

그리드시스템을 위한 부하재분배 메커니즘에 관한 연구

이성훈^{1*}

¹백석대학교 정보통신학부

Study on Load Redistribution Mechanism in Grid System

Seong-Hoon Lee^{1*}

¹Dept. of Information&communication, Baekseok University

요 약 그리드에서 수행할 수 있는 많은 응용들에 대해 그리드시스템은 시스템내 작업들에 대해 낮은 사용율을 보이는 머신에 스케줄링 함으로써 자원 균형 효과를 제공할 수 있다. 각 작업들이 수행되기 위해 서로 통신을 하는 동안 그리드 스케줄러는 통신 트래픽 혹은 통신 거리의 최소화를 고려하여 작업들을 스케줄링 한다. 본 연구에서는 통신 트래픽 혹은 통신 거리를 감안한 부하재분배 알고리즘을 제안한다. 이를 기반으로 시뮬레이션을 수행하였다. 실험 결과에서는 전반적으로 기존의 방법보다 반응시간에서 우수함을 보였다.

Abstract For applications that are grid enabled, the grid can offer a resource balancing effect by scheduling grid jobs on machines with low utilization. When jobs communicate with each other, the internet, with storage resources, an advanced scheduler could schedule them to minimize communications traffic or minimize the distance of the communications. We propose an load redistribution algorithm to minimize communication traffic and distance of the communication using genetic algorithm. The experiments show the proposed load redistribution algorithm performs efficiently in the variance of load in grid environments.

Key Words : Grid System, Resource Allocation, Genetic Algorithm, Scheduling

1. 서론

인터넷이 보편화되고 컴퓨터 및 네트워크의 성능이 향상됨에 따라 분야에 따라 컴퓨터를 이용하는 양상은 변화되고 있다. 특히 분산처리 기술 분야의 변화로는 통신망을 통해 서로 연결된 컴퓨터들을 하나의 문제를 해결하는데 사용할 수 있도록 하고 있다. IBM의 기술 자료에 따르면 가장 간단한 초기의 그리드 개념은 동종의 분산 시스템(homogeneous distributed system)으로 여겨졌지만, 현재는 자원통합에 있어 동일기종뿐만 아니라 서로 다른 컴퓨터 자원들과 대용량 저장장치 등을 포함하여 자원통합을 지향하고 있으며 이를 진정한 의미의 그리드(GRID)라는 용어로 불리고 있다[6,8]. 그리드 개념의 확대에는 그리드 시스템에 존재하는 다양한 기기종의 기계들뿐만 아니라 다양한 어플리케이션, 각종 디바이스들이 포함될 것이며 이를 이용하는 사용자 또한 급증할 것으로 보인

다[5,6,10]. 그리드 컴퓨팅 환경의 특징은 다음 3가지 특성을 지닌다.

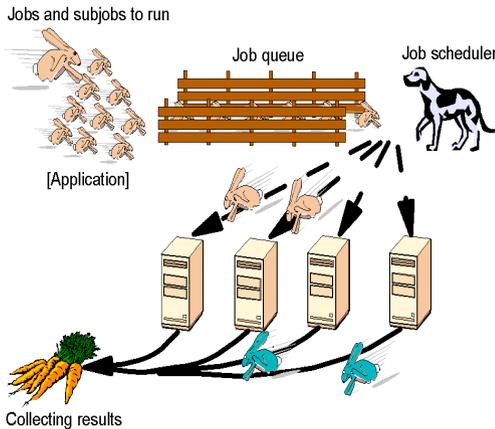
- 1) 정상적으로 어플리케이션이 수행되는 기계는 예외적인 액티비티(activity)들로 인하여 비정상적으로 과부하(busy)인 상태에 있을 수 있기 때문에 그리드 시스템내 저부하(idle) 기계에서 수행될 수 있어야 한다.
- 2) 자원의 균형(resource balancing)의 측면에서 그리드는 낮은 활용률을 보이는 기계에 그리드 작업(job)들을 스케줄링 함으로써 자원의 균형 효과를 극대화할 수 있다.
- 3) 하나의 어플리케이션이 여러 작업들로 나누어져 동시에 병렬처리 특성을 지니며 그리드 스케줄러는 최적의 결과를 얻기 위해 그림 1처럼, 많은 작업들을 서로 다른 기계에 스케줄링 할 수 있다.

