클라우드 서비스를 위한 최적 부하할당 기법

한영주*, 이요한**, 윤찬현**
*한국과학기술원 정보통신공학과
**한국과학기술원 전기및전자공학과
e-mail : y.han@kaist.ac.kr

Optimal Load Allocation Scheme for Cloud Service

Youngjoo Han*, Yohan Lee**, Chan-Hyun Youn**
**Dept. of Info. and Comm. Engineering, Korea Advanced Institute of Science and Technology
*Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology

요 약

이 기종의 컴퓨팅 자원들로 구성된 데이터센터에서 사용자가 제출한 클라우드 작업의 서비스 품질을 보장하기 위해서는 효율적인 작업할당 기법이 필요하다. 본 논문에서는 작업의 처리시간을 사용자의 QoS 로 정의하고 작업처리시간을 최소화하면서 동시에 모든 사용자에게 차별 없이 동일한 수준의 서비스를 제공할 수 있는 최적 작업할당 문제를 정의한다. 협력적 게임모형인 NBS(Nah Bargaining Solution)을 이용하여 정의한 문제를 해결하였고 이를 이용하여 각 자원에 할당되는 작업 부하를 효율적으로 제어함으로써 파레토 최적을 유지할 수 있는 협력적 부하분산 정책을 도출하였다. 또한, 성능 평가를 통해 제안한 기법이 작업처리시간을 최소화하면서도 모든 자원에서의 작업 처리시간이 동일함을 보였다.

1. 서론

클라우드 서비스가 상업적으로 성공하기 위해서는 클라우드를 이용하는 사용자에게 품질이 보장된 서비스를 제공하여야 한다. 궁극적으로 사용자의 QoS 로서 일정한 수준의 작업처리시간(응답시간)을 보장할 수 있어야 한다. 또한, 사용자에 대한 차별 없이 동일한 수준의 서비스를 제공하여야 한다. 따라서 클라우드 서비스를 제공하는 데이터센터에서는 사용자의 QoS 를 보장하면서 공정성을 보장할 수 있는 효율적인 작업할당 기법이 필요하다.

이 기종의 다양한 컴퓨팅 자원들로 구성된 데이터센터에서 최적 기법[1]에 근거하여 일부 자원만을 사용하여 작업을 처리하였을 때가 전체 자원을 이용하였지만 비효율적인 작업 할당을 하였을 경우보다 더 우월한 성능을 야기할 수 있다. 이는 최적의 작업처리시간이 작업부하(workload)의 효율적인 분배에 중요하게 의존한다는 것을 의미한다. 즉, 사용자가 QoS 를 만족하는 범위 내에서 작업을 수행하기 위해 각 자원의 성능과 작업부하의 균형(load balancing)을 고려한 작업할당이 필요하다.

지금까지 분산환경에서 자원의 효율적 관리를 위한 많은 연구들이 선행되었다. 그러나, 일반적으로 분산환경에서 자원 스케줄링이나 자원의 할당과 같은 대부분의 자원관리 기술은 사용자들에게 자원에 대해 공정하고 공평한 접근을 제공하는데 한계가 있음이 많은 연구자들에 의해서 밝혀졌다[2]. 또한, 이런 기술들은 QoS 제약조건과 같은 더욱 정교한 사용자의 요구 사항을 실현하는데 한계가 있다. 최근에는 이런 한계를 극복하고자 제한된 자원에 대한 다중 사용자의 협력적, 비협력적 경쟁 상황을 고려하기 위한 게임이론이 도입되고 있다.

게임 이론에서는 참여자가 시스템 내에서 얻을 수 있는 효용을 정량적으로 판단할 수 있고, 게임에 참여하는 모든 참가자들이 각자 자신의 효용함수에 근거하여 합리적인 의사결정을 한다고 가정한다. 게임이론에서 참여자의 효용을 고려하여 자원을 배분하고자 하는 협력적 게임모형인 Bargaining game 의 대표적인 해법으로 Nash Bargaining Solution(NBS)이 있다. NBS 는 사용자의 곱으로 표현되는 시스템 전체 효용을 최대화하는 방향으로 자원을 할당한다. 또한, NBS 는 크게 1) 파레토 최적(Pareto optimality), 2) 대칭성(symmetry), 3) 규모의 불변성(invariance), 4) 해와 무관한 대안의 독립성(independence of irrelevant alternatives) 이라는 네 가지 공리를 보장한다.

