

Easy Backfill 에 기초한 그리드 컴퓨팅

다중 큐 스케줄링 기법

박 미 선^{0*} 박 기 진^{**}

안양대학교 문리과학대학 컴퓨터학과^{*} 아주대학교 공과대학 산업정보시스템공학부^{**}
oooooo82@anyang.ac.kr^{*} kiejin@ajou.ac.kr^{**}

Multi-queue Scheduling Algorithm for Grid computing based on Easy Backfill Technique

Misun Park^{0*} Kiejin Park^{**}

Department of Computer Engineering, Anyang University^{*}
Division of Industrial & Information Engineering, Ajou University^{**}

요 약

그리드와 웹 서비스가 서로 밀접한 연관성을 가지고 있는 분야로 인식되고 있으면서, 그리드 기반 웹 서비스의 QoS(Quality of Services)에 대한 관심이 높아짐과 동시에 중요한 부분으로 여겨지고 있다. 본 논문에서는 QoS 요소들 중 응답 시간(Response time) 향상을 위해서 기존의 그리드 시스템에 사용되고 있는 Easy Backfill 스케줄링 기법을 기본으로 한 다중 큐 스케줄링을 연구하였으며, 또한 기존 Backfill 기법의 큰 작업에 대한 응답 시간이 느려진다는 단점을 보완하기 위해, 각 작업들에 대해서 예약 정책을 연구하였다. 이를 통해 그리드 기반 웹 서비스의 응답 시간 성능이 개선됨을 확인하였다.

1. 서 론

최근 차세대 인터넷 서비스로 각광받는 그리드(Grid)[1] 컴퓨팅은 지리적으로 분산된 컴퓨팅 자원들을 네트워크로 연동하여 공유할 수 있도록 하는 통신 서비스의 일종이라 할 수 있으며, Globus 와 Legion 과 같은 미들웨어를 사용하여, 그리드 시스템을 빠르게 구성할 수 있다. 한편 웹 서비스는 원격 프로시지 호출(Remote Procedure Call)기능을 제공하는 새로운 웹 응용으로, 웹 서비스가 인터넷상에 배치되면, 이미 존재하는 웹 서비스를 포함하여 어떠한 응용도 웹 상에 배치된 서비스를 검색하고 호출할 수 있다[2]. 서로 다른 목적을 위해 개발되고 있는 웹 서비스와 그리드 컴퓨팅이지만, 두 개의 기술은 "동적이며 여러 기관에서 참여하는 가상 조직을 통한 문제 해결과 자원 공유"와 같은 동일한 목표를 다루고 있다고 볼 수 있다. 향후 그리드 미들웨어를 통한 다수의 응용 그리드들이 구축될 것으로 예상되나 그리드에 대한 기술적 이해가 부족한 일반 사용자들은 여러 응용 그리드를 이용하

여 자신의 작업을 수행할 경우 그리드 사용에 어려움을 겪게 될 것이다. 예를 들어 그리드를 이용하는 생물학자가 유전자 분석 그리드와 단백질 분석 그리드를 함께 이용하고자 할 때 현재의 기술로는 각각 그리드의 사용법을 알아야 하며 제공되는 인터페이스를 통한 프로그래밍 작업을 거쳐 원하는 결과를 도출해야 한다. 하지만 분산 컴퓨팅의 또 다른 기술인 웹 서비스를 그리드의 기반 환경으로 구축하여 사용하면 사용자는 그리드의 기술적 내용에 대한 이해가 전혀 없어도 단지 웹 메소드를 호출하는 것만으로 쉽게 프로그래밍을 할 수 있게 된다. 이러한 이유에서 현재 그리드의 기반 환경으로 웹 서비스를 사용하는 방법들이 소개되고 있으며, 그리드와 웹 서비스는 서로 밀접한 연관성을 가지고 있는 분야로 인식되고 있다. 그로 인해 현재 웹 서비스의 사용이 급격히 증가하면서 사용자의 웹 서비스 QoS(Quality of Services)에 대한 관심이 높아지고 있다.

