

대규모 계산처리 서비스를 위한 가상머신 오토-스케일링 최적화 연구*

김서영 최지은^o 김상완 김직수 노승우 황순욱 김윤희^{o1}

한국과학기술정보연구원 슈퍼컴퓨팅본부

^o숙명여자대학교 컴퓨터과학부

{ssssyy77, sangwan, jiksoo.kim, sengwoo0926, hwang}@kisti.re.kr,

^o{jieun1205,yulan}@sookmyung.ac.kr

A Study of Virtual Machine Auto-scaling Optimization for a Large-scale High Throughput Computing Service

Seoyoung Kim Jieun Choi Sangwan Kim Jik-Soo Kim Seungwoo Rho

Soonwook Hwang Yoonhee Kim

Supercomputing, KISTI

^oDept. of Computer Science, Sookmyung Women's University

요 약

본 논문에서는 클라우드 인프라 기반의 대규모 계산처리 서비스를 위한 가상 머신 오토-스케일링 최적화 방법에 대해서 논의하고 실험을 통해 그 방법에 대해 구체화 한다. 대규모 계산처리 서비스에 활용되는 통합 인프라 중 클라우드 인프라의 경우 오토-스케일링을 통해 동적인 자원 확장이 가능하다. 본 서비스를 활용하는 계산 작업들은 각기 다른 특성을 갖고 있으며 이러한 특성을 반영하여 가상자원의 오토-스케일링 정책에 활용할 경우 성능 최적화에 도움이 될 수 있다. 본 논문에서는 KISTI에서 연구 및 개발한 클라우드 인프라 기반의 HTCaaS를 통해 실험을 수행하였고 그 결과를 분석하였다.

1. 서 론

계산과학 분야의 다양한 과학 응용들은 많은 수의 작업들로 구성이 되어있으며, 이를 지원하기 위한 다양한 종류의 계산 자원들을 연계/통합하는 기술의 중요성이 점점 커져가고 있다. 이에 따라 KISTI에서는 2010년부터 국가차원의 대규모 계산 처리 시스템인 HTCaaS(HTC-as-a-Service[1])을 연구/개발하기 시작하였고 국가슈퍼컴퓨팅 자원뿐만 아니라 로컬 클러스터(Cluster), 그리드(Grid), 클라우드(Cloud)에 아우르는 이기종 자원들을 통합하여 효율적으로 활용 가능하게 함으로써 국내외 계산과학분야 사용자들을 지원하고 있다. 최근 증가하는 사용자들을 지원하기 위해 서비스 부하에 따라 유연한 자원 확장이 가능한 클라우드 환경의 활용이 증대되는 추세이다.

클라우드 인프라를 활용하는 데에 있어서 가장 큰 장점은 필요한 양 만큼의 자원을 사용하는 것이 가능하다는 점과, 자원의 수요 증가시 오토-스케일링(Auto-Scaling)을 통해 요구되는 자원을 확장함으로써 전반적인 자원 사용 비용을 줄일 수 있다는 점이다. 또한 자원 가상화를 통해 한정된 물리적 자원을 가변적으로 늘려 확장 사용(오버 프로비저닝, Over-provisioning) 가능하므로 한정된 자원에 대한 효율적인

활용을 할 수 있으며, 처리량(Throughput) 증가에 도움이 될 수 있다. 클라우드의 사용이 자원 활용의 유연성을 제공하는 반면, 여전히 성능 저하라는 한계점을 갖는다[2]. 특히 오버프로비저닝의 경우, 실제 물리 CPU, 메모리, 네트워크 등의 성능을 분할하여 사용하게 되므로 적정 임계치를 초과하면 오히려 성능 저하의 결과를 초래한다. 현재 HTCaaS에서 사용하고 있는 기본 Auto-Scaling 정책은 최대 서브 작업(테스크) 수에 따라 전체 계산 자원의 코어 수(pCPU)를 초과시 최대 가용 가상CPU(vCPU)까지 오버 프로비저닝하는 것이다. 이러한 기본 정책은 전반적인 작업 처리량 증가에 도움이 되지만, 때때로 수행되는 작업 특성에 따라 성능 저하를 야기한다.

본 연구에서는 실제 다양한 계산과학 분야의 응용 연구자들을 위해 서비스를 하고 있는 대규모 계산처리 시스템의 클라우드 인프라 활용에 있어서 오토-스케일링 수행 시 성능을 최적화 하기 위한 방법을 논의하고 실험 결과 분석을 통해 구체화 한다.

