

가상화 환경에서 자원 사용량을 기반으로 하는 가상머신 유형분류 기법†

강지훈*, 박봉우*, 이재학*, 이은영**

*고려대학교 컴퓨터학과

**동덕여자대학교 컴퓨터학과

e-mail:{k2j23h, bongwoo, smreodmlvl}@korea.ac.kr, elee@dongduk.ac.kr

Virtual Machine Type Classification Technique based on Resource Usage in Virtualized Environments

Jihun Kang*, BongWoo Bak*, Jaehak Lee*, Eunyoung Lee**

*Dept. of Computer Science and Engineering, Korea University

**Dept. of Computer Science, Dongduk Women's University

요 약

다수의 사용자가 자원을 공유하는 클라우드 센터에서는 자원 사용량 예측이 힘들기 때문에 지속적으로 가상머신의 자원 부족과 특정 물리 서버에 가상머신들이 집중되는 것을 방지하고자 클라우드 센터를 구성하는 다수의 물리 서버 사이에서 가상머신을 이주시키는 마이그레이션 작업을 수행한다. 가상머신 마이그레이션은 가상머신에게 할당된 자원의 규모에 따라 물리 서버에 가상머신을 균등하게 배치시킬 수 있지만 가상머신의 특정 자원에 따른 자원 집중 현상은 방지할 수 없다. 본 논문에서는 가상머신 마이그레이션 작업 시 자원 집중 현상을 방지하기 위한 기반 정보를 생성하는 가상머신 유형분류 기법을 제안한다. 본 논문에서 제안하는 가상머신 유형분류 기법은 물리 서버에서 실행 중인 가상머신의 자원 사용 정보를 기반으로 CPU, 메모리 유형으로 분류하여 가상머신 마이그레이션을 수행할 때 사용할 수 있는 기반 정보를 제공한다. 또한 본 논문에서 제안한 기법은 실험을 통해 무시할 수 있는 수준의 오버헤드를 발생시키는 것을 확인하였다.

1. 서론

클라우드 환경은 다수의 사용자가 클라우드 센터의 자원을 공유하며, 사용자는 네트워크를 통해 클라우드 센터의 자원을 사용한다. 클라우드 센터는 일반적으로 대규모 물리 서버를 클러스터 형태로 연결하여 사용하며, 사용자는 자원에 대한 제약 없이 지불한 비용만큼 자원을 사용할 수 있다. 클라우드 센터의 가용 자원량은 일반적인 개인용 PC와 비교할 수 없을 정도의 규모이지만 클러스터 형태로 연결된 각각의 물리 서버의 자원은 제한적이다. 때문에 클라우드 관리자는 클라우드 센터에서 실행 중인 가상머신들이 특정 물리 서버에 집중되지 않도록 가상머신의 개수와 할당 자원량을 기반으로 가상머신을 이주시켜 클라우드 센터를 구성하는 물리서버 사이에서 자원 불균형이 발생하지 않도록 관리한다.

하지만 가상머신의 자원 할당량이나 물리 노드의 자원 사용량과 같은 단순한 정보는 가상머신들이 CPU, 메모리와 같은 특정 자원에 집중되는 것을 방지할 수 없다. 때문에 가상머신 각각의 자원 사용 정보를 분석하고 가상머신이 작업을 수행할 때 많이 사용하는 자원을 식별하여

특정 자원에 가상머신들이 집중되어 자원 경쟁이 심화되는 문제를 해결해야할 필요가 있다.

이에 따라 본 논문에서는 가상머신의 자원 사용량 모니터링 정보를 기반으로 가상머신의 CPU, 메모리 사용 비율을 통해 가상머신을 분류하는 가상머신 유형분류 기법을 제안한다. 본 논문에서 제안하는 기법은 오픈소스 기반 가상화 플랫폼인 Xen[1]에서 개발했으며, 주기적으로 유형분류 작업을 수행하여 클라우드 환경에서 자원 관리를 위해 가상머신을 마이그레이션할 때 사용할 수 있는 기반 정보를 제공한다. 분류된 가상머신은 CPU와 메모리 사용량을 기반으로 사용 비율을 도출하고 가상머신이 주로 사용하는 자원을 식별하여 CPU와 메모리 유형으로 분류하며, 클라우드 환경에서 가상머신 마이그레이션을 수행할 때 가상머신의 유형을 기반으로 특정 자원에 가상머신들이 집중되는 것을 방지할 수 있다.

