

효과적인 VM 프로비저닝을 위한 VM 간섭 모델에 대한 연구

주경노, 윤찬현
한국과학기술원 전기및전자공학과
e-mail : {eu8198, chyoung}@kaist.ac.kr

A Study on VM Interference Modeling for Effective VM Provisioning

Kyung-No Joo, Chan-Hyun Youn
Dept. of Electrical Engineering, KAIST

요 약

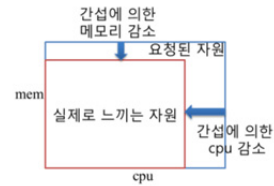
VM 간섭은 VM 프로비저닝을 할 때 예상된 VM의 컴퓨팅 자원이 실제와 다르지 않도록 하기 위해 반드시 고려되어야 할 사항이다. 이에 본 논문에서는 예상된 VM의 컴퓨팅 자원을 사용자가 보장받을 수 있도록 랜덤 워크를 이용해 간섭을 고려한 필요 자원을 구하는 방법에 대해 다루고 있다.

1. 서론

클라우드 컴퓨팅(Cloud Computing)은 CPU, RAM, 스토리지, 네트워크와 같은 여러 IT 자원들을 인터넷을 통해 필요한 만큼 빌려 쓰고, 사용한 만큼 비용을 지불하는 Pay for use의 형태를 갖는 서비스로, 클라우드로 표현되는 인터넷상의 서버에서 데이터를 저장, 처리 등 IT 관련 서비스를 한 번에 제공하는 컴퓨팅 기술을 말한다. 이를 통해 사용자들은 자원에 구애받지 않고 어플리케이션을 실행할 수 있게 되었다. 이를 가능케 하는 것은 가상화 기술이다. 이를 통해 하나의 물리적 머신(Physical Machine, PM)의 자원을 나눠 여러 가상 머신(Virtual Machine, VM)들을 만들 수 있게 되었다. 따라서 클라우드 서비스 제공자들은 사용자가 원하는 양의 자원에 해당하는 VM을 사용자에게 제공함으로써 클라우드 컴퓨팅을 실현시킬 수 있다. 또한, 서버들의 효율성(Utilization) 또한 증가되었다. 그러나 L2 캐시나 네트워크 대역폭 등의 자원은 다른 자원들과 달리 가상화가 불가능하기 때문에 PM 내의 모든 VM들은 위의 자원들을 공유해야 한다[1]. 따라서 한 PM 내에 여러 VM들이 각자의 어플리케이션을 돌리게 되면 위의 자원들에 부하가 생겨 같은 자원이더라도 성능이 떨어지는 효과가 생기게 된다. 이를 VM 간섭(VM Interference)라고 한다[2].

따라서 사용자가 어플리케이션을 돌리기 위해 SLA(Service Level Agreement)를 정해놓은 경우, VM 간섭으로 인해 원하는 만큼의 자원을 얻지 못해 데드라인을 넘기게 될 수 있으며 그에 따라 불이익을 볼 수 있다. 따라서 VM 간섭은 VM 프로비저닝을 할 때 예상된 VM의 컴퓨팅 자원이 실제와 다르지 않도록 하기 위해 반드시 고려되어야 할 사항이다. 이에 본 논문에서는 랜덤 워크를 이용해 간섭을 고려한 필요 자원을 구하는 방법에 대해 다루고 있다. 이에 따라 사용자는 간섭이 있더라도 예상된 VM의 컴퓨팅 자원 이상을 보장받게 되어 데드라인을 지킬 수 있게 된다.