본 논문에서는 클라우드 서비스를 이용하는 사용자로부터 제출된 작업의 평균 작업처리시간을 최소화하고 각 사용자의 작업에 대한 공정성을 보장하기 위한 작업 할당 정책을 제안한다. 자신에게 할당된 작업의 평균 서비스 시간을 최소화하고자 경쟁 관계에 있는 자원들은 제한된 시스템 자원 상황 하에서 자신의 효용을 극대화하려는 경향을 가지며 이것은 게임이론 기반 접근법을 이용하여 해결할 수 있다. 본 논문에서는 NBS 를 통해 각 자원에 할당되는 작업부하를 효율적으로 제어함으로써 작업처리시간을 최소화하면서 공정성을 보장하는 파레토 최적을 유지할 수 있는 협력적 부하 배분 정책을 도출한다.

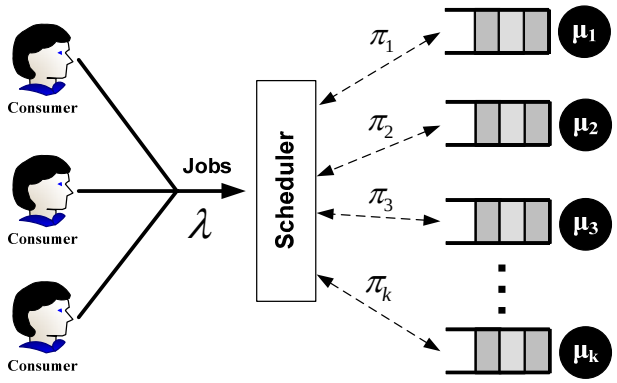

그림 1 System Model

2. 협력적 부하분산 정책

그림 1 에 도시된 시스템 모델은 사용자(consumer), 스케줄러, 컴퓨팅 자원으로 구성되어 있다. 각 사용자는 다른 사용자와 독립적으로 평균 λ_i (jobs per second) 비율로 Poisson 분포에 따라 작업을 생성한다. 사용자들이 생성한 작업은 데이터센터에 도착하고 도착한 전체 작업은 평균 λ (jobs per second)이다. 스케줄러는 QoS 를 보장하도록 사용자로부터 제출된 작업을 각 자원에 할당한다. 스케줄러는 사용자로부터 제출된 전체 작업을 자원 j 에 π_j (jobs per second)로 할당한다. 또한, 컴퓨팅 자원은 가상머신 또는 실제 물리적 자원이 될 수 있으며 자원 j 는 평균 μ_j 비율로 작업을 처리한다. 각 자원은 1 개의 큐를 가지고 있으며, 선입선출(FCFS) 정책으로 작업을 처리한다. 추가적으로, 본 논문에서는 작업처리시간을 특정한 분포로 가정하지 않고 일반분포를 가정한다. '일반분포'라 하면 어떤 특정한 분포가 아닌 모든 형태의 분포를 의미한다. 따라서 각 자원은 M/G/1 큐잉 시스템으로 모델링 될 수 있다. 큐에서의 대기시간과 서비스 시간의 합을 작업처리시간으로 정의할 때, 각 자원에서의 평균 작업처리시간(T_j)은 다음과 같다[3].

$$T_j = \overline{s_j} + \frac{\overline{s_j^2}\cdot\pi_j}{2(1-\overline{s_j}\cdot\pi_j)} \tag{1}$$

상기 식에서 $\overline{s_j} = 1/\mu_j$, $\overline{s_j^2}$ 은 2 차 모멘트이다. 따라서, 본 논문에서 해결하려는 문제는 공정성을 보장하면서 모든 자원의 평균 작업처리 시간을 최소화하는 것이다. 수학적으로는 아래와 같이 정의할 수 있다.

$$minimize \quad \left(\overline{s_j} + \frac{\overline{s_j^2}\cdot\pi_j}{2(1-\overline{s_j}\cdot\pi_j)} \right), \ \forall j \tag{2}$$

또한, 다음의 제약조건을 만족해야 한다.

