

# 하이브리드 클라우드에서의 SLA 에 따른 적응형 클라우드 작업 관리

고정인<sup>o</sup>, 강혜정, 김윤희

숙명여자대학교 컴퓨터학과

{jungin, hjkang, yulan}@sookmyung.ac.kr

## 요 약

최근 과학 연구 분야에서 대규모 처리를 위한 컴퓨팅 수요가 급증하면서 다양한 컴퓨팅 자원 관리에 대한 연구가 진행되고 있다. 특히, 그리드 사설/상용 클라우드를 포함하는 하이브리드 클라우드 자원의 경우, 각 자원에 대한 이용 효율 향상을 위해 각 자원의 특성을 고려할 필요가 있다. 본 논문에서는 주어진 데드라인 안에 응용 수행을 보장하되 자원의 이용 효율을 높일 수 있도록 하이브리드 클라우드 자원을 이용한 온디맨드 자원 할당에 기반을 둔 적응형 작업 할당 기법을 제안하고자 한다. 또한 정책별 작업 할당 기법을 제공함으로써 사용자의 SLA(Service Level Agreement)를 만족시키고자 한다. 마지막으로 실험을 통하여 제안하는 기법의 우수성을 검증하였다.

## 1. 서 론

정보화 사회가 고도화됨에 따라 각종 IT 계산자원의 수요가 급격히 증가하면서 더욱 높은 계산능력을 가진 자원을 확보하려는 노력이 증가하고 있다. 하지만 현대 응용의 동적인 작업 부하로 인해 응용 수행에 필요한 계산 자원의 수요가 가변적이고 이에 따라 필요한 양을 유동적으로 제공해야 할 필요가 있다. 그리드 자원의 경우 적절한 양을 탄력적으로 서비스하기에는 어려움이 따른다. 반면, IT 분야에서 새로운 패러다임으로 떠오르는 클라우드 컴퓨팅은 각종 컴퓨팅 자원들을 가상화 기술을 통해 필요한 양만큼만 자원을 제공하는 것이 가능하여 기존 계산 자원들과는 달리 자원 사용의 유연성을 제공한다. 그러나 가상화는 성능면에서 추가적인 오버헤드가 발생할 수 있으므로, 각 분산 자원의 서로 다른 특성을 파악하고 높은 계산 성능을 위해 다양한 분산 컴퓨팅 자원을 포함하는 하이브리드 클라우드에 관한 효율적인 자원 관리 및 작업 할당 기법이 반드시 필요하다. 특히, Amazon EC2[1]와 같은 상용 클라우드(Public Cloud) 컴퓨팅의 경우 사용 시간에 비례하여 비용이 증가하기 때문에 자원 사용에 따른 비용 발생 여부 및 비용 발생량이 핵심적인 고려사항이 될 수 있다.

본 논문에서는 이기종 자원을 효율적으로 활용하기 위한 적응형 작업 할당 기법을 제안한다. 다양한 이기종 자원으로 인해 복잡해진 사용자의 응용 수행 요구(SLA)를 충족시키기 위해 계산 과학 응용의 특성과 자원의 속성을 고려한 성능 및 비용 정책을 수립하고, 정책을 수용할 수 있는 데드라인 기반의 동적 작업 할당 기법을 제안한다. 전산 유체역학 분야(CFD; Computational Fluid Dynamics)[4]의 응용을 대상으로 제안한 기법의 성능 측정 및 비교, 분석하는 실험을 수행하여 제안하는 기법의 우수성을 검증한다.

서론에 이어 2 장에서는 관련 연구들을 살펴보고, 3 장에서는 본 논문에서 제안하는 정책 기반의 적응형 작업 할당 기법에 대하여 설명한다. 4 장에서는 실험을 통해 제안하는 기법을 검증하고, 마지막으로 5 장에서 결론을 도출한다.

