메시지 패싱 시스템의 통신 특성을 고려한 개선된 태스크 스케줄링 기법

노두호[0] 김성천
서강대학교 컴퓨터학과
stdu@nate.com[0] ksc@arqlab1.sogang.ac.kr

Improved Task Scheduling Algorithm considering the Communication Features of Message-Passing System

Duho Ro[0] Sungchun Kim
Sogang University. Dept. of Computer Science

요 약

본 논문에서는 메시지 패싱 시스템에서의 태스크 스케줄링에 대해 다룬다. 병렬/분산 시스템의 어플리케이션의 태스크에 대한 적절한 스케줄링이 이루어지지 않는 경우, 병렬/분산 처리를 이용한 이득을 기대하기는 어렵기 때문에 이 주제에 대한 연구는 컴퓨터 아키텍처의 발달과 함께 지속되고 있으며, 많은 연구들이 태스크 스케줄링에 대한 다양한 기법들을 제안하고 있다. 하지만 통신비용을 데이터를 소모하는 한쪽의 태스크에만 부과하는 기존 기법들을 메시지 패싱 시스템에 적용하기는 부족한 면이 있다. 본 논문에서는 기존 연구의 모델과 메시지 패싱 시스템에서 통신비용이 통신과 관계된 모든 노드에서 발생함을 고려하여 리스트 스케줄링 기법에 기초한 개선된 우선순위 함수와 새로운 프로세서 선택 기준을 제안한다. 이들 두 가지 제안을 적용한 태스크 스케줄링 기법은 메시징 패싱 시스템에서 통신비용이 누적되는 특징으로 발생하는 비효율적인 스케줄링을 개선한다.

1. 서 론

태스크 스케줄링은 태스크의 실행 순서와 태스크가 실행될 프로세서를 태스크들의 연관성을 고려하여 전체적인 어플리케이션의 실행 시간을 단축시킬 수 있도록 컴파일 시간에 결정하는 것이다[1, 2]. 최적의 실행 시간을 보장하는 태스크 스케줄링을 찾는 문제는 몇 가지 제한된 경우를 제외하곤 NP-complete이다[1, 2]. 그렇기 때문에 많은 연구들이 각기 연구 대상으로 삼고 있는 병렬/분산 어플리케이션이나 병렬/분산 처리 시스템에 적합한 휴리스틱 기법을 제안하고 있다.

최근 활발히 연구되고 있는 클러스터 시스템이나 그리드 시스템들은 메시지 패싱을 통하여 각 노드간의 통신이 이루어진다. 그러나 태스크 스케줄링에 대한 연구들은 공유메모리 시스템에서 출발하였으며, 통신에 있어서의 비용증가로 통신비용이 고려한 기법들이 연구되어 왔지만 메시지 패싱에 의해 통신이 이루어지는 시스템에서 사용되기에는 부족한 면이 있다. 이 연구에서는 메시지 패싱을 통한 병렬/분산 환경에서의 통신 특성인, 동기식 통신이 순차적으로 일어나는 환경을 고려하여 개선된 우선 순위 함수와 새로운 프로세서 선택 기준을 포함하는 새로운 스케줄 기법을 제안한다. 새로운 기법은 기존의 리스트 스케줄링 기법을 메시지 패싱 시스템에서 사용될 때 발생하는 비효율적 스케줄링을 개선한다.

2. 기존의 태스크 스케줄링 기법

태스크 스케줄링의 기본적인 아이디어는 다음과 같다.

태스크 순서화 과정 - *그래프에 존재하는 모든 태스크에 태스크의 속성에 따라 우선 순위를 부여하고, 우선 순위에 따라서 태스크 리스트가 작성된다.*

프로세서 선정 과정 - *리스트의 모든 태스크가 할당 될 때까지 다음 과정이 반복된다. 1)태스크 리스트에서 가장 우선하는 준비(ready) 상태의 태스크를 태스크가 실행될 유휴(Idle) 상태의 프로세서를 선정하여 할당한다. 2)태스크 리스트에서 부모 프로세스의 실행이 모두 종료된 태스크를 준비 상태로 바꾼다.*

통신비용이 보다 높은 비중을 차지하게 됨에 따라 통신비용에 대한 고려가 필요하게 되었고, 통신을 고려한 리스트 스케줄링 기법, 클러스터 기법, 태스크 복제기법 등이 제안되었다[1, 4]. 이들 기법들은 통신비용을 최소화하기 위해 불필요한 태스크의 병렬화(parallelism)를 제거하고 되도록 통신이 같은 프로세서에서 이루어지도록 한다. 하지만 이들 기법은 클러스터 시스템과 같은 메시지 패싱 기반의 환경에는 적합하지 않다.

