그리드 시스템을 위한 예약 기반 혼합 백필 스케줄링

최창열[O] 강창훈[+] 박기진 김성수
아주대학교[O], 극동정보대학교
{clchoi[O], kiejin, sskim}@ajou.ac.kr, {chkang[+]}@kdc.ac.kr

Reservation-based Hybrid Backfilling Scheduling for Grid Systems

Changyeol Choi[0], Chang-Hoon Kang[+], Kiejin Park, Sungsoo Kim
Ajou University[0], Keukdong College[+]

요 약

그리드 시스템은 지리적으로 분산된 컴퓨팅 자원들을 네트워크로 연동하여 공유할 수 있도록 하는 통신 서비스의 일종이라 할 수 있으며, 향후 그리드 미들웨어를 통한 다수의 응용 그리드들이 구축될 것으로 예상된다. 본 논문에서는 우선순위가 높은 작업은 예약이 가능한 작업 큐(Job Queue)로 분배하고 우선순위가 낮은 작업은 백필이 가능한 백필 큐(Backfill Queue)로 할당시키면서, 백필 효율성을 증대시킬 뿐만 아니라 전체 그리드 시스템의 성능을 향상시킬 수 있는 혼합 백필 스케줄링 기법을 제안한다.

1. 서 론

최근 과학 기술이 발전함에 따라 여러 복잡한 문제를 해결하기 위하여 고성능의 계산 자원이 필요하게 되었다. 이러한 요구를 충족시키기 위하여 최근 지역적으로 분산되어 있는 이질적인 고성능 컴퓨팅 자원을 하나로 묶어 거대한 시스템을 구성하는 그리드 시스템에 대한 연구가 활발하게 이루어지고 있다. 더욱이 이기종 병렬 컴퓨팅 환경에 적합한 스케줄링 기법 중 빠른 응답 시간을 제공하는 백필(Backfill) 기법을 기반으로 한 다양한 그리드 스케줄링 기법이 소개되었다[1]. 백필을 사용한 스케줄링에서는 작업의 특성과 그 우선순위에 따라서 결과가 달라지며 이는 시스템의 성능에 영향을 준다. 하지만 기존 백필 기법에 대한 연구에서는 시스템의 자원 효율을 높이면 작업의 지연시간이 늘어나는 문제점이 있거나, 반대로 작업의 지연 시간을 줄이면 시스템의 자원 효율이 낮아지는 문제점을 가지고 있다. 이는 그리드를 구성하는 각 노드에 작업들을 분배하는 메타 스케줄링 방법에만 치중하였기 때문이며, 더욱이 각 작업들을 각 노드에 분배 한 후에 각 노드 내에서 작업 스케줄링을 병행하여 처리하는 방법은 거의 연구되지 않고 있는 실정이다.

따라서 본 논문에서는 작업의 특성을 고려하여 예약 기법을 기반으로 우선순위를 조정함으로써 전체 시스템의 자원 효율을 증가시키거나 작업의 지연시간을 단축시킬 수 있는 혼합 백필 스케줄링 기법을 제안한다. 이는 그리드 시스템 상의 그리드 컴퓨팅 노드로 제출된 작업을 실행시키는데 필요한 프로세서 수와 예상 작업수행 시간의 특성에 따라 그리드 요청 작업들을 구분하여, 우선순위가 높은 작업은 예약이 가능한 작업 큐(Job Queue)로 분배하고 우선순위가 낮은 작업은 백필이 가능한 백필 큐(Backfill Queue)로 할당시킴으로써 시스템의 효율을 높이는 방법이다.

"본 연구는 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행되었음." (KRF-2005-041-D00630)

2. 관련연구

그리드 상의 적절한 노드에 작업을 분배하는 메타 스케줄링은 Centralized 스케줄링, 계층적 스케줄링, De-centralized 스케줄링으로 나눌 수 있다. Centralized 스케줄링은 그리드 상의 모든 노드의 상태 정보를 관리하며 제어하는 메타 스케줄러가 있고, 그리드의 모든 작업은 메타 스케줄러에게 제출되고 메타 스케줄러는 적절한 노드에 분배하는 방법이다. 하지만, 이 방법은 메타 스케줄러가 모든 정보를 관리하며 작업을 분배하기 때문에 Scalable 하지 못하다[2]. De-centralized 스케줄링은 메타 스케줄러가 별도로 존재하지 않고 모든 작업은 해당 노드의 스케줄러에게 제출되고 각 노드의 스케줄러는 적절한 스케줄링 정책에 의해 그리드 상의 일부 혹은 전체 노드의 상태를 파악하여 적절한 노드로 작업을 분배하는 방법이다. 이 방법은 Scalable 하지만 구현하기 어렵고 동기화 하기 어려운 단점이 있다[3].

