 외 4명 “Above the Clouds: A Berkeley View of Cloud Computing”  
 [2] <http://www.gogrid.com/products/infrastructure-cloud-servers>  
 [3] A. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts, vol. 32, no. 8. Wiley, 2005, p. 921.