

그리드자원의 정적 정보를 활용한 중복배정 스케줄링 알고리즘¹⁾

강상성[○] 강오한

안동대학교

web2edu@gmail.com, ohknag@andong.ac.kr

A Workqueue-Replication Scheduling Algorithm Using Static Information of Grid Resources

Sang-Seong Kang[○] Oh-Han Kang

Andong National University

그리드(Grid) 시스템은 지역적으로 분산되어 있는 자원을 묶어 하나의 고성능 처리기 또는 대용량 저장소처럼 사용할 수 있도록 해주는 병렬·분산 시스템이다. 그리드를 고성능 처리기로 활용하는 계산그리드는 프로세서의 처리능력을 공유함으로써 복잡하고 많은 시간을 소요하는 작업들을 각 자원에 배분하여 보다 짧은 시간에 작업을 완료하도록 한다. 넓은 지역에 분산되어 있는 이질적인 자원들로부터 최적의 처리결과를 얻기 위해서 효율적이면서도 효과적인 스케줄링 알고리즘은 그리드 시스템의 중요한 기반이 된다.

전통적인 병렬·분산 컴퓨팅 환경에 비하여 그리드 시스템을 구성하는 자원들은 보다 이질적이고 자율적이다. 또한 동적인 성능 특성을 가지고 있으며 자원의 선택과 연산에 필요한 데이터가 분리되어 있다. 따라서 비교적 동질적이고 전용성이 보장된 자원들로 구성된 시스템에 사용된 전통적 병렬·분산 시스템의 알고리즘들은 그리드 환경에서는 좋은 성능을 기대할 수 없다. 최근 그리드 시스템의 특성을 반영한 스케줄링 알고리즘이 연구되어 왔지만 여전히 전통적 병렬·분산 환경의 알고리즘에 의존한 연구에 비하여 그 수가 적고 연구의 수준 역시 초기 단계에 머무르고 있다. 이에 본 연구에서는 기존의 그리드 스케줄링 알고리즘을 분석하여 스케줄링에 포함하거나 배제할 요소를 도출하여 개선된 알고리즘을 개발하였다.

본 연구에서는 상호독립적인 작업집합을 스케줄링 하는 Min-Min 알고리즘, Buyya의 경제모델 중 시간 최적화 알고리즘, Workqueue, WQR 등 네 가지 알고리즘을 웹기반 그리드 스케줄링 시뮬레이션 도구인 WGridSP를 사용하여 성능을 평가하고 결과에 대한 원인을 분석하였다. 시뮬레이션 환경으로 그리드를 구성하는 자원의 개수는 50개로 지정하고 각 자원의 최소부하는 0%로 지정되어 있으며, 최대 부하는 10%, 30%, 50%, 70%, 90%로 변화를 주도록 하였다. 이와 같이 부하에 따라 알고리즘의 성능을 비교한 이유는 그리드 시스템에서 반환시간에 가장 많은 영향을 주는 요인이 자원의 부하인 것으로 판단하였기 때문이다. 자원의 부하가 과도하게 높은 경우는 자원이 로컬에서 사용되고 있거나 많은 작업이 몰려 있을 수 있으며 때로는 일시적으로 사용이 불가능 경우로도 받아들여질 수 있다. 각 자원의 부하가 최소 부하에서 최대부하까지 변화하도록 구성하고 최대 부하와 시간별 가중치를 곱하여 각 자원의 현재부하가 변하도록 하였다. 또한 모든 자원이 동시에 같은 부하를 가지지 않도록 하기 위해 자원의 GMT 기준 시간대를 [0, 23]의 단일분포에서 무작위 배정하였다. 각 알고리즘에서 이용하는 정보에 대한 효율성을 중심으로 분석하기 위하여 두 가지 종류의 시뮬레이션을 수행하였다. 첫 번째는 자원의 정적 정보인 CPU 처리능력을 300으로, CPU 개수를 1개로 고정하였다. CPU 처리능력을 나타내는 숫자는 1초당 처리할 수 있는 명령어의 개수를 의미한다. 예컨대 길이가 1200인 작업을 처리능력 300인 CPU 1개를 소유한 자원은 4초만에 처리할 수 있다. 두 번째는 CPU 처리능력과 개수를 무작위로 배정하여 각 자원이 서로

1) 본 연구는 정보통신부의 2006년 IT정책개발지원사업(No. B1220-0601-0018)에 의하여 수행되었음.