2. 관련 연구

2.1 오토-스케일링

오토-스케일링이란 클라우드 컴퓨팅상에서 서비스의 부하량과 사용량에 맞게 탄력적으로 컴퓨팅 자원을 늘렸다가(Scale-out) 줄이는(Scale-in) 기능을 의미하며, 기존의 인프라가 갖지 못한 클라우드 인프라의 큰 장점 중 하나이다. 가장 대표적인 오토-스케일링

¹ 교신저자

*이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A3007866)

서비스로는 아마존의 오토-스케일링[3] 서비스와 Right-Scale[4] 등이 존재하며, 일반적으로는 스케줄 또는 규칙기반의 오토-스케일링으로 분류될 수 있다. 이들은 기본적으로 CPU 및 디스크의 사용량과 같은 하드웨어 성능 수치를 기준으로 제공된다. 스케줄 기반의 오토-스케일링은 사용자가 주어진 시간에 용량(capacity)을 늘리거나 줄일 수 있도록 하며, 가령 “매일 오전10시부터 오후7시까지 20개의 인스턴스를 실행하고 그 외의 시간에는 10개의 인스턴스를 실행”과 같은 요구가 이에 해당한다. 규칙 기반의 오토-스케일링은 사용자가 인스턴스 조절에 관한 일련의 규칙을 정의함으로써 이루어지는데, 예를 들어, “평균 CPU 사용률이 5분동안 10%이하(또는 80%이상)인 경우 5개 인스턴스를 삭제(증가)”와 같이 정의될 수 있다. 아마존(AWS, Amazon Web Service)의 오토-스케일링 서비스[3]의 경우는 사용자가 정의한 규칙을 통해 자원의 규모를 확장하고 축소하는 규칙 기반의 오토-스케일링 서비스를 제공하며, RightScale[4]은 두 종류 모두 서비스하고 있으며 이에 추가적으로 스케일링에 도움이 되는 메트릭(Metric) 수치를 제공함으로써 오토-스케일링이 더욱 용이하게 이루어 질 수 있도록 서비스를 제공하고 있다.

2.2 HTCaaS

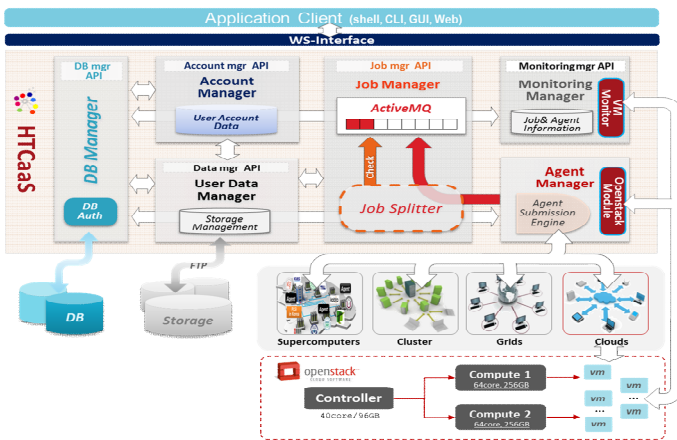


그림 1. HTCaaS-Openstack 연결 구조

HTCaaS(High Throughput Computing-as-a-Service) [1]는 파일럿 작업(Pilot-Job)[5]기반의 분산 자원관리 시스템으로 대규모의 독립적인 작업들로 이루어진 다양한 과학 응용들을 효율적으로 지원하고, 분산되어 있는 이기종의 다양한 컴퓨팅 자원들을 효율적으로 통합하고 관리한다. 대규모 작업 수행을 위한 메타 작업(Meta-Job, 서브작업들의 집합)과 작업기술 언어로 JSDL(Job Submission Description Language)[6]을 지원하고 있으며 이러한 메타 작업을 통해 파라미터 스위프(Parameter Sweep)과 같이 많은 수의 작업들로 구성되는 응용들을 보다 효율적으로 지원할 수 있다. 작업 수행의 자세한 과정은 [1]에 설명되어 있다. 그림 1은 HTCaaS 모듈의 전체 구조를 나타내며, 최근 확장된 오픈스택(Openstack[7]) 모듈이 포함된 모습을

보인다. 모니터링 매니저 (Monitoring Manager) 에서 가상 머신의 상태 정보 및 자원의 사용량을 관리하며, 이를 기반으로 에이전트 매니저(Agent Manager)에서는 가상 머신의 배치를 결정/담당한다.

현재 HTCaaS에서의 기본 Auto-Scaling 정책은 총 서버 작업(테스크) 수를 기반으로 결정되고 있으며, 그 수가 전체 계산 자원의 코어 수($pCPU$)를 초과시 최대 가용한 가상CPU($vCPU$)까지 오버 프로비저닝 하여 서비스를 제공하고 있다. 실제 서비스에서 이러한 정책이 전반적인 작업 처리량 증가에 도움이 되고 있으나, 때때로 작업 특성에 따라 성능저하가 극대화되기도 한다. 따라서 본 연구에서는 오버프로비저닝 된 자원 상에서 응용특성에 따라 작업을 분류하여 수행 후 그 성능을 분석하고자 하며 추후 이를 오토-스케일링 에 활용하고자 한다.