본 연구는 한국과학재단 목적기초연구(R05-2003-000-10345-0) 지원으로 수행되었음.

이기종(Heterogeneous) 그리드 컴퓨팅 환경에 적합한 스케줄링 기법 중 빠른 응답 시간을 제공하는 Backfill 기법을 기본으로 한 다양

한 스케줄링 기법[3,4]들이 소개되고 있지만, Backfill 기반 스케줄링들은 크기가 큰 작업들의 무한 대기(Starvation)를 발생시키는 문제점이 있다. 이를 해결하기 위해서 본 논문에서는 이기종 그리드 환경 하에서 비선입선출 기법을 기본으로 하고 있는 Easy Backfill 기법에 기반하여, 기존 연구에서 고려하지 않은 다중 큐를 사용하였다. 또한 그리드 기반의 웹 서비스에서 기존 Backfill 기법의 큰 작업에 대한 응답 시간이 느려진다는 단점을 보완하기 위해, 각 작업들에 대해서 예약 정책을 제안하여, 평균 응답 시간 문제와 큰 작업들의 무한 대기 문제를 해결하고자 하였다. 본 논문의 2 장에서는 관련 연구를 언급하고, 3 장에서 Backfill 기반의 다중 큐 스케줄링 기법을 제시하였으며, 4 장에서는 제안한 방법의 성능 평가를 수행하고, 5 장에는 결론을 내렸다.

2. 관련 연구

그리드 시스템에서의 3 가지 대표적인 스케줄링 기법으로는 선입선출(FCFS), 랜덤기법(Random), Backfill 기법이 있으며[5], 선입선출 기법은 대기 시간이 길어지기 때문에 비효율적이며, 랜덤기법은 작업들에 대해서 랜덤하게 선택하는 방법으로 공정성이 보장되는 장점이 있다. 마지막으로 Backfill 기법은 큰 작업으로 인한 필요 없는 휴지 시간(idle time)을 예방하는 장점이 있다. 이 기법들은 작업 환경에 따라 사용 용도가 결정되는데, 이기종 환경에서는 Backfill 기법이 가장 응답 시간이 빠르며, Job-pool 환경에서는 선입선출 기법이, 계층적인 스케줄링을 원할 때는 랜덤기법이 유리하다.

한편 이기종 그리드 환경에서 유리한 Backfill 기법의 종류로는 Easy Backfill 과 Conservative Backfill 이 있는데, Easy Backfill 기법이란 작업의 도착 시간으로 수행되되, 특정한 부분에서는 필요 노드 수와 작업 크기를 고려하여, 작업들의 수행 순서가 큐의 가장 처음 작업은 대기 하지 않는다는 가정 하에 바뀌게 되는 것을 의미하며, Conservative Backfill 은 Easy Backfill 기법과 같되, 큐 안의 모든 작업들이 대기하지 않는다는 전제 하에 순서가 바뀌게 되는 스케줄링이다. 대체적으로 Easy Backfill 기법이 우수하지만, 요청 프로세서 수가 많은 작업에 대해서는 Conservative Backfill 이 우수하다[5].

3. 다중 큐 스케줄링

제안한 Easy Backfill 기반의 다중 큐 스케줄링은 큐의 개수를 기존 1 개에서 3 개로 하자는 아이디어이며, 큐의 개수를 늘림으로써 큰 작업들로 인한 작은 작업들의 대기 시간을 최소화하여 응답 시간 향상을 목표로 한다. 그리드 요청 작업(Job) 스케줄링에 적용된 가정은 다음과 같다.

- 1. 수행되는 모든 작업들은 작업도착시간(Arrival time), 실제수행

시간(Execution time), 예측수행시간(Estimated Execution time)과 요청된 프로세서의 수(Number of processors)의 파라미터에 의해 구분된다.