본 논문은 2장에서 가상머신들이 특정 자원에 집중되었을 때 발생하는 자원 경쟁으로 인한 성능 저하 현상을 분석하고 3장에서는 가상머신의 유형을 분류하기 위해 제안된 이전 연구를 설명하고 4장에서는 본 논문에서 제안하는 가상머신의 자원 사용량을 기반으로 하는 가상머신 유형 분류 기법과 성능 측정을 수행한다. 그리고 마지막 5장에서는 결론과 향후 연구에 대해 논한다.

† 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 이공분야기초연구사업임 (NRF-2015R1C1A2A01054813)

2. 자원 경쟁으로 인한 가상머신의 성능 저하

앞서 설명한 것과 같이 가상머신이 주로 사용하는 자원에 대한 고려 없이 가상머신을 마이그레이션하게 된다면 특정 자원에 가상머신이 집중되고 자원 경쟁이 심화되어 가상머신의 성능 저하가 발생한다. 본 장에서는 실험을 통해 가상머신이 특정 자원에 집중되었을 때 발생하는 성능 저하에 대해 분석한다. 자원 집중으로 인한 가상머신의 성능 저하를 확인하기 위한 실험 환경은 <표 1>과 같다.

<표 1> 실험 환경

분류	호스트 머신	가상 머신
CPU	Intel i5 4570 3.4 Ghz	2 VCPU
Memory	16 GB	1GB
HDD	1 TB	50 GB
OS	Ubuntu 14.04 LTS	Ubuntu 14.04 LTS
Hypervisor	Xen 4.4.1	-

본 장에서는 가상머신의 자원 집중 현상으로 인한 성능 저하를 확인하기 위해 3가지 실험을 수행한다. 첫 번째는 가상머신 6개에 동일한 CPU 집약적 작업을 실행시켜 성능을 측정하며, 두 번째는 가상머신 6개에서 동일한 메모리 집약적 작업을 실행시켜 성능을 측정한다. 마지막 세 번째는 3개의 가상머신에서는 CPU 집약적 작업을 수행하고 다른 3개에는 메모리 집약적 작업을 수행시킨다. 본 장의 실험은 가상머신들이 사용하는 자원이 CPU, 메모리에 집중된 환경과 두 가지 작업이 섞여 집중이 완화된 상태에서 가상머신의 성능을 측정하고 자원 집중으로 인해 발생하는 가상머신의 성능 저하를 확인하는 것이 목적이다. 성능 평가를 위해 Sysbench[2]를 사용하였으며, CPU 집약적 작업으로는 자연수에서 소수를 검출하는 작업을 사용하고 메모리 집약적 작업으로는 메모리 입출력 작업을 수행하는 작업을 사용한다. 실험 결과는 <표 2>와 같다.

<표 2> 자원 집중으로 인한 가상머신의 성능 저하

		CPU 집약적 가상머신만 있는 경우	메모리 집약적 가상머신만 있는 경우	가상머신 유형이 혼합된 경우
수 행 시 간 (초)	VM 1	324.5031	392.8933	202.7013 (CPU 집약)
	VM 2	325.7753	393.7775	202.6237 (CPU 집약)
	VM 3	325.8942	393.4798	202.7391 (CPU 집약)
	VM 4	325.5934	393.8911	369.5391 (메모리 집약)
	VM 5	325.2640	394.3149	368.7029 (메모리 집약)
	VM 6	325.1203	393.5632	368.2444 (메모리 집약)

<표 2>에서 보여주는 것과 같이 가상머신들이 작업을 수행할 때 가상머신이 CPU나 메모리와 같은 특정 자원에 집중된 경우 자원 사용을 위한 경쟁 상태가 심화되어 가상머신들의 성능이 저하되는 것을 확인할 수 있다. 하지만 CPU, 메모리 집약적 작업을 수행하는 가상머신들이 섞여있을 때 자원 집중 현상이 완화되어 가상머신에서 수행되는 작업의 성능이 향상된 것을 확인할 수 있다. 실험 결과에서 보여주는 것과 같이 가상머신들이 작업을 수행할 때 특정 자원에 집중되면 가상머신의 성능 저하가 발생되며, 이를 완화하기 위해서는 가상머신의 자원 사용량을 기반으로 자원 집중 현상을 최소화 할 수 있도록 가상머신을 관리해야 한다.