또한 백필 스케줄링은 Conservative 백필과 EASY 백필 방법으로 구분할 수 있다. Conservative 백필은 현재

큐의 뒤쪽에 있는 우선순위가 낮은 작업이 그 작업보다 먼저 수행되어 야 하는 모든 작업을 지연시키지 않는 경우에만 백필하여 작업을 수행시키는 방법이며 EASY 백필은 우선순위가 낮은 큐의 뒤쪽에 있는 작업들이 큐의 맨 처음에 있는 작업을 지연시키지만 않으면 백필하여 수행시키는 방법이다. EASY 백필 방식을 사용하면 Conservative 백필 방식보다 더 많은 작업을 백필하여 수행시킬 수 있어 전체 시스템의 효율을 높일 수 있지만 사용자에게 수행 시간을 약속하기가 어려운 무한대기(Unbounded Delay)로 인해 지연시간이 길어지면 시스템의 성능이 저하된다.

3. 혼합 백필 스케줄링을 위한 그리드 시스템의 구조

그림 1은 Hybrid 스케줄링을 사용하는 그리드 컴퓨팅 시스템의 구조를 보여준다. 각 노드(Node)는 그리드 컴퓨팅에 참여한 컴퓨터(Worker)와 지역 스케줄러 및 전역 스케줄러(Superscheduler)로 구성되어 있다.

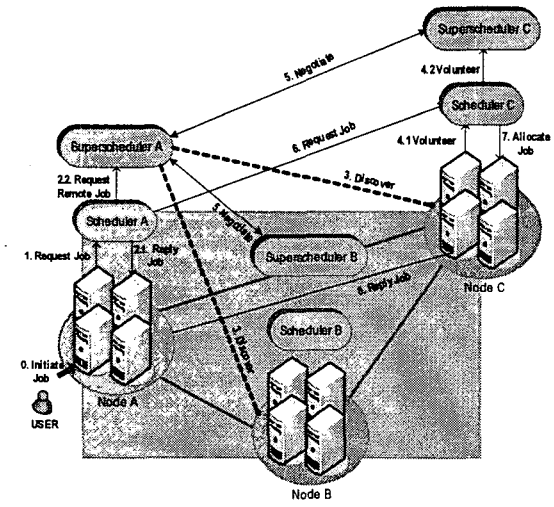

그림 1. Hybrid 스케줄링 적용을 위한 흐름도

따라서 Hybrid 스케줄링 구조란 노드내 스케줄링 부하를 최소화하기 위한 계층적 스케줄링 전략과 노드간 협업을 위한 전역 스케줄러간 상호작용을 지원하기 위한 De-Centralized 스케줄링 전략을 동시에 고려하기 위한 그리드 시스템 구조이다. 따라서, 이는 전통적인 계층적 스케줄링 전략의 지역/전역 작업 스케줄링이 가능하다는 장점과 De-Centrailzed 스케줄링 전략의 통신 병목현상을 제거하고 SPOF(Single Point of Failure)가 없으며, 확장성이 좋다는 장점을 모두 활용하기 위해 채택한 혼합 구조이다. 다시 말해 Hybrid 스케줄링 구조는 메타 스케줄링과 지역 작업 스케줄링을 병행하여 전체 시스템의 효율성 및 가용성을 증가시킬 수 있는 구조이다. 또한 원격 요청 작업과 지역 요청 작업 분배를 결정하기 위한 지역 스케줄러는 성능 향상을 위해 다중 큐 백필 메커니즘을 탑재한다. 다중 큐 분배는 해당 노드로 제출되는 작업들을 작업의 특성에 따라 작업 큐와 백필 큐로 분배하는 과정이다. 백필 스케줄링에서는 얼마나 많은 작업을 효율적으로 백필 하느냐가 중요한데, 백필 될 확률이 높은 작업은 필요 프로세서 수가 적은 작업들이다. 이는 필요 프로세서 수가 적을수록 노드내 유휴 프로세서 확보가 용이하기 때문에 백필 될 확률이 높아지고 따라서 전체 프로세서의 이용률을 향상시킬 수 있기 때문이다. 본 과정을 간략히 기술하면, 먼저, 작업들을 해당 노드의 시스템의 전체 프로세서 수를 기준으로 세 개의 그룹(N,M,W)으로 나누고, 두 번째로, 첫째 단계에서 나누어진 각 그룹의 작업들을 예상 실행시간에 따라 두 개의 그룹(L,S)으로 나눠 작업을 총 6가지 (NL, NS, ML, MS, WL, WS)로 구분한다.또한 먼저 노드내 발생한 작업(Local job)이 다른 노드에서 발생한 작업(Remote job)에 의해 지연되는 것을 방지하기 위해 크게 작업 유형(flag)을 2가지로 분류한다.