## 2. 관련 연구

여러 가지 자원들을 복합적으로 이용할 때에 다양한 정책들을 고려하여 작업들을 효율적으로 클라우드 자원에 할당 하기 위한 연구가 활발하게 진행되고 있다. Bossche R, et al.[2], Aneka[3], Comet Cloud[4]와 같은 연구들은 기본적으로 데드라인을 기준으로 그리드 및 클라우드 자원에 대한 작업 할당 기법을 연구한다. Bossche R, et al 은 계산 비용 및 데이터 처리 비용을 모두 고려하여 자원을 선택하는 방식을 제안하였고, Aneka 의 경우, 작업들의 평균 실행 시간과 데드라인을 고려한 동적 자원 프로비저닝 서비스를 연구하였다. Comet Cloud 는 응용과 인프라 구조를 계층별로 구분하여 주어진 예산 및 데드라인의 우선순위를 통한 최적의 자원 통합 방법을 제안하였다. 하지만 상기한 연구들은 응용의 특성을 고려하지 않거나, 작업 수행 시간을 데드라인에 맞추면서 동시에 비용은 고려하지 않는 한계가 존재한다.

본 논문에서 제안하는 작업 할당 기법은 응용 수행 이력을 바탕으로 한 데드라인 기준을 제시하고, 해당 데드라인 내에서 비용을 동시에 고려하여 자원을 효율적으로 활용할 수 있도록 한다.

## 3. 정책별 적응형 작업 할당 기법

본 논문에서는 하이브리드 클라우드를 활용한 적응형 작업 할당 기법에 대해 모델링 하였다. 일반적으로 작업 수행에 대한 예측이 어렵기 때문에 초

기 작업 할당에 차질이 발생할 수 있으므로, 주기적인 모니터링을 통해 작업의 지연을 검사하고 지연 발생 시에 데드라인 위반 여부를 확인하여 대기 중인 작업을 재할당할 필요가 있다. 이 때, 재할당되는 작업들은 데드라인 안에 수행을 마칠 수 있는 방향으로 적정 자원을 할당 받는데, 필요에 따라 상용 클라우드에 관한 추가 확장이 가능하다. 그림 1은 적응형 작업 할당 기법의 수행 시나리오를 나타낸다. 각각의 세로 열은 하나의 자원에 대응된다. G1, G2 는 그리드 자원, P1, P2 는 사설 클라우드에 해당하는 비상용 자원들을 나타낸다. (a)는 상기한 자원들에 대한 초기 작업 할당 결과로서 각 열에 나타난 숫자는 작업 번호를 의미하고, 길이는 작업 수행 시간을 의미한다. (b)는 작업 지연 발생으로 인한 잠재적 응용의 데드라인 위반을 예방하기 위한 적응형 작업 할당 기법을 보인다. 제안하는 적응형 작업 할당 기법은 응용의 수행 시간이 데드라인을 초과하지 않도록 작업을 자원에 재할당한다. 또한 기존 자원만으로 응용 수행이 어려울 경우, 온디맨드 서비스를 제공하는 상용 클라우드를 이용한다.