본 논문에서는 실험에서 보여준 것과 같이 가상머신 마이그레이션을 수행할 때 자원 집중 현상을 최소화할 수 있도록 기반 정보를 생성하는 가상머신 유형 분류 기법을 제안한다. 자세한 내용은 4장에서 설명한다.

3. 기존 연구

앞서 설명한 것과 같이 가상머신에게 할당된 자원만을 가지고 가상머신을 관리하는 단순한 방법은 자원 집중 현상으로 인한 한계가 존재한다. 때문에 가상머신의 자원 사용량이나 작업 유형에 따른 가상머신 관리 기법에 대한 다양한 연구가 진행되었다.

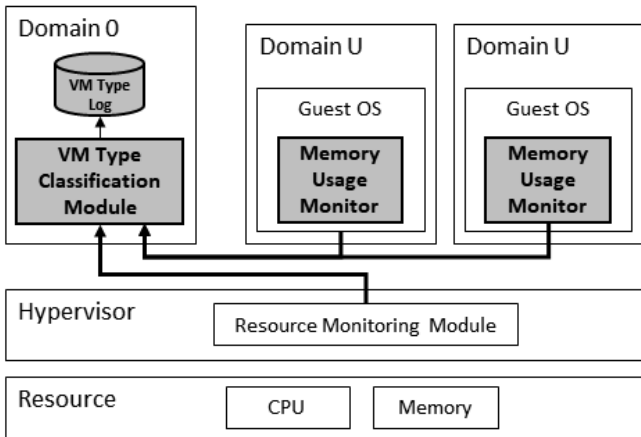
데이터 집약적 작업을 수행하는 가상머신 사이의 클러스터링 정보를 기반으로 가상머신의 지역성을 고려한 가상머신 배치 기법을 제안한 기존 연구는[3] 데이터 집약적 작업을 수행하는 가상머신 사이의 통신, 데이터 전송을 고려해 물리머신에게 배치시키기 위한 가상머신 배치 기법을 제안한다. 그리고 가상화 환경에서 I/O 작업을 수행하는 가상머신의 효율성을 향상시키기 위한 기존 연구[4]는 모니터링 정보를 통해 I/O 집약적인 가상머신을 식별하고 가상머신 스케줄링 시 유형 분류 정보를 사용한다. 또한 가상머신의 작업 유형을 고려한 가상머신 관리 기법 연구[5]는 가상머신이 수행하는 작업 유형을 분류하고 에너지 효율성을 고려한 가상머신 관리에 사용한다. 이 기법은 유형에 따라 요구되는 전력 사용량을 고려하여 전력 사용량이 많은 유형의 가상머신이 집중되지 않도록 다양한 유형의 가상머신들을 다수의 물리노드에 혼합하여 배치하는 기법을 제안한다.

이전 연구들은 가상머신의 유형에 따른 가상머신 관리 기법을 제안하고 있지만 가상머신의 유형을 미리 정의하거나 사용자에게 입력받는 방식을 사용하거나 특정 유형의 가상머신만을 고려하기 때문에 자원 사용량, 가상머신의 개수 및 가상머신에서 수행하는 작업 유형이 지속적으로 변하는 클라우드 환경에 적용하는데 제약이 따른다. 또한 가상머신 사용자에게서 입력받은 가상머신 유형 정보를 기반으로 하는 가상머신 관리는 사용자 정보에 의존하게 되기 때문에 악의적인 사용자로 인한 문제를 야기할 수 있다. 또한 각각의 가상머신이 동일한 작업만을 수행한

다는 가정을 하고 있기 때문에 현실 환경과는 동떨어져있다. 따라서 본 논문에서는 가상머신을 관리할 때 사용자 입력 정보나 가상머신의 유형을 미리 지정해야할 필요 없이 클라우드 관리자 수준에서 수집할 수 있는 정보를 사용하는 가상머신 유형 분류 기법을 제안한다.