$$\lambda < \sum_{j=1}^{m} \mu_j, \ j = 1,2,...,m; \ i = 1,2,...,n, \tag{3}$$

$$\pi_j \le \mu_j, \forall j, \tag{4}$$

$$0 \le \pi_j, \forall j, \tag{5}$$

$$\lambda = \sum_{j=1}^{m} \pi_j, \forall i, \forall j, \tag{6}$$

제약식 (3), (4), (5), (6)은 안정조건을 의미하며, 안정조건들이 보장되지 않는다면 각 자원의 큐에 대기하는 작업들은 무한히 증가하게 된다. 상기에 제시된 수식 (2)는 각 자원의 목적 함수를 의미하고 제약조건 내에서 convex 이고 아래로 유계이다. 그러므로 본 문제는 해결이 가능한 문제임을 알 수 있다. 앞에서 언급한 것처럼 본 논문에서는 상기의 문제를 협력적 게임이론을 기반으로 해결하며 이때, 각 자원은 게임에 참여하는 player 가 된다. 협력적 게임에서는 모든 player 가 최소한의 성능(Initial performance)을 얻지 못 할 경우에는 협력이 이루어지지 않으므로 모든 자원은 최소 성능보다 최소한 같거나 우수한 성능을 얻을 수 있어야 한다. QoS 를 보장하는 최대 작업처리시간이 최소한의 수용 가능한 성능이 되고 협력적 게임의 합의점이 된다. 따라서, 자원 j 가 QoS 를 보장할 수 있는 최대 도착률을 $\pi_j^{\max}$ 라 할 때, 최소한의 성능 합의점이 될 수 있는 최대 작업처리시간(T_j^0)은 다음과 같이 정의될 수 있다.

$$T_j^0 = \overline{s_j} + \frac{\overline{s_j^2}\cdot\pi_j^{\max}}{2(1-\overline{s_j}\cdot\pi_j^{\max})} \tag{7}$$

위의 식으로부터 $\pi_j^{\max}$ 는 다음과 같이 도출된다.

$$\pi_j^{\max} = \frac{2(T_j^0 - \overline{s_j})}{2\overline{s_j}(T_j^0 - \overline{s_j}) + \overline{s_j^2}} \tag{8}$$

수식 (1)의 최적화 문제를 해결하기 위한 NBS 는 아래 최적화 문제를 해결함으로써 찾을 수 있다[4].

$$max \prod_{j=1}^{m} \left(T_j^0 - \overline{s_j} - \frac{\overline{s_j^2}\cdot\pi_j}{2(1-\overline{s_j}\cdot\pi_j)} \right) \tag{9}$$

subject to (3), (4), (5), (6), 그리고

$$\pi_j \le \pi_j^{\max} \tag{10}$$

Theorem 1. 작업 처리시간을 최소화하기 위한 최적화 문제인 수식 (9)의 결과는 다음과 같다.

$$\pi_j = \frac{1}{2\overline{s_j}} + \frac{\pi_j^{\max}}{2} - \frac{\sqrt{(\overline{s_j}\pi_j^{\max}-1)(\overline{s_j}(\gamma\pi_j^{\max}-4)-\gamma)}}{2\sqrt{\gamma\overline{s_j}}}. \tag{11}$$

Proof. T_j^0 를 $\pi_j^{\max}$ 를 이용하여 정의하고, 수식 (9)를 NBS 의 특성을 기반으로 Problem transform 을 하면 수식(9)는 다음과 같이 정의될 수 있다 [4].

$$max \sum_{j=1}^{m} \ln \left(\frac{\overline{s_j^2}\cdot\pi_j^{\max}}{2(1-\overline{s_j}\cdot\pi_j^{\max})} - \frac{\overline{s_j^2}\cdot\pi_j}{2(1-\overline{s_j}\cdot\pi_j)} \right) \tag{12}$$

비선형 최적화 문제인 수식 (12)는 선형 제약조건을 가지며 π_j 에 대하여 concave 함수이다. 따라서, Karush-Kuhn-Tucker(KKT) 조건이 최적화를 위한 필요충분 조건이 되며, Lagrangian 을 이용하여 문제의 해

를 구할 수 있다. Lagrangian 은 다음과 같다.

$$L(\pi_j,\gamma,\eta_j)=\sum_{j=1}^{m}\ln\left(\frac{\overline{s_j^2}\cdot\pi_j^{\max}}{2(1-\overline{s_j}\cdot\pi_j^{\max})}-\frac{\overline{s_j^2}\cdot\pi_j}{2(1-\overline{s_j}\cdot\pi_j)}\right) - \gamma\left(\sum_{i=1}^{n}\lambda_i-\sum_{j=1}^{m}\pi_j\right)-\sum_{j=1}^{m}\eta_j(\pi_j-\mu_j). \quad (13)$$

KKT 조건은 다음과 같다.