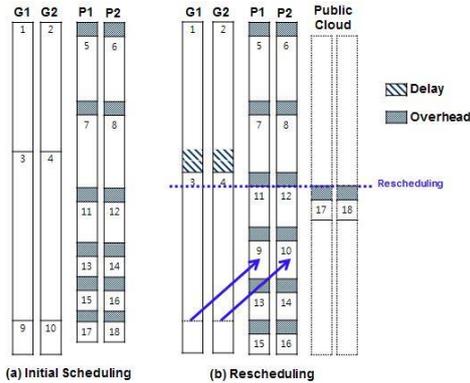


그림 1. 적응형 작업 할당 과정

제안하는 작업 할당 서비스는 성능 또는 비용 정책에 기반을 둔다. 두 정책 모두 사용자가 원하는 데드라인을 응용 수행의 척도로 하여, 데드라인 내에서 정책 기준을 만족하도록 작업을 할당한다. 성능 지향 정책의 경우 데드라인 내에 응용 수행을 완료하되 상용 클라우드를 포함한 보유 중인 자원을 최대한 활용하여 작업 수행 시간을 단축하는 성능 향상을 목표로 작업을 할당한다. 비용 지향 정책은 이와 반대로 데드라인 내의 성공을 보장하되 상용 클라우드 소비에 따른 비용을 최소화하기 위한 작업 할당 기법이다. 제안하는 기법에서 상용 클라우드 인스턴스의 타입에 대한 가격 대비 성능을 *Efficiency* 라하고, 높은 *Efficiency* 를 우선 택하는 방식으로 작업 할당이 수행된다. *Efficiency* 를 구하는 수식은 다음과 같다. (\*MIPS: the number of Million Instruction per Second)

$$Efficiency = \frac{(MIPS \times \text{Number of Cores})}{\text{Price per hour}}$$

#### 4. 성능 평가 및 분석

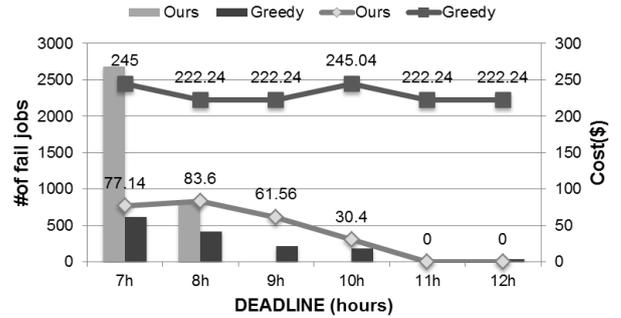


그림 2. Greedy 알고리즘과의 데드라인 위반율 및 비용 비교

그림 2는 데드라인에 따른 실패한 응용 작업 개수 및 처리 비용에 대하여 Greedy 알고리즘과 본 연구의 작업 할당 기법을 비교한 결과이다. Greedy 알고리즘은 처리비용에 관련 없이 수행 자원들 중 성능이 가장 좋은 자원에 우선적으로 작업 할당하는 것을 의미한다. 총 6 개 데드라인에서 Greedy 알고리즘은 약 33%(6 번 중 2 번)의 성공률을 보였다면 제안하는 작업 할당 기법은 상대적으로 높은 약 67%(6 번 중 4 번 성공)의 성공률을 보였다. 또한 비용 측면에서 Greedy 알고리즘의 경우, 데드라인의 변화에 크게 영향을 받지 않고, 시종일관 높은 가격을 책정하는데 반해 제안하는 기법은 비용이 점차 감소하는 경향을 보인다.

#### 5. 결론

본 논문에서는 주어진 데드라인 안에서 응용을 수행 시, 다중 인프라 자원을 고려해 자원의 효율성을 증대시키고, 유연한 작업 할당 기법을 통해 사용자의 SLA 를 만족시키는 작업 관리 서비스를 제안하였다. 향후 연구과제로는 과학 워크플로우의 파이프라인, 병렬, 하이브리드 등의 다양한 응용 모델에 대한 확장과 저전력 정책 등 추가 정책 수립을 포함하는 작업 할당 기법으로 확장할 계획이다.

#### 6. 참고 문헌

- [1] Amazon Elastic compute cloud, <http://aws.amazon.com/ec2>
- [2] Bossche R., Vanmechelen K., and Broeckhove J., "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds," *3rd IEEE International Conference on Cloud Computing technology and Science*, pp. 320-327, 2011.3
- [3] Vecchiola C., Chu X., Mattess M., and Buyya R., "Aneka-a software platform for .NET-based cloud computing." *In Proceeding of High Performance and Large Scale Computing*, IOS Press, pp. 267-295, 2009.
- [4] Kim, Hyunjoo, et al. "An autonomic approach to integrated hpc grid and cloud usage," *e-Science'09 Fifth IEEE International Conference on IEEE*, 2009.

#### 7. 기타

“본 연구는 한국과학기술정보연구원의 국가슈퍼컴퓨팅 서비스 기반 강화의 연구결과로 수행되었음” (K-13-L01-C01)