3. 제안 방법 및 실험

HTCaaS에서는 주로 신약개발, 고에너지 물리, 핵물리, 고체물리, 천체 물리 분야의 응용들을 대상으로 서비스를 제공하고 있다. 이러한 분야들의 응용은 크게 “CPU중심”, “I/O중심” 으로 분류할 수 있으며, 실험을 위해 Synthetic작업을 구성하여 사용하였으며, 그림 1에 보이는 바와 같이 HTCaaS를 통해 Openstack기반 자원으로의 작업제출 및 실행이 이루어진다. 실험환경의 정보는 표 1과 같으며, 실질적으로 계산에는 Compute1, Compute2(총 128코어)가 사용되었다.

표 1. 클라우드 환경 자원 정보

호스트	CPU	Core	RAM
Controller	Intel(R)Xeon(R) CPU E5-2600 v2 @2.20GHz	40	96GB
Compute1	AMD Opteron(TM) Processor 6378	64	256GB
Compute2	AMD Opteron(TM) Processor 6378	64	256GB

실험은 두 가지 특성을 갖는 작업들을 대상으로 1~5까지의 $vCPU/pCPU$ 비율로 오버프로비저닝한 가상 자원 상에서 수행 후 성능 결과를 비교하였으며, 각각 2000개의 서브작업을 갖는 메타작업을 $vCPU/pCPU$ 비율에 따라 총 5회씩 수행한 후 평균치를 계산하였다. 그림 2는 CPU중심 응용의 작업들의 평균 makespan의 결과이며, 그림 3은 I/O중심 응용의 작업에 대한 평균 makespan을 그래프로 도식화한 것이다. CPU중심 응용의 작업을 수행하는 경우, $vCPU/pCPU$ 비율이 3배가 되는 시점부터 평균 make span 이 급격히 증가하는 경향이 있으나 $vCPU/pCPU$ 비율이 증가할수록 다시 감소하는 경향을 보였다. 즉, $vCPU/pCPU$ 의 비율 증가에 따른 전반적인 처리량(Throughput) 증가 결과를 보였다. 하지만 I/O중심 응용 작업의 경우, $vCPU/pCPU$ 비율이 증가할수록 평균

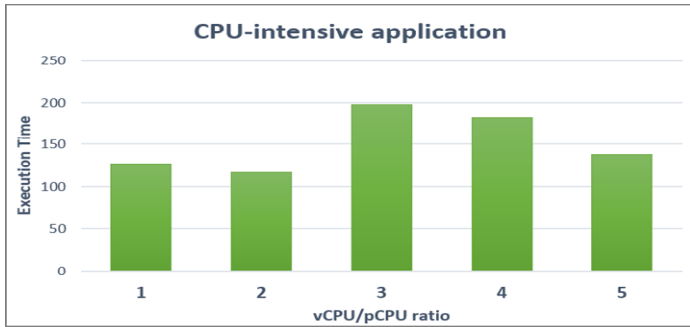


그림 2. CPU중심 응용 수행 결과

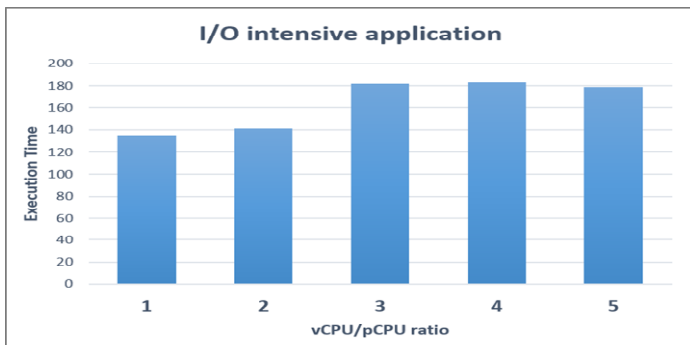


그림 3. I/O중심 응용 수행 결과

vCPU/pCPU의 비율 증가에 처리량이 감소하는 결과를 보임을 알 수 있다. (Throughput) 증가 결과를 보였다. makespan이 증가하거나 유지되는 경향을 보였다.

하지만 I/O중심 응용 작업의 경우, vCPU/pCPU 비율이 증가할수록 평균 makespan이 증가하거나 유지되는 경향을 보였다. vCPU/pCPU의 비율 증가에 처리량이 감소하는 결과를 보임을 알 수 있다.

