Grid환경기반 자원 선택에 관한 연구

노남수° 홍필두 장택수 이용우
서울시립대학교 전자전기컴퓨터공학부
{topnons°, iamhpd, wichid}@metalab.uos.ac.kr, ywlee@uos.ac.kr

A Study on the Resource Brokering in Grid

Namsu No°, Pildu Hong, Taeksoo Jang, Yongwoo Lee
Faculty of Electrical & Computer Engineering, The University Of Seoul

요 약

그리드 시스템은 기존의 클러스터링과는 달리 지역적으로 분산되어 있는 컴퓨팅 자원을 네트워크로 상호 연동하여 사용하는 시스템이다. 그러므로 사용할 자원의 조합의 선택에 따라 그리드 컴퓨팅 성능은 큰 영향을 받는다. 그러나 현재 대표적인 Grid Middleware Toolkit인 Globus에서는 자원 선택에 관련해서는 별다른 방안을 제시하고 있지 않다. 이와 관련하여 본 논문에서는asymmetric MPI application에 대해 효율적인 자원을 선택하기 위한 적절한 cluster조합의 선택기법을 제시한다.

1. 서 론

병렬처리 개념은 큰 작업을 여러 개의 작은 작업으로 나누어 여러 자원에 동시에 할당함으로써 궁극적으로 성능과 효율성을 극대화 하는데 목적이 있다. 이에 대한 하나의 아키텍쳐로 메시지패싱 방식의 대표적인 클러스터링을 들수있다. 그리고 메시지패싱 라이브러리로는 MPI가 널리 사용되고 있다[5]. 그러나 클러스터링은 local area network상에서 동일 자원의 연결을 하는 방법으로 해당 클러스터 상에서만의 작업 수행이라는 비효율적인 측면이 존재한다. 이에 대한 해결법 및 사용자의 작업을 효과적으로 수행하기 위해 그리드라는 개념이 도입되었고, 이에 대한 활발한 연구가 진행되고 있다. [1]

그리드 시스템은 지리적으로 분산되어 있는 고성능 컴퓨팅 자원을 네트워크로 상호 연동하여 사용할 수 있는 구조와 방법론을 의미한다. 현재 그리드 미들웨어로Alchemi, Gridbus, Legion, Globus등 몇가지 Toolkit이 제시되어있다. 이중 현재 그리드 개발 과제에서 가장 많이 사용되는 미들웨어는 Globus[6]이다.

Globus Toolkit에서 제공하여 주는 핵심 서비스는 크게 Information Service, Resource Management, Security, Data Management 등이 있다[2]. 그런데, 유효한 자원들 중 어느 자원들을 선택해서 job을 실행 시킬지 결정하는 resource broker에 대한 부분은 Globus Toolkit의 취약 부분이다. 하지만 wide area network상에서 서비스 자원을 사용하는 그리드 시스템의 특성상 그리드 컴퓨팅은 네트워크 상태 및 자원의 상태에 따른 적절한 자원의 선택은 컴퓨팅 성능에 큰 영향을 주는 중요한 요인이 된다.

이와 관련하여 본 논문에서는 Asymmetric MPI algorithm[3]으로 동작하는job에 대해 네트워크와 시스템의 상태에 따른 자원 선택 기법을 제시하였다.

2. 고려사항

2.1 가정

그리드 시스템에서의 job 수행은 wide area network상에서 수행된다. 그래서, 일반적으로, 서비스 자원과 root 노드 사이의 network 상태가 컴퓨팅 성능을 좌우한다. 하지만, 모든 경우가 이에 해당하지는 않는다. 수행하는 job의 특성이 네트워크 성능에 영향을 별로 받지 않을 경우, 더 나은 시스템을 보유한 클러스터로의 선택이 더 우수한 성능을 보이기도 했다.[4] 즉, 자원 선정시 수행할 job의 특성을 고려해야한다. 그러나 job의 특성을 시스템이 임의로 판단하기란 어렵다. 그래서 이번 논문에서는 job의 dominant특성을 job실행자로부터의 입력을 가정한다.

이 논문에서 제시하는 resource selection방법은 asymmetirc MPI algorithm에 대해서 고려한다. asymmetric MPI algorithm은 2.2섹션에서 간단히 다루는데, 2가지 단계로 operation이 일어난다(root와 coordinators, coordinators와 cluster노드들). 여기서 root와 coordinators는 WAN, coordinators와 cluster노드들은 LAN구간으로 네트워크 측면에서는 WAN구간이 dominant로써 각 cluster들 LAN 네트워크의 성능은 동일하다고 가정한다.