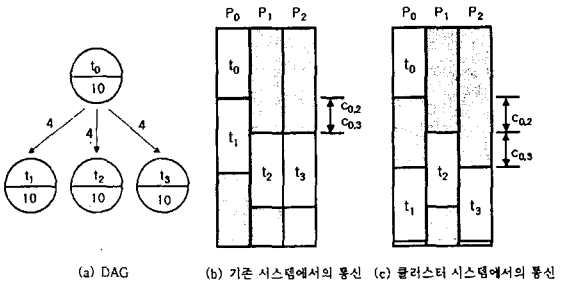


그림 1. 기존 시스템과 클러스터 시스템에서의 통신

메시지 패싱 환경에서는 다른 프로세서가 갖고 있는 데이터에 접근하기 위해 시스템이 제공하는 서비스를 이용한다. 따라서 통신은 자식 태스크가 할당된 프로세서와 부모 태스크가 할당된 프로세서 모두에 영향을 준다. 따라서 동기식의 통신이 순차적으로 이루어지며, 기존의 기법에서 사용한 한쪽에만 통신비용을 반영한 기법은 이러한 통신 특성을 반영하기에 부족하다.

3. 개선된 스케줄링 기법

메시지 패싱 시스템에서의 모델과 기존의 통신모델의 가장 큰 차이점은 순차적 통신에 의한 지연이 통신과 관계된 자식 태스크의 실행뿐만 아니라 다른 자식 태스크들의 실행까지 지연시킨다는 점이다. 그림 1.는 한쪽에만 통신비용을 고려한 전형적인 리스트 스케줄링 모델과 메시지 패싱을 사용하는 클러스터 시스템에서 통신의 차이점을 보여준다.

3.1 태스크 레벨의 개선

기존 리스트 스케줄링 기법에서 태스크에 우선 순위를 부여하기 위하여 사용된 특성중 가장 대표적인 것이 레벨이다. 어떤 태스크 t_i의 레벨은 t_i에서 종료 태스크까지의 가장 긴 경로의 길이이다[1, 2]. 기존 기법에서 사용된 통신비용을 고려한 레벨은 다음과 같이 정의된다.

$$level(t_i) = w_i + \max(c_{i,j} + level(t_j)),$$
$$t_j \in \{t_k \mid t_k \text{ is child of } t_i\} \quad \cdots\cdots(1)$$

최대값을 취한 이유는 각각의 통신이 다른 통신에 영향을 주지 않고 개별적으로 이루어지기 때문이다. 본 논문에서는 기존의 레벨을 순차적 통신을 고려하여 다음과 같이 개선하였다.

$$level(t_i) = w_i + \sum_{t_k} c_{i,k} + \max(level(t_k)),$$
$$t_k \in \{t_j \mid t_j \text{ is child of } t_i\} \quad \cdots\cdots(2)$$

자식 태스크들과의 통신비용의 합은 다수의 태스크와 통신이 이루어지는 경우 통신비용이 누적되는 것을 반영한다. 같은 프로세서에 태스크가 할당돼서 태스크간의 통신비용이 제거되는 것을 고려하지 않을 때, 태스크 t_i와 자식 태스크 t_j의 간에 생길 수 있는 가장 큰 통신비용은 모든 자식 태스크와의 통신비용의 합이기 때문이다.

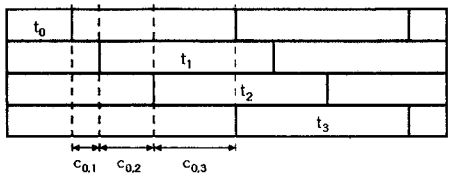
3.2 우선 순위 함수의 개선

그림 2.는 부모 태스크와의 통신비용이 $c_{01} < c_{02} < c_{03}$ 일 때, (a)는 t_1, t_2, t_3의 순서로 태스크가 할당된 경우이며 (b)는 t_3, t_2, t_1의 순서로 할당된 경우이다. 여기서 통신비용이 작은 태스크를 먼저 스케줄링하는 것이 통신비용이 큰 태스크를 먼저 스케줄링하는 것보다 전체적인 태스크 시작 시간에 유리하다는 것을 알 수 있다.