4. 가상머신 유형 분류 모듈 구현 및 성능 평가

본 논문에서 제안하는 가상머신 유형 분류 모듈은 오픈소스 기반 가상화 플랫폼 Xen 4.4.1 버전에서 모듈 형태로 구현하였으며, 하이퍼바이저에 내장된 Xen의 가상머신 모니터링 모듈에서 측정된 모니터링 정보를 추출하여 가상머신의 개발 자원 사용량을 분석한다.



(그림 1) 가상머신 유형 분류 모듈 구조도

(그림 1)은 본 논문에서 제안하는 가상머신 유형 분류 모듈의 구조도이며, 음영 처리된 부분이 본 논문에서 새로 추가한 부분이다. (그림 1)에서 보여주는 것과 같이 본 논문에서 추가한 모듈들은 각각의 Domain(Xen에서는 가상머신에 Domain이라는 명칭을 사용함)에서 실행된다. 가상머신 유형 분류 모듈과 가상머신 유형 로그 저장소는 Xen 환경에서 가상머신들의 입출력 작업을 관리하며, 일반적인 가상머신 사용자들은 접근할 수 없는 특권 가상머신 Domain 0[6]에 존재한다. 유형 분류 모듈에서 CPU 사용량에 대한 정보는 Xen의 모니터링 모듈인 Xentop[7]을 사용해 추출한다.

하지만 Xen에서는 가상머신에게 메모리를 할당할 때 고수준 메모리 관리 기법을 사용하기 때문에 가상머신에 할당된 메모리의 전체크기는 추출할 수 있지만 가상머신이 할당받은 메모리 중에 실제 사용 중인 메모리량 정보를 추출하는 것이 불가능하다. 때문에 가상머신 내부에 메모리 사용량을 추출하여 가상머신 유형 분류 모듈에 전송하기 위한 메모리 사용량 모니터링 모듈을 추가하였다. 가상머신에 존재하는 메모리 사용량 모니터링 모듈은 가상머신의 유형을 분류하기 위해 단위 시간 동안만 모니터링하기 때문에 가상머신의 성능에 큰 영향을 미치지 않는다.

본 논문에서 제안한 가상머신 유형분류 모듈은 가상

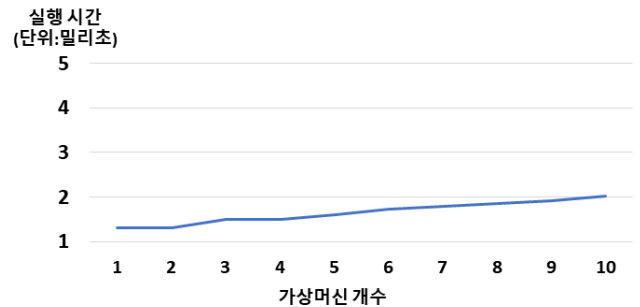
머신에 존재하는 메모리 사용량 모니터링 모듈과 Xen의 모니터링 모듈에서 추출한 CPU 사용량 정보를 기반으로 자원 사용 비율을 계산하여 가상머신의 유형을 분류한다. 또한 주기적으로 유형 분류 작업을 수행하여 가상머신이 새로 추가되거나 가상머신의 자원사용 유형이 변경되어도 지속적으로 가상머신의 유형 정보를 관리할 수 있다. 가상머신의 유형 분류를 위한 알고리즘은 가상머신 유형 분류 알고리즘과 같다.

가상머신 유형 분류 알고리즘

```

1  get vm_list ;
2  get Monitoring_Info;
3  vm_list[m] = { VM1 ... VMm };
4  cpu_usage[m][n] = { cusage1m ... cusagenm };
5  mem_usage[m][n] = { musage1m ... musagenm };
6  FOR all vm list
7    average_cpu[m] = (cusage1m + ...cusagenm)/n
8    average_mem[m] = (musage1m + ...musagenm)/n
9    total_cpu_usage += average_cpu[m];
10   total_mem_usage += average_mem[m];
11   cpu_ratio[m] = average_cpu[m]/total_cpu_usage;
12   mem_ratio[m] = average_mem[m]/total_mem_usage;
13   IF cpu_ratio[m] > mem_ratio[m]
14     vm_type[m] = cpu_vm;
15   ELSE
16     vm_type[m] = mem_vm;
    
```

본 논문에서 제안하는 가상머신 유형 분류 기법은 각 자원에 대해 단위시간동안 자원 사용량을 추출하고 모든 가상머신들의 평균 자원 사용량을 기반으로 정규화를 작업을 수행한다. 그리고 자원 사용 비율을 통해 각각의 가상머신을 CPU 혹은 메모리 유형으로 분류한다.