2. VM 간섭에 의한 효과

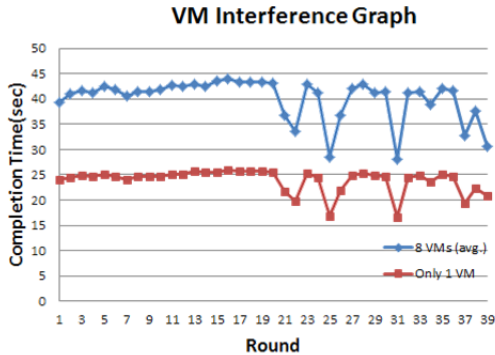


(그림 1) 간섭으로 인해 감소된 리소스

그림 1은 VM 간섭으로 인해 리소스가 줄어드는 효과가 생기는 것을 나타낸 그림이다. 파란 선이 사용자가 의도한 VM의 스펙이며, 빨간 선은 실제로 사용자가 갖게 되는 리소스로 간섭이 생기기 때문에 사용자가 의도한 양보다 더 적은 리소스만 갖게 된다. 특히, 한 어플리케이션을 실행하기 위해 여러 VM을 사용하는 경우에는 각각의 VM에서의 간섭에 의한 리소스 감소 효과가 합쳐지게 되기 때문에 무시할 수 없게 된다. 따라서 여러 VM 위에서 어플리케이션이 돌아가는 경우 간섭에 의한 효과는 훨씬 커진다. 따라서 실제 생성되는 VM의 리소스와 의도된 리소스 양이 그림 1처럼 다르기 때문에 실제 어플리케이션을 돌리기 위해 사용된 VM들의 실제 리소스 합이 워크로드를 넘지 못한다면 제대로 VM을 할당하더라도 주어진 워크로드를 데드라인 내에 마치지 못하게 될 수 있다. 본 논문에서는 이런 현상을 막기 위해 VM 간섭이 생기더라도 사용자가 사용하고자 의도한 자원을 보장받기 위해 간섭을 고려하였을 때 얼마만큼의 자원을 요청할 지를 결정하도록 한다.

그림 2는 실험적으로 간섭의 효과를 측정한 그래프이다. 같은 워크로드를 하나의 VM에서 돌릴 때 약 25초가 걸린 반면, 한 PM 내에 8개의 VM이 돌아가고 있는 경우 어플리케이션을 마치는 데 걸리는 시간이 약 45초가 걸린 것을 확인할 수 있다. 이는 간섭에 의한 리소스 감소 효과가 무시할 수 없을 정도임

을 보여준다.



(그림 2) VM 간섭이 미치는 영향

3. VM 간섭 모델링

4.1. 가정

(1) 사용자는 구체적인 자원의 형태로 VM 을 요청하며, 사용자가 요청한 컴퓨팅 자원만을 보장토록 하는 것이 목적이기 때문에 어플리케이션을 돌리는 데 필요한 자원이 얼마인지는 생각하지 않는다. 본 논문에서는 cpu, 메모리, I/O 의 3 개 자원만을 고려하였다.

(2) VM 간섭을 모델링 하기 위해서는 PM 내에 있는 VM 의 개수, VM 에서 동작하는 어플리케이션이 주로 사용하는 자원 등의 정보가 필요하며, 시간에 따라서 어떻게 변하는지를 파악해야 하나, 시간적인 무리가 있다. 따라서 본 논문에서는 리소스별 VM 간섭이 평균 m , 분산 σ^2 인 Gaussian Distribution 을 따른다고 생각하였다. 즉, 각 리소스 컴포넌트들이 위와 같은 정규분포를 따를 때,

$$\begin{aligned} \Delta VM_{cpu} &\sim N(m_{cpu}, \sigma_{cpu}^2) \\ \Delta VM_{mem} &\sim N(m_{mem}, \sigma_{mem}^2) \\ \Delta VM_{io} &\sim N(m_{io}, \sigma_{io}^2) \end{aligned}$$

가 되며, VM 간섭의 크기를 (cpu, mem, io)의 형태로 나타내면 $\Delta VM = (\Delta VM_{cpu}, \Delta VM_{mem}, \Delta VM_{io})$ 와 같이 된다.

(3) VM 이 다른 PM 위에 있더라도, 리소스별 VM 간섭의 평균과 분산은 크게 다르지 않으므로 같게 생각하였다.