*교신저자 : 이성훈(shlee@bu.ac.kr)

접수일 09년 07월 30일

수정일 09년 08월 30일

게재확정일 09년 09월 16일



[그림 1] 그리드 스케줄러

본 연구에서는 그리드컴퓨팅 환경에서 각 시스템의 부하를 균등하게 유지하기 위한 방법을 연구한다. 지금까지의 관련 연구는 매우 미진한 상태이며 또한 전통적인 부하분산과 관련된 대부분의 연구들은 초기 그리드 의미인 동일기종이거나 동일한 서비스 유형등에 국한되어 왔다. 그리드환경에서의 사용자들은 요구하는 서비스 유형이 매우 다양하며, 또한 서로 다른 이기종간에도 서비스가 이루어져야만 한다. 본 연구에서는 그리드컴퓨팅 환경에서 유전정보를 기반으로 하는 부하재분배방법에 대해 연구하였다. 2장에서는 본 논문에서 제안하는 유전정보를 활용한 부하재분배 알고리즘등을 기술하였고 3장에서는 관련된 실험을 하였다. 마지막으로 4장에서 결론을 기술하였다.

2. 그리드환경에서의 부하재분배

본 장에서는 유전정보를 이용한 그리드 컴퓨팅환경에서 부하분산을 위해 필요한 내용들을 기술한다. 먼저, 각 프로세서의 부하정도를 판단하기 위한 부하척도, 프로세서에 대한 코딩방법, 적합도 함수, 연구의 주 내용인 유전 정보 기반의 부하분산 방법에 대해 기술하고자 한다.

2.1 부하척도

초기 그리드 모형에서의 부하를 측정하기 위해 사용된 척도로는 큐(queue)에 들어있는 작업들의 개수를 이용하였다. 하지만 현재의 그리드 모형의 가장 큰 특징은 동일한 TASK에 대해서도 각 프로세서들이 이질성을 갖기 때문에 다른 처리시간을 보인다는 점이다. 따라서 본 연구에서 사용하는 부하척도는 다음 식 (1)과 같이 정의하

여 사용한다.

$$VQL = T_{no} \times P_{tu} \quad (1)$$

매개변수 T_{no} 는 큐에 들어있는 TASK의 개수를 의미하고 P_{tu} 는 각 프로세서에서 동일한 하나의 TASK를 처리하는 소요시간을 의미한다.

2.2 코딩방법

그리드 시스템내에 존재하는 프로세서의 개수가 n 개 일 때 i 번째 프로세서는 p_i 로 표현되며 i 는 $0 \sim n-1$ 의 범위 값을 갖는다. 본 연구에서는 개체표현 방법으로 이진 코딩 방법을 사용하며, 따라서 하나의 개체는 이진코드 벡터(binary-coded vector)로 정의할 수 있으며 $\langle v_0, v_1, v_2, \dots, v_{n-1} \rangle$ 과 같이 표현할 수 있다. 각각의 v_i 값은 "0" 혹은 "1"의 값을 갖는다.

2.3 알고리즘 개요

그리드 시스템의 부하재분배를 위하여 시스템내 모든 프로세서의 부하상태를 3단계로 분리하며, 이들은 각 프로세서의 VQL 로 결정된다. 이같은 부하표현 내용은 표 1과 같다.

[표 1] 부하표현

부하상태	의미	기준
light (idle)	저부하	$VQL_i \leq T_{low}$
middle	정상부하	$T_{low} \leq VQL_i < T_{up}$
heavy (busy)	과부하	$VQL_i \geq T_{up}$