(그림 2) 가상머신의 개수에 따른 유형 분류 시간

(그림 2)는 물리 서버에서 실행 중인 가상머신의 개수에 따른 가상머신의 유형분류 작업 수행 시간을 보여준다. 실험 결과에서 보여주는 것과 같이 가상머신의 유형 분류를 위한 작업은 밀리초 단위의 아주 적은 시간동안 실행되기 때문에 전체 시스템에 무시할 수 있을 정도의 오버

헤드만 발생시키며, 동시에 실행되는 가상머신의 개수가 증가함에 따라 유형 분류에 대한 대상이 증가해도 유형 분류를 위한 작업 시간이 크게 발생하지 않는다. 또한 본 논문에서 제안하는 가상머신 유형 분류 모듈은 평상시 대기하고 있다 일정 시간 간격마다 유형분류 작업을 수행할 때만 작업을 수행하기 때문에 유형 분류 모듈의 추가로 인한 성능 영향을 매우 적다.

5. 결론 및 향후 연구

본 논문에서는 가상머신 마이그레이션을 수행할 때 자원 집중현상을 방지할 수 있도록 지원하는 가상머신 유형 분류 기법을 제안했다. 2장의 실험을 통해 물리 서버에서 실행 중인 가상머신들이 특정 자원에 집중된 경우 가상머신의 성능저하가 발생하는 것을 확인하였으며, 자원 집중이 완화된 환경에서는 가상머신의 성능 저하가 해소되는 것을 확인하였다. 때문에 본 논문에서는 가상머신 마이그레이션을 수행할 때 자원 집중 현상을 완화할 수 있도록 지원하기 위해 가상머신의 자원 사용량 모니터링 정보를 기반으로 가상머신들이 주로 사용하는 자원을 식별하고 유형을 분류하는 유형분류 기법을 개발하고 오픈소스 가상화 플랫폼인 Xen에서 유형분류 모듈을 구현하고 실험을 수행하였다. 실험 결과 본 논문에서 제안한 유형분류 모듈로 인한 성능 저하는 무시할 수 있을 정도의 오버헤드만을 발생시켰으며 가상머신의 개수가 증가해도 전체 시스템에 미치는 영향이 크지 않다는 것을 확인하였다.

추후 연구로는 가상머신 마이그레이션을 수행할 때 유형분류 정보를 기반으로 가상머신 관리를 할 수 있도록 본 논문에서 개발한 가상머신 유형분류 모듈을 클라우드 플랫폼인 오픈스택에 적용하고 가상머신 유형에 따른 가상머신 마이그레이션 정책을 설계하고 구현하는 것이다. 또한 가상머신 유형 분류를 위해 모니터링 정보를 수집하는 기간의 최적화를 통해 허용 가능한 오버헤드를 발생시키면서 모니터링 정보를 최대한 많이 수집하여 유형 분류 정확도를 향상시킬 계획이다.

참고문헌

- [1] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. In Proceedings of the nineteenth acm symposium on operating systems principles, SOSP '03. ACM:New York, NY, USA, 2003; 164 - 177.
- [2] Kopytov, Alexey. SysBench manual. MySQL AB, 2012.
- [3] Shabeera, T. P., Kumar, S. M., Salam, S. M., & Krishnan, K. M. (2017). Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm. Engineering Science and Technology, an International Journal, 20(2), 616-628.
- [4] Zeng, L., Wang, Y., Fan, X., & Xu, C. (2017). Raccoon: A Novel Network I/O Allocation Framework for

Workload-Aware VM Scheduling in Virtual Environments. IEEE Transactions on Parallel and Distributed Systems, 28(9), 2651-2662.

[5] Rani, Ankita, and Sateesh K. Peddoju. "A workload-aware vm placement algorithm for performance improvement and energy efficiency in OpenStack cloud." Computing, Communication and Automation (ICCCA), 2017 International Conference on. IEEE, 2017.

[6] https://wiki.xen.org/wiki/Xen_Project_Software_Overview

[7] [https://wiki.xenproject.org/wiki/Xentop\(1\)](https://wiki.xenproject.org/wiki/Xentop(1))