$$\frac{\partial L}{\partial \pi_j}=\frac{1}{\pi_j-\pi_j^{\max}}+\frac{\overline{s_j}}{1-\overline{s_j}\pi_j}+\gamma-\eta_j=0,\ 1\le j<m. \quad (14)$$

$$\frac{\partial L}{\partial \gamma}=\sum_{i=1}^{n}\lambda_i-\sum_{j=1}^{m}\pi_j=0 \quad (15)$$

$$\eta_j(\pi_j-\mu_j)=0,\ \eta_j\ge 0,\ 1\le j\le m. \quad (16)$$

위의 KKT 조건들을 풀어서 정리를 하면 다음과 같은 해를 도출할 수 있다.

$$\pi_j=\frac{1}{2\overline{s_j}}+\frac{\pi_j^{\max}}{2}-\frac{\sqrt{(\overline{s_j}\pi_j^{\max}-1)(\overline{s_j}(\gamma\pi_j^{\max}-4)-\gamma)}}{2\sqrt{\gamma\overline{s_j}}} \quad (17)$$

수식 (17)의 라그랑지 상수 γ 는 수식 (6)과 수식 (17)을 이용하여 결정할 수 있다. 따라서, Theorem 1 의 증명이 완료되었다. ■

수식 (11)은 최대 작업처리시간 이내로 작업처리시간을 최소화하기 위해서 스케줄러가 각 자원에 작업을 할당할 때, 자원 j 의 작업 도착율을 의미한다. 추가적으로 NBS 는 4 가지 공리를 만족하기 때문에 협력적 게임에 참여하는 참여자에게 공정한 효용을 보장한다. 따라서 모든 자원에서 동일한 수준의 평균 작업처리시간을 보장하므로 어떤 자원을 사용하더라도 사용자에게 공정한 서비스를 제공할 수 있다.

3. 성능평가

본 논문에서는 본 연구와 직접적인 연관이 있는 2 개의 성능평가 지표를 고려한다. 먼저, 본 연구의 궁극적인 목표가 평균 작업처리시간을 최소화하는 것이기 때문에 작업처리시간을 성능평가 지표로 한다. 또한, 각 자원으로 부하가 균등하게 할당되었는지를 확인하기 위한 성능평가 지표로써 *fairness index*(*FI*)를 이용하며 다음과 같이 정의된다.

$$FI=\frac{\left(\sum_{i=1}^{n}T_i\right)^2}{n\sum_{i=1}^{n}T_i^2} \quad (18)$$

공정성 지수는 다른 자원에서 작업처리시간의 동일성을 측정하기 위한 지표이다. 그러므로 부하의 균형을 측정하는 것이다. 만약 모든 자원이 동일한 작업처리시간을 가진다면 공정성 지수는 1 이 되고, 모든 작업

<표 1> Processing rate of Resources

Resource	1-10	11-25	25-40	40-50
Processing Rate (job/sec)	0.01	0.02	0.033	0.1

에 대해서 100% 공정함을 의미한다. 또한, 제한한 기법과의 성능비교를 위해 Proportional load allocation scheme 을 구현하였다[5]. Proportional load allocation scheme 은 각 자원의 작업처리율에 비율적으로 작업을 할당하는 기법이다. 성능평가를 위해 각 자원의 작업처리시간은 지수분포를 따르고, 자원의 수는 50 개로 가정하였으며, 각 자원의 처리율은 Table 1 에 따른다. 시스템 부하(system load)는 전체 자원의 작업처리율의 합에 대한 전체 작업의 도착률 λ 에 대한 비율로 정의한다.