2.2 asymmetric & symmetric MPI algorithm

asymmetric & symmetric algorithm이란 MPI로 작성된 application이 clusters상에서 collective operations이 어떤 방식으로 동작하는지에 대한 algorithm으로 아래와 같다.

asymmetric MPI algorithm이란 MPI로 job을 수행 시킬 때 하나의 전담 프로세스(이하 root process라 칭함) 가 sender와 receiver의 역할을 수행하며, 또한 cluster상에서 coordinator의 역할을 하는 것을 말한다.

symmetric MPI algorithm이란 모든 프로세스들이 send와 receive를 수행하는, 즉, 모두 동급의 peers로써 취급되어지는 것을 말한다.[3]

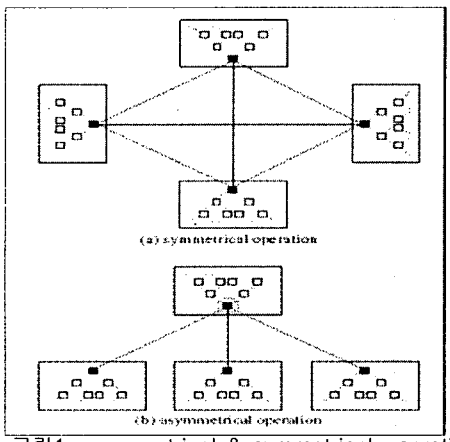

그림1. asymmetrical & symmetrical operation

3. Cluster Selection

. 이 논문에서는 최적의 조합을 찾아내기 위해 Greedy Search Alogrithm을 사용하였다. 이 경우 다음과 같이 Cluster의 Selection을 한다

- 최적의 cluster조합을 찾아낸다.
- greedy search algorithm에서는 cluster조합의 경우의 수가 최대 n! 이므로, 유효 cluster의 수가 증가함에 따라 기하급수적으로 cluster조합의 경우의 수가 증가한다. 따라서, 최대 clusters조합에 대한 제약조건이 필요하다.

cluster selection을 위해 사용되는 parameter들은 다음과 같다.

- user factor α, β

α 는 수행할 프로그램의 네트워크 의존도, β는 시스템 의존도를 의미하며, α+β=1 의 관계를 가진다.

- Greedy Search Cost $f(n)$

  f(n) = α * g(n) + β * h(n)

  g(n) = 1/network_factor(N)

  h(n) = 1/utilization_factor(N)

  network_factor(N) = normalized Max latency among n and its parents

  utilization_factor(N) = normalized Max CPU and Memory Utilization among selected clusters' nodes

  n : test node from greedy search tree

  N : set of n and its parents

- branching limit : MaxClusterNum, node number

  MaxClusterNum = ceiling(α*nNum + β*sNum)

  nNum = Total clusters' number capable of node number ordered by latency

  sNum = Total clusters'number capable of node number ordered by utilization

MaxClusterNum은 앞에서 언급한 최대 Cluster들 조합을 제한하기 위한 최대 cluster조합 수(Greedy Search Tree 의 depth 제한요소)를 의미하며, node number는 사용자가 입력한 요구 processor의 수를 의미한다.

위의 parameter들을 이용하여 Greedy Search Algorithm을 적용한 tree의 depth가 MaxClusterNum범위 내에서 최종 도달하는 fringe들 중에 최적의 우선순위를 가지는 cluster 조합을 반환한다.

4. 성능 모델링 및 분석

3장에서 제시한 Cluster Selection의 성능평가를 위해 인위적인 5개의 clusters(표1)와 cluster에서의 실행시간의 상대적인 평가의 기준이 되는 요구 할당 processor 수별 실행시간을 Amdahl's Law에 근거하여 표2 와 같이 모델링 하였다.

이번 실험에서는 프로그램의 특성상 network의존도가 더 큰 프로그램과 utilization의 의존도가 더 큰 프로그램 (각각 (α , β) = {(0.8,0.2), (0.2,0.8)} 에 대해 시뮬레이션을 하고, 사용자가 임의로 랜덤한 조합을 선택했을 경우와 비교분석 하였다.