- 2. 큐는 아래의 조건에 의해 총 3 개로 이루어진다. 여기서 x 값은 수행시간을 요청 프로세서 수로 나눈 값이며, 이 값이 1 인 경우를 기준으로 계산량이 적은 작업과 많은 작업을 구별한다.

Queue 1 => $x > 1$: 계산량이 적은 작업

Queue 2 => $x = 1$

Queue 3 => $x < 1$: 계산량이 많은 작업

- 3. 예약된 작업은 우선 처리하며, 예약되지 않은 작업들은 모두 동일한 우선 순위를 갖고 있다.

3.1 다중 큐 알고리즘

제안된 다중 큐 알고리즘은 도착 작업의 우선 순위에 따라 작업을 처리하며, 각각의 큐에 도착된 작업은 아래와 같이 처리된다(그림 1 참조).

- 1. 그리드 클라이언트들의 요청들은 작업도착시간, 실제수행시간, 예측수행시간과 요청된 프로세서의 수에 대한 정보를 갖고 분배기(Dispatcher)에 입력된다.
- 2. 작업에 필요한 계산량(x)에 의해 각각의 큐에 작업이 들어간다.
- 3. 작업들이 큐에 들어갈 때 선별적으로 예약이 가능하다.
- 4. 각 큐에 들어간 작업들은 Easy Backfill 기법에 의해 작업의 실행 순서가 정해지게 되는데, 이는 요청 프로세서 수와 수행시간에 의한 정보로 정해진다.
- 5. 각각 큐의 첫 번째 작업들 중에서 도착시간이 빠른 작업이 가장 먼저 수행된다.

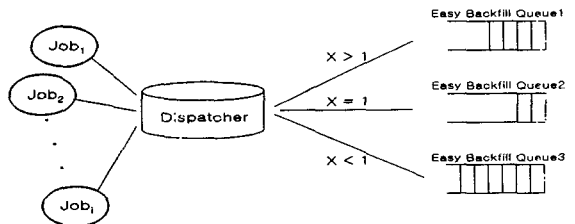


그림 1 다중 큐 스케줄링 구조도

3.2 예약(Reservation)

예약이 완료된 후에 취소할 수 없는 정적 예약에 비해, 작업 부하를 고려하여 취소 및 재예약이 가능한 동적 예약은 어떤 작업에 대한 무한대기 방지를 위한 방법이며, 임의의 큐에 작업이 들어감과 동시에 선택적으로 예약 여부를 결정한다. 예약된 작업들의 우선 순위는

큐의 첫 번째 작업의 도착시간과 예약된 작업의 요청시작시간을 비교하여, 빠른 순으로 결정된다. 이 방법은 작업 부하 즉 지연률(Delay_ratio)에 따라 취소나 재예약이 이뤄지며, 지연률은 식(1)에 의해 계산된다.

$$Delay_ratio = \frac{1}{number_of_jobs} * \sum_{i=1}^n job_i(e-a) * 100 \quad - (1)$$

(e : 예측수행시간, a : 실제수행시간)

Delay_ratio 값이 10%를 넘지 않으면, 예약에 대한 취소가 가능하며, 90%를 넘게 되면, 빠른 답변을 요하는 작업들에 한하여, 재예약이 가능하도록 하였다.

- Delay_ratio < 10% - 작업 지연 시간 적음, 예약 취소 가능
- 10% ≤ delay_ratio ≤ 90% - 예약 취소 및 재예약 필요요함.
- Delay_ratio > 90% - 작업 지연 시간 큼, 재예약 가능

4. 성능 평가

그리드 작업도착시간, 실제수행시간, 예측수행시간과 요청된 프로세서의 수 등의 평균값 및 분포는 기존 연구 결과의 성능 분석에 사용된 값을 적용하였다[3]. 그리드 평균 작업 크기의 경우 1000 초로 하며, 180 초 안에는 작업이 반드시 어느 큐에든 할당이 되어야 하므로 도착 시간 간격을 180 초로 한다(표 1 참조).