그리드 시스템의 스케줄링은 최초 사용자가 작업 요청을 하면 전역 스케줄러는 작업을 해당 노드에서 처리하게 할 것인지, 원격 노드에서 처리하게 할 것인지에 대한 의사결정을 내린다. 이를 위해 전역 스케줄러는 현 노드 및 그리드 시스템의 상태 정보를 수집한다. 또한 전역 스케줄러는 항시 다른 노드에서의 원격 처리 작업 요청을 바로 처리하기 위해 대기 상태를 유지하며, 필요시 다른 전역 스케줄러와의 협업을 수행한다. 또한 기존 연구에서 개발된 백필 방법 중 대표적인 Conservative 백필과 EASY 백필 두 방법의 단점을 보완하고 무한대기를 방지하기 위하여 두 백필 기법의 장점을 혼합하였다. 이를 위해 Conservative 백필의 단점인 시스템 이용률 향상을 위해 예약기법을 도입하였으며, Easy 백필의 단점인 무한대기 현상을 제거하기 위해 백필에 대한 횟수의 임계치를 사용한다. 다시 말해서 첫 번째 예약 방법으로 작업 큐에 있는 작업이 무한대기 하지 않도록 하기 위해 작업 큐에서 실행이 보장되는 작업의 수를 여러 개로 늘렸으며, 한 백필 큐에서 백필 기회를 기록하여 백필되지 못한 수치가 일정 값보다 커질 경우 이 작업을 작업 큐로 보내어 일정 시간 후에 실행을 보장 받도록 하여 무한 대기를 방지한다.

4. 성능평가

성능 평가를 위한 실험 데이터 생성을 위해 본 논문에서는 평균 작업 도착 시간(Mean Arrival Time), 평균 작업 요청 시간(Mean Estimated Execution Time), 평균 필요 프로세서 수(Mean Width, Mean Number of Processor) 등을 실험을 위해 필요한 파라미터 값을 Feitelson Archive[4]의 작업부하 로그(Workload logs)의 집합으로

부터 추출하였고, 해당 파라미터를 기본값으로 하는 확률 함수를 근간으로 만들어진 RGD(Randomly Generated Data)를 입력 데이터로 사용하였다. 이는 기존 연구결과를 분석해 보면 동일한 기법이라 할지라도 대상 Workload에 따라 성능이 개선될 수도 또는 저하될 수 있기 때문이다. 실험을 위해 사용된 기본 설정은 그리드 시스템의 노드수는 8개, 노드당 평균 보유 프로세서 수는 64개, 작업 도착율은 평균 분당 0.167 (7200개/월)을 갖는 지수함수, 작업 요청 시간은 평균 100 분을 갖는 지수함수를 따르며, 평균 필요 프로세서 수는 1에서 64개의 Uniform 함수를 따른다. 또한 다중 큐를 사용하는 경우 ($AverageSlowdown_m$)가 단일 큐를 이용하였을 경우 ($AverageSlowdown_1$)보다 얻을 수 있는 성능 이득이 어느 정도인지 분석하기 위해 성능 파라미터로 평균 작업 지연율을 사용한다.

$$SlowdownRatio = \frac{AverageSlowdown_1 - AverageSlowdown_m}{\min(AverageSlowdown_1 - AverageSlowdown_m)}$$

즉 SlowdownRatio가 0보다 크다는 의미는 단일 큐를 이용한 경우 평균 지연시간이 더욱 크다는 것을 뜻하며 따라서 다중 큐를 사용하였을 경우 평균 지연시간을 줄일 수 있다는 것을 의미한다. 반대로 0보다 작다는 것은 다중 큐를 사용함으로써 관리 오버헤드가 증가하여 평균 지연시간이 증가되었다는 것을 의미한다.