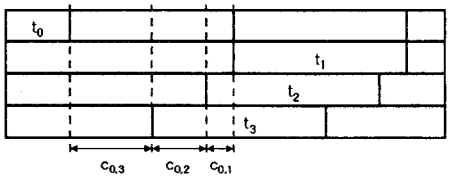
이러한 특성을 반영하기 위해서 본 논문에서는 태스크의 크기가 비슷한 경우 통신 시간의 길이가 짧은 태스크가 먼저 할당되도록 $SCF(t_i)$ 함수를 정의하였다.

$$C_j = \sum_{t_k} c_{j,k},\ t_k \in \{t_n \mid t_n \text{ is child of } t_j\} \quad \cdots\cdots(3)$$
$$SCF(t_i) = C_j - c_{i,j},\ t_j \text{ is parent of } t_i. \quad \cdots\cdots(4)$$

위 식에서 SCF(ti)의 값이 크면 ti의 통신비용은 작은



(a) 통신비용이 작은 태스크가 먼저 할당된 경우



(b) 통신비용이 큰 태스크가 먼저 할당된 경우

그림 2. 태스크 스케줄 순서에 따른 태스크 시작 시간의 변화

것을 의미하며, 이것은 tj의 다른 자식 태스크들이 ti와 tj의 통신으로 이해서 지연되는 시간이 줄어드는 것을 의미한다. 앞에서 정의한 $level(t_i)$와 $SCF(t_i)$에 의해서 본 논문의 알고리즘에서 사용되는 우선 순위 $priority(t_i)$는 다음과 같이 정의된다.

$$priority(t_i) = level(t_i) + SCF(t_i) \quad \cdots\cdots(5)$$

3.3 개선된 프로세서 선택 기준

프로세서 선정 과정에서는 태스크가 할당될 프로세서를 선정하는 과정으로 선택기준에 따라서 성능이 달라진다. 기존 연구에서는 프로세서 선택기준으로 태스크가 가장 빨리 시작될 수 있는 시간이 사용되었다[4]. 이 경우 같은 프로세서 내에서의 통신은 비용이 제거되기 때문에 부모 태스크가 할당된 프로세서가 가장 먼저 선택되는 경향이 있다. 하지만 본 논문에서는 순차적 통신을 가정하기 때문에 최초로 선정되는 태스크가 부모 태스크와 같은 프로세서에 할당되게 되면 부모 태스크의 다른 통신들이 끝날 때까지 태스크 실행이 지연된다. 따라서 태스크가 아직 할당되지 않은 자식 태스크를 갖는 부모 태스크와 같은 프로세서에 할당되는 것을 회피해야 한다. 새로운 프로세서 선택 기준은 다음과 같다.

프로세서 선택 기준 :

1) 태스크가 하나의 부모 태스크를 갖으며, 부모 태스크의 유일한 자식 태스크인 경우 부모 태스크와 같은 프로세서를 선택한다. 부모 태스크의 유일한 자식 태스크가 아닌 경우 부모 태스크가 할당된 프로세서를 제외한 프로세서 중 가장 빨리 태스크를 실행할 수 있는 프로세서를 선택한다.

2) 태스크가 둘 이상의 부모 태스크를 갖는 경우 현재 태스크를 유일한 자식 태스크로 갖는 부모 태스크가 할당된 프로세서가 유휴상태가 되는 시간과 가장 빨리 태스크를 실행할 수 있는 프로세서에서의 태스크 시작 시간 중 더 작은 값을 갖는 프로세서를 선택한다.

이러한 개선점을 포함한 개선된 리스트 스케줄링 기법은 기존의 기법이 통신비용의 누적을 고려하지 못하여 발생하는 비효율적인 스케줄링을 개선한다.

4. 실험 및 결과 분석

각 기법의 스케줄러는 C언어로 구현하였다. 태스크 스케줄링 기법의 평가를 위해 전체 어플리케이션이 실행되는 시간을 비교한다. 이것은 CP(critical path)상의 태스크들의 계산비용의 합에 대한 *makespan*(태스크들이 프로세서에 스케줄링 된 길이)으로 성능을 평가하며, 스케줄 길이 비율(Schedule Length Ratio, SLR)이라 불린다. *SLR*은 다음과 같이 정의된다.

$$SLR = \frac{makespan}{\sum_{t_i} w_i}, t_i \in CP \quad \cdots\cdots (6)$$

이 실험에 사용된 어플리케이션 모델은 가우스 소거법(Gaussian Elimination)의 태스크 그래프를 사용하였다.

4.1 프로세서 변화에 따른 성능 비교

성능의 비교를 위하여 통신비용이 고려된 리스트 스케줄링 기법과 비교하였다. 프로세서의 수를 4에서 20까지 증가시키면서 실험하였으며, 각기 200번의 시행을 통하여 평균값을 얻어냈다. 행렬의 크기는 16이며 *CCR*(communication to computa tion ratio)은 0.4이다.