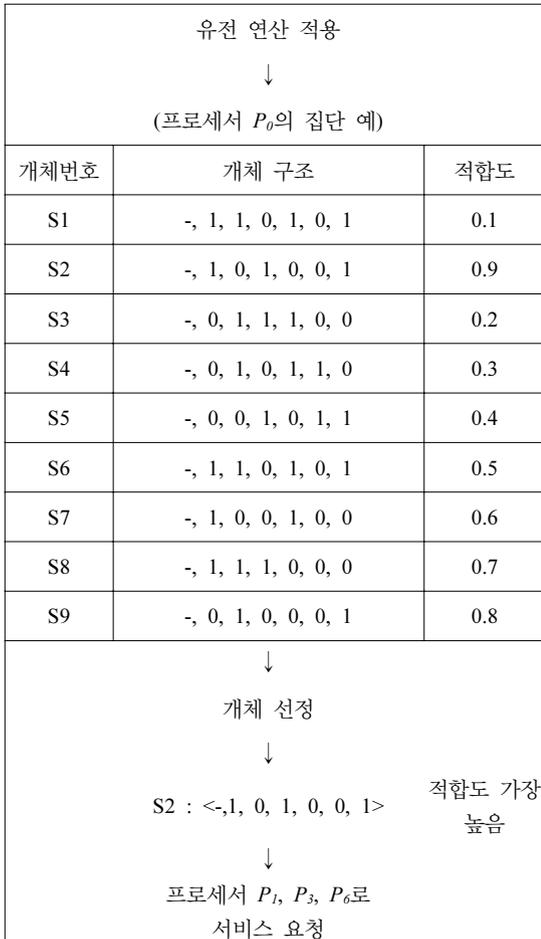
저부하 상태인 프로세서는 자신의 VQL 값이 적기 때문에 다른 과부하 상태인 프로세서가 가지고 있는 TASK를 이전받을 수 있는 상태이며, 반대로 과부하 상태인 프로세서는 자신의 TASK를 다른 프로세서로 이전해 줄 수 있는 상태이다. 3단계 부하상태를 유지하기 위해 2개의 임계값 즉, 상한(T_{up}) 및 하한(T_{low}) 임계값을 이용하였다.

한 집단에 포함된 각 스트링은 적합도 함수 식 (2)로 평가되며 유전 연산자가 적용된 하나의 집단 내 모든 스트링 중에서 가장 적합도가 높은 스트링이 선정되어 선정된 스트링의 내용 즉, 해당 스트링 내 "1"로 설정된 비트에 대응되는 프로세서로 TASK 이전을 위한 이전요청

메시지를 보내게 된다. 이같은 처리개요는 다음 그림 2와 같다.

$$F_i = \left(\frac{1}{\alpha \times TMPT + \beta \times TMTT + \gamma \times TPTT} \right) \quad (2)$$

[식 2]에서 *TMPT*는 전송될 요청 메시지들의 처리시간으로서 전송될 요청메시지의 개수와 각 메시지 처리시간의 곱으로 정의된다. *TMTT*는 송신자로부터 선정된 스트링내 “1”로 설정된 각 비트에 대응되는 수신자대상 프로세서들까지의 메시지 전송시간의 합으로 정의하며 *TPTT*는 선정된 스트링내 “1”로 설정된 각 비트에 대응되는 수신자 대상 프로세서들에서 타스크를 처리하는데 소요되는 시간으로 각 프로세서에서의 타스크 처리시간은 난수 발생기에 의해 결정된 임의수로 가정하였다.



[그림 2] 알고리즘 처리 개요

3. 실험 환경

연구내용에 대한 실험을 위해 그리드 시스템에 대하여 다음과 같은 내용을 가정하였다. 먼저 그리드시스템에 24개의 이질형 프로세서가 존재하며 각 프로세서로부터 다른 프로세서로의 메시지 전송 및 타스크 전송시간은 2차원 배열로 표현하였다. 만일 3개의 프로세서를 대상으로 나타낸다면 다음 표 2와 같으며 각 원소의 값은 난수 발생기로부터 생성되는 임의의 수로 하였다.

[표 2] 처리시간

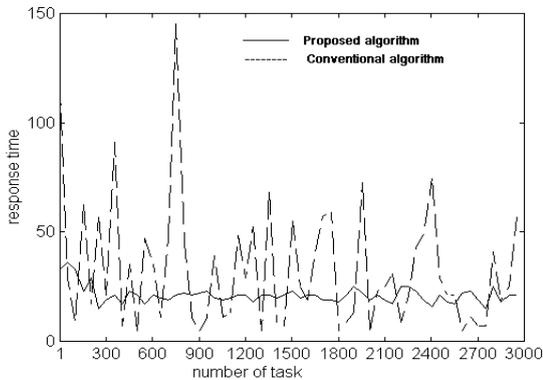
	송신자				
	수신자				
	P_0	0	1	3	2
	P_1	1	0	3	2
	P_2	3	2	0	1
	P_3	3	3	1	0

유전알고리즘의 적용 및 적합도 함수에 나타난 매개변수들에 대한 파라메타 값은 다음 표 3과 같이 설정하여 실험을 수행하였다. 선택 및 돌연변이 확률은 많은 응용들에서 적절한 것으로 인용되고 있다[3].