다른 CPU 성능과 개수를 가지도록 하였다. CPU 처리능력은 [100, 500]의 단일분포에서, CPU 개수는 [1, 8]의 단일분포에서 무작위 선택하도록 설정하였다. 처리할 응용을 구성하는 작업은 총 200개로 구성하였으며 작업의 길이는 [1000000, 5000000]의 단일분포에서 무작위 선택되도록 하였다. 결과적으로 부하가 0%인 자원에 배정된다고 가정을 했을 때 하나의 작업 처리시간은 2000초에서 50000초 사이가 된다. 작업을 자원에 배정하고 결과를 반환받는데 소요되는 통신시간은 무시되도록 각 작업의 입출력 데이터를 모두 0으로 설정하였으며, 모든 시뮬레이션은 동일한 조건하에 10회 실시하여 산술평균값을 총 완료시간(Makespan)으로 사용하였다.

시뮬레이션 결과 자원의 CPU 개수를 1개로, CPU 처리능력을 300으로 고정한 상태에서 최대 부하 70%이하에서는 거의 비슷한 성능을 보이다가 70%이상에서 차이가 나타나기 시작한다. 자원의 최대부하가 100%에 근접할수록 소수의 극단적으로 느린 자원의 영향을 많이 받는 상황에 직면하게 되며 그러한 자원을 피해서 작업을 배정하는 WQR 알고리즘이 우수한 결과를 나타낸 반면 Min-Min 알고리즘은 자원의 동적 정보를 활용함에도 불구하고 가장 낮은 성능을 나타냈다. 이 결과로 극단적으로 느리게 처리하는 자원을 회피할 전략이 필요하며 동적인 실시간 부하를 고려하는 것은 비교적 장시간 실행하는 그리드 기반 작업에 적절하지 않음을 알 수 있다.

자원의 CPU 개수를 [1, 8]의 단일분포에서 무작위 선택하도록 하고, CPU 처리능력도 [100, 500]의 단일분포에서 무작위 선택하도록 한 상태에서는 반대로 WQR 알고리즘과 가장 나쁜 성능을 보인 Min-Min 알고리즘의 순위가 반대로 나타났다. WQR 알고리즘은 CPU의 수와 관계없이 모든 자원에 하나씩의 작업만을 배정하지만 Min-Min 알고리즘은 자원의 동적 정보를 활용하기 때문에 CPU의 수가 많으면 상대적으로 많은 작업을 배정한다. 그 결과 Min-Mix 알고리즘이 WQR보다 전체 자원의 활용도를 높임으로서 더 빠른 시간 내에 전체작업을 완료할 수 있는 것으로 보인다. Min-Min 알고리즘이 높은 성능을 보인 이유는 정적정보인 CPU 수가 간접적으로 고려되었기 때문인 것으로 보인다.

그리드 스케줄링에 활용하는 정보의 종류와 자원의 분포에 따른 시뮬레이션 결과를 분석하여 도출한 요소인 “극단적 부하 또는 불능 자원의 회피”, “동적 정보 활용의 배제”, “정적 정보의 활용”을 반영하기 위해 WQR의 재배정 기법을 활용하고 각 자원들을 CPU 처리능력 순으로 정렬하여 작업을 배정할 때는 자원이 소유한 CPU의 수만큼 배정하도록 하는 WQRuSI(Workqueue-Replication using Static Information) 알고리즘을 개발하였다.

제안된 알고리즘의 성능을 검증하기 위해 기존알고리즘 분석에서 두 가지 상황에 각각 우수한 성능을 보인 WQR과 Min-Min 알고리즘과 함께 시뮬레이션을 수행하였다. 정적 자원정보를 모두 일정하게 구성한 환경에서는 기존의 WQR과 비슷한 성능을 보였고, 정적 자원정보를 무작위로 선택한 환경에서는 WQRuSI 알고리즘이 가장 우수한 결과를 나타내고 있다. 정적 자원정보를 모두 동일하게 한 경우 정적 자원정보를 고려한 알고리즘의 효과가 드러나지 않은데 반해 자원의 프로세서 수와 처리능력을 다양하게 설정한 환경에서는 성능의 개선이 눈에 띄게 나타났다.

본 연구에서는 장애 또는 극단적 저성능 자원을 회피하기 위해 작업을 2개 또는 그 이상의 자원에 중복배정 하는 방법을 이용했으나 동시에 여러 사용자가 같은 방법을 사용할 경우 오버헤드가 크게 발생할 위험성이 존재한다. 이를 위해 이상 자원을 회피하기 위한 방법이 더 연구되어야 할 것이다. 또한 근본적으로 멀리 떨어져 있는 자원으로 구성되어 있는 그리드 시스템에서 데이터의 전송에 소요되는 시간을 무시할 수 없으므로 전송시간에 대한 요소도 알고리즘에 포함하고 실시간 부하정보에 따라 기 배정된 작업을 타 자원으로 이전하는 방법을 사용함으로써 정적 정보뿐만 아니라 실시간 부하정보를 활용하여 성능을 향상시키는 연구가 필요하다.