Cluster	Capacity	network_factor	utilization_factor
A	1	0.7	0.6
B	3	0.6	0.7
C	4	0.65	0.5
D	6	0.8	0.45
E	7	0.5	0.55

표1. Cluster Characteristic

No. Processors	1	3	5	7	10
Time	100	40	28	23	19

표2. Processor수별 실행시간 기준표

그림2는, 사용자의 요구 프로세서의 수가 3,5,7,10으로 변할 때, 상대적으로 네트워크에 의존도가 높은 수행프로그램의 실행시간을 보여주고 있다. 그림3 는 상대적으로 네트워크의 영향을 덜 받는 작업에 대한 실행시간 결과 값이다. 그림2를 보면 일반적으로 작업을 분배할 노드의 수가 많을수록 실행시간이 감소하지만, 노드의 수가 10일 때 오히려 더 증가함을 볼 수 있다. 이는 필요노드의 수를 만족 시키기도록 cluster를 선택해야 하므로 오히려 네트워크 성능이 더 안 좋은 cluster가 추가됨이 원인이다. 그림3의 경우 조합의 cluster의 수가 필요 노드수의 증가에 따라 증가하더라도 네트워크의 영향을 덜 받는 작업이므로 분배할 노드의 수가 증가함에 따라 원만한 성능향상을 보임을 볼 수 있다. 각각의 경우에 대해 사용자가 임의로 조합을 선택했을 때와의 성능비교를 보면 (α, β) = (0.8,0.2) 의 경우 요구 노드의 수별로 <15%, 25%, 34.3%, 23.5%>, (α, β) = (0.2,0.8) 의 경우 <6%, 2.4%, 7.8%, 7.2%> 정도의 성능향상이 이루어 졌다. 즉, 네트워크의 의존도가 높은 작업일수록 본 논문에서 제시한 cluster selection기법은 더 높은 성능을 보인다.

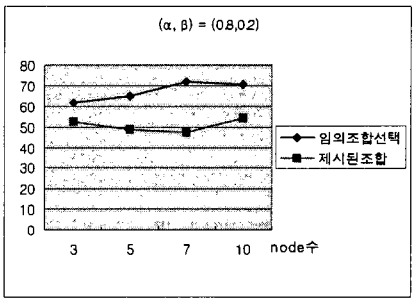

그림2. (α, β) = (0.8,0.2)일 때, 시뮬레이션 결과값 : 임의조합과 제시된 조합의 결과 값 비교

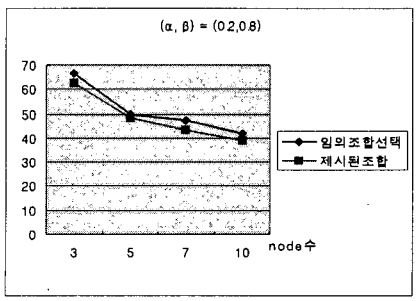

그림3. (α, β) = (0.2,0.8)일 때, 시뮬레이션 결과값 : 임의조합과 제시된 조합의 결과 값 비교

5. 결론 및 향후과제

본 논문에서는 MPI가 asymmetric algorithm으로 수행될 경우에 대해, wide area network에 분산되어 있는 유효 cluster자원의 최적의 조합을 찾기 위해 프로그램의 특성, network, cluster utilization을 적용한 cluster selection기법을 제시, 모델링 및 분석하였다.

하지만 본 논문에서 제시한 clusters selection 기법은 root와 coordinators 사이의 network latency를 기반으로 하기 때문에 symmetric algorithm에 대해서는 적절할 cluster selection이 되지 않을 수 있다. 즉, 본 논문에서 제시한 방법은 asymmetric MPI algorithm이라는 제약 요소가 있고 더불어 user factor에 대한 능동적 접근 및 기준제시가 요구된다. 이와 관련하여 향후 더욱 일반화 할 수 있는 자원 선택 algorithm에 대해 연구할 계획이다.

참고문헌

[1] I.Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid: Enabling Scalable Virtual Organizations,"Journal of High-Performance Computing Applications, Vol.15, 2001

[2] http://www-unix.globus.org/toolkit

[3] Thilo Kielmann, Rutger F. H. Hofman, Henri E. Bal, Aske Plaat, Raoul A. F. Bhoedjang "MAGPIE: MPI's Collective Communication Operations for Clustered Wide Area Systems." Symposium on Principles and Practice of Parallel Programming, Atlanta, GA, 1999

[4] 윤상용, "A study on the Resource Allocation and Division for a Cluster GRID",2002

[5] Message Passing Interface Forum, MPI: A Message Passing Interface Standard. International Journal of Supercomputing Applications, 1994

[6] I.Foster and Carl Kesselman, Globus : A metaco-mputing infrastructure Toolkit, Intl J, Supercomputer Application, 1997