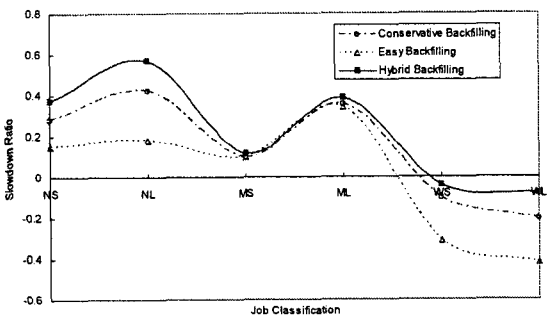

그림 2. 작업 분류 기준에 따른 평균 지연율

그림 2는 요구 프로세서 수 및 작업 예상 수행시간에 따라 본 논문에서 크게 6가지로 분류한 기준을 근간으로 작업의 평균 지연율을 분석한 결과로 3 방식 모두 단일 큐만을 사용할 때보다 다중 큐를 사용함으로써 작업 지연율을 줄일 수 있는 효과를 얻을 수 있다는 결과를 보여준다. 그리고 다중 큐를 사용한다 할지라도 백필에 의한 대기 현상으로 발생할 수 있는 지연 현상은 상대적으로 적은 프로세서를 요구하는 작업의 경우에는 발생하지 않았다. 하지만 3방식 모두 다수 프로세서를 필요로 하는 작업 (WS & WL)의 경우 우선순위가 낮은 작업이 백필되는 것에 의해 지연이 발생함을 볼 수 있다. 그러나 Easy 백필 방식과 Conservative 백필 방식보다는 본 논문에서 제안한 Hybrid 다중 큐 스케줄링 방식이 단일 큐를 사용한다 할지라도 다수 프로세서를 필요로 하는 작업의 지연율이 적게 발생함을 보여준다. 따라서 Hybrid 스케줄링 방식은 단일 큐를 사용하더라도 예약 기법을 통해 백필 비율을 높일 수 있는 Easy 백필 방식과 무한 대기 현상을 방지할 수 있는 Conservative 백필 방식의 장점을 모두 제공하고 있다는 것을 알 수 있다.

5. 결론

그리드 시스템의 성능 및 이용률을 향상시키기 위한 스케줄링 기법 개발에 대한 연구가 활발히 진행되고 있지만, 문제 해결 범위를 축소화하기 위해 각 노드에 작업들을 분배하는 메타 스케줄링 방법과 노드내 작업 스케줄링 방법에 대한 연구를 분리하였다. 하지만, 보다 그리드 시스템의 성능을 향상시키기 위해선 둘 모두를 동시에 고려할 수 있는 스케줄링 기법 개발이 필요하여, 본 논문에서는 예약 기법을 기반으로 하는 혼합 백필 스케줄링 기법을 개발하였다. 이는 기존 구조의 문제점인 확장성과 비효율성 모두 제거할 수 있는 대안으로, 전체 그리드 시스템의 작업 지연율을 줄여 성능 또한 향상시켰다. 향후에는 원격 작업 처리를 위한 네트워크 오버헤드를 줄이기 위한 방안을 마련하여 본 논문에서 제안한 스케줄링 기법의 세부 성능을 향상시킬 것 이다.

참고문헌

[1] A. Mualem, et al., "Utilization, Predictability, Workloads and User Run time Estimates in Scheduling the IBM SP2 with Backfilling," IEEE Transactions on Parallel and Distributed System, Vol. 12, No. 6, pp. 529-543, June 2001.

[2] Q. Wang, et al., "De-centralized Job Scheduling on Computational Grids Using Distributed Backfilling," Proceedings of the 3rd International Conference on Grid and Cooperative Computing, pp. 285-292, Oct. 2004.

[3] V. Hamscher, et al., "Evaluation of Job-Scheduling Strategies for Grid Computing," Proceedings of the 1st IEEE/ACM International Workshop on Grid Computing, pp. 191-202, Dec. 2000.

[4] D. Feitelson, "Logs of Real Parallel Workloads from Production Systems," http://www.cs.huji.ac.il/labs/parallel/workload/logs.html.