그림 3.에서 기존기법이 보다 큰 프로세서 수에서 수렴하며, 보다 프로세서를 효율적으로 활용하는 것을 알 수 있다.

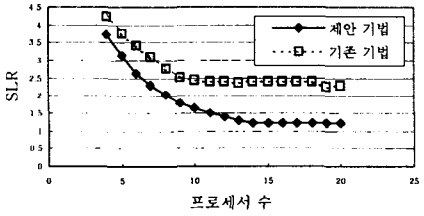


그림 3. 프로세서 수에 따른 SLR

4.2 태스크 규모에 따른 성능 비교

행렬 크기를 5에서 21까지(태스크 수 14에서 230개까지), *CCR*의 값이 0.15와 0.7이 되도록 하였다. 프로세서의 수는 8개라고 가정하였으며, 실험 결과는 각각 500번의 시행을 통하여 평균값을 산출하였다.

그림 4.에서 *CCR*이 0.15일 때 제안기법이 보다 짧은 실행시간을 보이며 행렬의 크기가 프로세서 수보다 적을 때 제안기법의 *SLR* 값은 1에 근접한다. 평균적으로 약 37% 짧은 스케줄 길이를 갖는 결과를 얻을 수 있었다.

그림 5.에서는 두 기법 모두 *CCR*이 작을 때보다 나쁜 성능을 보이지만 제안기법이 평균적으로 97% 짧은 스케줄링 결과를 보여준다. 또한 행렬크기가 프로세서 수보다 적을 때 *SLR*은 최적화에 가까운 성능을 보인다. 앞의 *CCR*이 0.15일 때와 비교하면, 두 기법 모두 *CCR*이 증가에 따라 성능이 저하되고 있지만, 성능감소가 기존 기법에 비하여 적은 것을 알 수 있다.

5. 결 론

본 논문은 동기식 통신이 순차적으로 발생하는 메시지

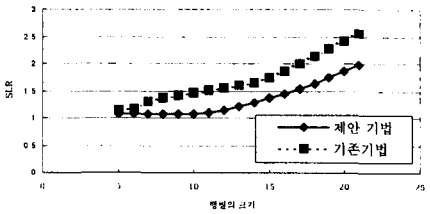


그림 5. *CCR*=0.15에서의 행렬의 크기 변화에 따른 *SLR*의 변화

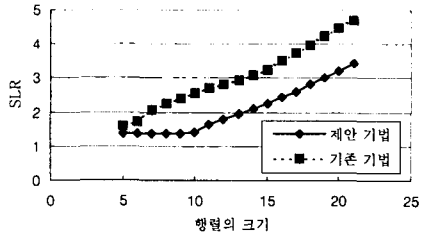


그림 4. *CCR*=0.7에서의 행렬 크기 변화에 따른 *SLR*의 변화

패싱 시스템에 적합하도록 리스트 스케줄 기법을 개선하였다. 개선된 우선 순위 함수에서는 기존의 레벨을 개선하여 통신비용이 누적되는 것을 고려하였으며, 프로세서 선정 과정을 개선하여 스케줄링 되지 않은 자식 태스크가 남아있는 경우 부모 태스크와 같은 프로세서로 태스크가 할당되는 것을 회피하도록 하였다. 두 가지 제안을 통하여 기존 리스트 스케줄링 기법이 통신비용이 누적되는 병렬/분산 시스템에 적용될 수 있도록 하였다.

실험결과 프로세서가 증가함에 따라 기존기법보다 짧은 스케줄링 길이를 갖는 결과를 얻을 수 있었고, 통신비용의 영향을 더 적게 받으며, 제안기법이 좀더 효율적인 스케줄링을 수행하는 것을 알 수 있었다. 따라서 메시지 패싱 환경의 통신 특성을 고려하지 않아 발생한 비효율적인 스케줄링이 개선되었음을 확인 할 수 있었다.

참고 문헌

[1] H. El-Rewini, T. G. Lewis and H. H. Ali, "Task Scheduling in Parallel and Distributed Systems", PTR Prentice Hall, USA, pp.17-81, 1994.

[2] Y.-K. Kowk and I. Ahmad, "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", ACM Computing Surveys, vol. 31, no. 4, pp.406-471, 1999.

[3] K. Hwang and Z. Xu, "Scalable Parallel Computing", McGraw-Hill, pp.13-36, 1998.

[4] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing", IEEE Transactions on parallel and distributed system. vol. 13, no. 3, pp.260-274, March 2002.

[5] T. Hsu and D. R. Lopez, "Task Allocation on a Network of Processors", IEEE Transactions on computers, vol. 49, no. 12, pp.1339-135, December 2000.