4.2. 모델링

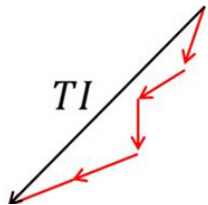


그림 3 간섭벡터

여러 VM 들 위에서 어플리케이션이 돌아가고 있기 때문에 우리는 모든 VM 들에서의 합산된 간섭 효과를 그림 3 과 같이 벡터의 형태로 모델링하려고 한다. 그림 3 에서, 빨간색 벡터는 각각의 VM 에 대한 간섭이다. 이때 모든 VM 들의 간섭 효과를 총합한 벡터를 TI(Total Interference)라고 하였다.

이 때, TI 의 분포를 구할 수 있으면 사용자는 의도된 리소스에 예상된 TI 만큼의 자원을 요청함으로써 의도된 자원을 보장받을 수 있다. TI 는 n step 을 거친 random walk 로 모델링할 수 있다. 단, random walk 에서는 각 스텝의 크기가 1 이었으나, 이 모델에서는 각

스텝의 크기가 평균 m , 분산 σ^2 인 Gaussian Distribution 을 따른다.

사용자가 2 개의 VM 을 사용한다고 생각하자. 이 때 각각의 VM 간섭 벡터를 $\Delta VM_1, \Delta VM_2$ 라고 할 때, cpu 에 해당하는 간섭을 더해보면 다음과 같다.

$$\begin{aligned} \Delta VM_{cpu,1}, \Delta VM_{cpu,2} &\sim N(m_{cpu}, \sigma_{cpu}^2) \\ TI_{cpu} &= \Delta VM_{cpu,1} + \Delta VM_{cpu,2} \end{aligned}$$

이 때, TI 는 두 개의 Gaussian Random Variable 의 합으로 아래의 분포를 따른다.

$$TI_{cpu} \sim N(m_{cpu,1} + m_{cpu,2}, \sigma_{cpu,1}^2 + \sigma_{cpu,2}^2)$$

가정 3 에 의해, $TI_{cpu} \sim N(2m_{cpu}, 2\sigma_{cpu}^2)$ 가 된다.

같은 원리로, n 개의 VM 이 사용된 경우 n 개의 step 을 밟는 Gaussian Random Walk 라고 생각하여 풀면

$$TI_{cpu} \sim N(nm_{cpu}, n\sigma_{cpu}^2)$$

가 된다. 다른 리소스 축에 대해서도 풀면 아래와 같은 식이 나온다. 본 논문에서는 cpu, mem, i/o 의 3 개 축만 가정하였으나, 다른 자원들에 대해서도 같은 식이 세워진다.

$$\begin{aligned} TI_{mem} &\sim N(nm_{mem}, n\sigma_{mem}^2) \\ TI_{io} &\sim N(nm_{io}, n\sigma_{io}^2) \end{aligned}$$

따라서 벤치마킹 툴을 통해 리소스별로 간섭의 평균과 분산을 측정하면 n 개의 VM 을 생성하였을 때의 평균 간섭과 분산을 구할 수 있다. 따라서 n 개의 VM 에 의한 간섭을 보완하기 위해서는 위의 분포로 이루어진 정규분포를 표준정규분포로 변환한 후 0.4(90%) 혹은 그 이상에 해당하는 값을 계산해 기존에 사용자가 요구한 자원에 추가 지급하면 된다.

4. 결론 및 제언

본 논문에서는 사용자가 요청한 VM 자원을 보장받을 수 있도록 랜덤 워크 모델을 이용해 간섭을 모델링하고 높은 확률로 모든 간섭을 다룰 수 있는 추가 리소스 양을 계산하였다. 이를 통해 사용자가 요청한 자원을 보장받음으로써 SLA 를 무조건 만족할 수 있게 되며, 표준정규분포표에서 적당한 값을 곱함으로써 간섭도 높은 확률로 케어하면서, 자원 낭비를 최소화하는 값도 찾을 수 있다. 논문에서 제안한 모델의 적합성을 실험하기 위한 테스트베드 구축과 검증, 실험하는 과정이 향후 이루어져야 할 것이다.

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] Chi Xu et al., "Cache Contention and Application Performance Prediction for Multi-core Systems", ISPASS 2010
 [2] Younggyun Koh et al., "An Analysis of Performance Interference Effects in Virtual Environments", ISPASS 2007