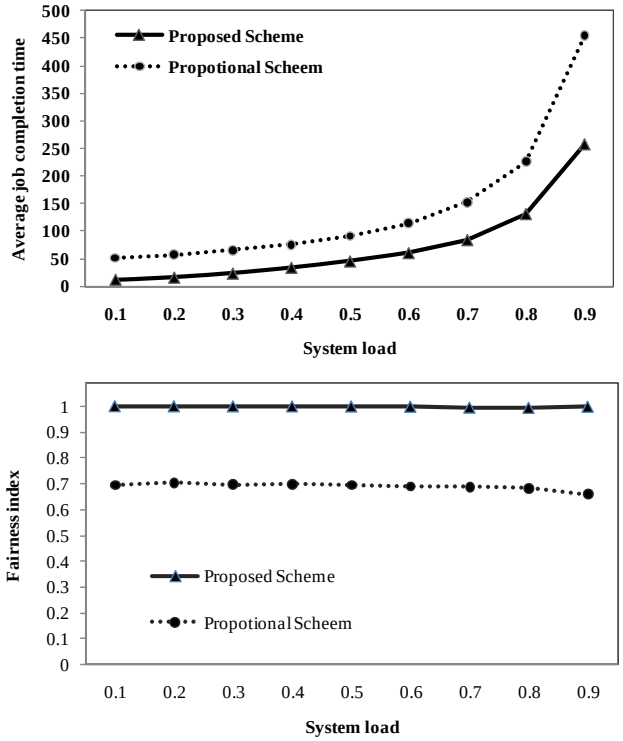

그림 2 Average completion time and fairness index

그림 2 는 시스템 부하가 0.1 에서 0.9 까지 변화함에 따른 평균 작업처리시간과 공정성 지수를 나타낸 것이다. 먼저, 주된 고려사항인 평균 작업처리시간을 고려한다. 제안하는 기법은 Proportional scheme 과 비교하였을 때 모든 시스템 부하에서 더 작은 작업처리시간을 가진다. 시스템 부하가 0.1 에서 0.6 까지는 2 가지 기법 모두 작업처리시간이 선형적으로 증가하였으나 0.7 이후부터는 작업처리시간의 증가량이 커진다. Proportional scheme 의 작업처리시간이 좋지 않은 것은 비효율적인 작업분배로 인해 상대적으로 작업처리 능력이 작은 자원에 큰 부하가 할당되었기 때문이다.

그림 2 에서는 2 가지 기법의 공정성 지수를 확인할 수 있다. 제안한 기법은 전체 시스템 부하에서 공정성 지수를 1 로 유지하며 이것은 앞에서 언급한 것처럼 자원의 성능에 대비하여 부하가 균등하게 할당된 것을 의미한다 또한, 사용자의 모든 작업에 대해서 공정하다는 것을 의미한다. 반면에. Proportional scheme 은 평균적으로 0.69 의 공정성 지수를 가진다. 이와 같은 경우에는 사용자의 작업이 어떤 자원에 할당되느냐에 따라 다른 작업처리시간이 달라질 수 있기 때문에 균일한 QoS 를 보장할 수 없다. 그림 2 에 도시

된 시스템 부하에 따른 평균 작업처리시간과 공정성 지수를 고려할 때, 본 논문에서 제안하는 협력적 작업할당 기법은 사용자의 작업을 처리함에 있어서 최소의 작업처리시간과 공정성 모두를 달성할 수 있음을 확인하였다.

4. 결론

본 논문에서는 클라우드 사용자들이 제출한 작업을 데이터센터 내의 각 자원들이 할당된 작업의 처리시간을 최소화하고자 경쟁 관계에 있는 상황 하에서 효율적으로 부하를 분산할 수 있는 작업할당 문제에 대해 논의하였다. 상기에 제시한 작업할당 문제를 해결하기 위해 협력적 게임이론인 NBS 를 통해 최적화 문제를 해결하였다. 본 논문에서는 각 자원에 할당되는 작업부하를 효율적으로 제어함으로써 모든 자원의 효용이 공평하게 최대화되는 파레토 최적을 유지할 수 있는 협력적 부하분산 기법을 도출하고 증명하였다. 시뮬레이션 기반의 성능평가를 통해 제안한 협력적 부하분산 기법이 모든 자원에서의 작업처리시간을 최소화할 수 있음을 확인하였다. 또한, 제안하는 기법을 적용하였을 경우, 모든 자원에서의 평균 작업처리시간이 동일함을 확인하였고, 이는 작업할당 기법이 사용자들에게 공정함을 의미한다.

Acknowledgement

본 연구는 2010 년도 정부(교육과학기술부)의 재원으로 한국연구재단-미래기반기술개발사업(첨단융복합분야)의 지원을 받아 수행된 연구임[N01100428].

참고문헌

[1] V. Bharadwaj and T. G. Roberazzi, “Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems,” *Cluster Computing*, Vol. 6, No 1, pp. 7-17, 2003.

[2] R. Wolski et al., “Analyzing market-based resource allocation strategies for the computational grid,” in *Proc. Int’l Parallel and Distributed Processing Symposium*, 2001.

[3] L. Kleinrock, “Queuing Systems Volume 1: Theory,” *Wiley-Interscience*, Jan. 1975.

[4] John F. Nash, Jr., “The Bargaining Problem,” *Econometrica*, vol. 18, no. 2, pp. 155-162, Apr. 1950.

[5] Y.C. Chown and W.H. Kohler, “Models for Dynamic Load Balancing in a Heterogeneous Multiple Processor System,” *IEEE Trans. Computers*, vol. 28, pp.354-361, 1979.