표 1 시스템 파라미터 (단위:second)

작업수행시간(예측,실제)	Exp(1000)
작업도착시간	Exp(180)
총 프로세서 수(노드)	128

그림 2에서는 프로세서를 고려하지 않고 작업 크기만을 고려한 기존의 방식과 작업의 계산량(x)을 고려한 다중 큐 스케줄링의 응답 시간 변화를 표시하고 있으며, 후자가 기존 스케줄링 기법보다 우수함을 나타내고 있다. 이는 작업 크기 대비 요청 프로세서 수를 고려하여 성능이 우수한 것으로 판단되는데, Easy Backfill을 기반으로 한 두 기법에서 요청 프로세서 수는 중요한 요소임을 확인하였다.

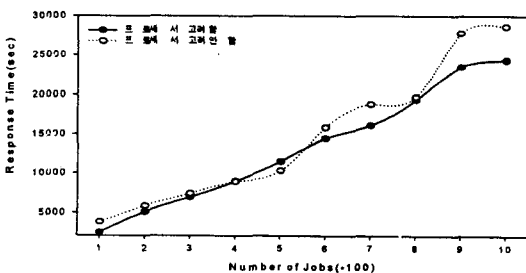


그림 2 프로세서 고려 유무에 따른 응답 시간 비교

그림 3에서는 예약 기법을 기존 방식의 정적에서 동적으로 한에 따른 응답 시간 변화를 나타내는데, 작업들의 지연률에 따라 예약의 취소나 재예약이 가능하게 됨으로 인해 응답 시간 또한 향상된 것으로 판단된다.

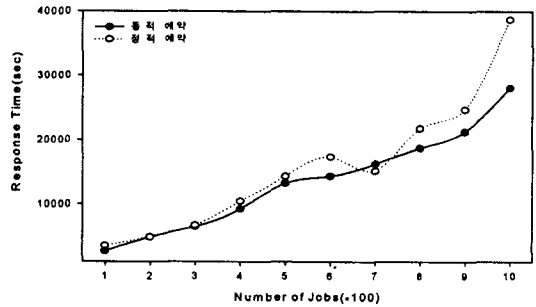


그림 3 예약 기법에 따른 평균 응답 시간 비교

5. 결론 및 향후 연구

본 연구에서는 Easy Backfill 기법을 기본으로 한 다중 큐 스케줄링 기법을 이용하여, 기존 Easy Backfill 스케줄링 기법의 장점은 극대화시키고, 단점은 보완하여, 그리드 기반 웹 서비스의 응답 시간을 향상시켰다. 추후에는 그리드 기반 웹 서비스에서 다른 QoS 요소의 성능을 향상시키기 위한 새로운 스케줄링 기법을 연구할 계획이다.

참고문헌

- [1] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "Grid Services for Distributed System Integration," Computer, Vol. 35, No. 6, pp. 37-46, 2002.
- [2] http://www.universe.co.kr/business/business_net_06.html
- [3] Barry G. Lawson and Evgenia Smirni, "Multiple-queue Backfilling Scheduling with Priorities and Reservations for Parallel Systems", 8th International Workshop, JSSPP 2002 Edinburgh, Scotland, UK, pp. 72 - 87, July 24, 2002..
- [4] D. G. Feitelson and A.M. Weil, "Utilization and predictability in scheduling the IBM SP2 the backfilling," in 12th International Parallel Processing Symposium, pp. 542-546, Apr.1998.
- [5] Volker Hamscher, et. al., "Evaluation of Job Scheduling Strategies for Grid Computing," the 1st IEEE/ACM International Workshop on Grid Computing (Grid 2000) at the 7th International Conference on High Performance Computing (HiPC - 2000), LNCS 1971, pp. 191-203, 2000.