[표 3] 유전알고리즘 관련 파라메타 값

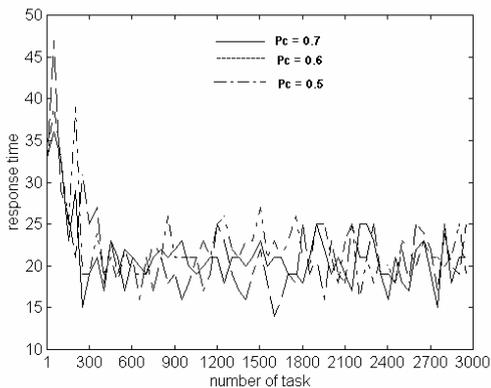
가중치(α)	0.5	선택확률(P_c)	0.7
가중치(β)	0.15	돌연변이확률(P_m)	0.05
가중치(γ)	0.05	집단수	50

다음 실험 결과는 표 2와 표 3을 기반으로 하여 타스크 개수가 3000개일 때, 기존의 방법과 제안 알고리즘과의 비교 결과를 보인 것이다. 그림 3에서 알 수 있듯이 기존 방법은 임의의 기반임으로 인해 많은 요동을 보이는 반면에 제안된 알고리즘에서는 최적해로의 수렴성을 보이므로 요동성이 아주 미미하다.



[그림 3] 실험 결과

본 실험은 교배비율(p_c)에 따른 결과를 파악하고자 한 실험 내용으로서 돌연변이 비율(p_m)이 0.05일 때 결과는 다음 그림 4와 같다.



[그림 4] 교배비율 변화에 따른 실험 결과

위 결과에서 보여주는 것처럼 각 구간별로 약간의 차이는 있지만 전체적으로 교배비율이 0.7일 때가 가장 좋은 결과를 보이는 것으로 조사되었다.

4. 결론

본 연구에서는 그리드 환경에서의 자원 활용성을 높이기 위해 유전 알고리즘을 기반으로 하는 부하재분배 문제를 다루었다. 이를 위해 부하척도 및 코딩방법, 유전알고리즘 기반의 부하재분배 메커니즘등을 제시하였다. 제안된 내용의 효율성을 위해 기존의 임의 방식 스케줄링 방법과의 비교/분석을 위한 실험결과, 제안 알고리즘이 유전알고리즘의 특성을 반영하여 최적해로의 수렴성을

보이고 있다.

참고 문헌

- [1] D.L.Eager, E.D.Lazowska, J.Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," *IEEE Trans on Software Engineering*, vol.12, no.5, pp.662-675, May 1986.
- [2] N. G.Shivaratri, P.Krueger, and M.Singhal, "Load Distributing for Locally Distributed Systems," *IEEE COMPUTER*, vol.25, no.12, pp.33-44, December 1992.
- [3] J.Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," *IEEE Trans on SMC*, vol. SMC-16, no.1, pp.122-128, January 1986.
- [4] T. Kunz, "The Influence of Different Workload Descriptions on a Heuristic Load Balancing Scheme," *IEEE Trans on Software Engineering*, vol.17, No.7, pp.725-730, July 1991.
- [5] Y. Li, Z. Lan, "A Survey of Load Balancing in Grid Computing," LNCS 3314, pp. 280-285, Dec 2004.
- [6] B. Jacob, M. Brown., *Introduction to grid computing*. ibm.com/redbooks.
- [7] Thomas G. R., Ten Reasons to Use Divisible Load Theory. *IEEE Computer* Vol. 36, No. 5, pp. 63-68, 2003.
- [8] B. Jacob, M. Brown, K. Fukui, "Introduction to Grid Computing," IBM Redbooks, <http://ibm.com/redbooks>.
- [9] <http://www.cs.wisc.edu/condor>.
- [10] <http://www.gridforum.org>.
- [11] <http://www.gridforumkorea.org>.
- [12] <http://www.globus.org/>.

이 성 훈(Seong-Hoon Lee)

[정회원]



1995년 2월 : 고려대학교 석사
 1998년 2월 : 고려대학교 박사.
 1998년 3월 ~ 현재 : 백석대학교 교수

<관심분야>

분산시스템, 그리드 시스템, 웹서비스, 유전알고리즘 등.