그림 4는 오버 프로비저닝 시 발생한 가상 머신 오류 발생률을 조사한 것이다. 가상 머신이 'Error'가 되며, 작업의 재실행이 이루어진다. I/O중심 응용 수행 시에는 가상 머신 오류가 발생하지 않았고, CPU중심 응용 수행 시에는 평균 9~21회 정도의 오류가 발생하였다. 이는 하이퍼바이저(KVM) 단의 오류로 예측되며 추후 수정이 가능한 것으로 판단된다. CPU중심의 응용 수행에서 가상 머신 오류 발생이 감소한다면 평균 makespan 감소에 도움이 될 수 있으며 처리량 증가의 결과를 이끌 수 있다.

본 연구에서 수행된 결과를 통해, 하나의 노드에서 많은 수의 vCPU를 활용하기 위해서는 CPU-중심의 작업들과 I/O-중심의 작업들을 효율적으로 함께 스케줄링하거나 동적인 오토-스케일링 정책을 적용해야 함을 예상할 수 있다.

4. 결론 및 향후 계획

본 논문에서는 대규모 계산처리 서비스인 HTCaaS를 위한 가상 머신 오토-스케일링 최적화 방법에 대하여 논의하였고 실험을 통해 구체화하였다. HTCaaS에서 주로 수행되는 응용들을 특징에 따라 분류하여 오토-스케일링 시 활용되는 오버 프로비저닝에서의 성능을

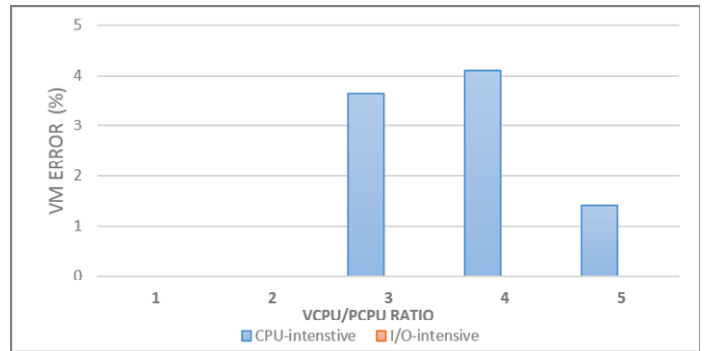


그림 4. 가상 머신 오류발생률 비교

비교/분석 하였다. 많은 작업으로 이루어진 대규모 과학 응용의 효율적인 지원을 위해서는 가용한 자원을 최대한 활용해야 하며, 이러한 관점에서 물리적으로 가용한 CPU들을 최대한 오버 프로비저닝 하는 것이 유용한 정책이 될 수 있다. 그러나, 본 논문에서 수행한 실험 결과에서 볼 수 있듯이, I/O 중심의 응용을 실행시 단일 노드당 그만큼 프로비저닝 할 수 있는 가상CPU의 수를 줄일 수 밖에 없음을 확인할 수 있다. 이는 곧, 하나의 노드에서 많은 수의 vCPU를 활용하기 위해서는 CPU중심의 작업들과 I/O 중심의 작업들을 효율적으로 함께 스케줄링해야 한다는 점을 시사하고 있다.

향후에는 기존연구[8]과 관련하여 응용 분류에 대한 프로파일링 방법을 적용 후 오토-스케일링 최적화를 위한 실험을 추가로 수행할 계획이다.

5. 참고문헌

- [1] S. Rho, S. Kim, S. Kim, S. Kim, J.-S. Kim, S. Hwang, "HTCaaS: A Large-Scale High-Throughput Computing by Leveraging Grids, Supercomputers and Cloud," IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC12), Nov. 2012.
- [2] N. Regola, J.-C. Ducom, "Recommendations for Virtualization Technologies in High Performance Computing," Cloud Computing Technology and Science(CloudCom), 2010 IEEE Second International Conference on, 30 2010-Dec., pp. 409-416.
- [3] Amazon Auto-Scaling Service: <http://aws.amazon.com/autoscaling/>
- [4] RightScale. <http://rightscale.com>
- [5] Pilot-Job, https://en.wikipedia.org/wiki/Pilot_job
- [6] A. Anjomshoa et al., "Job Submission Description Language (JSDL) Specification, "Version 1.0. Open Grid Forum Grid Final Document Nr. 136, 2008.
- [7] Openstack, <http://www.openstack.org>
- [8] S. Kim, J.-S. Kim, S. Hwang, Y. Kim, "Towards effective science cloud provisioning for a large-scale high-throughput computing," Cluster Computing 17(4): 1157-1169 (2014)