

그리드 환경에서 서버 가상화를 이용한 작업 스케줄링 기법

김오범^{*1}, 박지수², 손진곤¹

¹한국방송통신대학교 평생대학원 정보과학과

²고려대학교 일반대학원 컴퓨터교육과

ohbeomkim@gmail.com^{*1}, bluejisu@korea.ac.kr², jgshon@knou.ac.kr¹

Job Scheduling Mechanism using Server Virtualization in Grid Environments

Oh Beom Kim^{*1}, Ji Su Park², Jin Gon Shon¹

¹Dept of Computer Science, Korea National Open University

²Dept of Computer Science Education, Korea University

요 약

그리드 컴퓨팅은 분산된 자원을 하나로 묶어 거대한 시스템을 구성하며 자원을 공유하여 성능을 높이고 비용을 절감 할 수 있는 시스템이다. 그리드 컴퓨팅 환경에서 사용되는 노드는 높은 성능을 가지고 있지만 단일 운영체제가 모든 하드웨어 자원을 제어하기 때문에 작업 진행시 부하가 높을 때 유휴자원을 쉽게 활용 할 수 없는 문제점이 발생한다. 본 논문에서는 서버에 가상화된 자원을 효율적으로 사용할 수 있도록 가상화를 적용하고 구성된 가상환경에 작업 스케줄링을 대행 할 수 있는 마스터노드를 이용하는 작업 스케줄링을 제안한다.

1. 서론

오늘날 과학 기술은 고성능 계산 자원을 사용하여 복잡한 문제를 해결한다. 이러한 요구를 충족시키기 위하여 지리적으로 분산된 고성능 컴퓨팅 자원을 하나로 묶어 사용할 수 있는 그리드 컴퓨팅이 연구되었다[1,2,3]. 그리드 컴퓨팅은 많은 수의 노드로 구성되기 때문에 작업 진행에 따른 노드관리가 중요하다. 그리드 컴퓨팅에 사용되는 서버는 높은 성능을 가지고 있지만 하나의 운영체제가 가상화된 자원을 제어하기 때문에 유휴자원을 쉽게 사용할 수 없는 문제점이 발생한다. 본 논문에서는 서버 자원을 효율적으로 사용하기 위해서 서버 가상화를 구성하고 구성된 가상자원에 작업 스케줄링을 대행 할 수 있는 마스터노드를 이용하는 작업 스케줄링을 제안한다. 작업 스케줄링은 마스터노드 작업 스케줄링과 마스터노드 작업 스케줄링을 관리하는 메타 스케줄링으로 구분한다. 구성된 마스터노드는 작업분배 및 결과수집을 대행하기 때문에 그리드 작업 스케줄링 효율을 증가 시킬 수 있다. 또한 가상화를 사용하기 때문에 유휴자원 문제를 해결 할 수 있다.

2. 관련 연구

2.1 그리드 작업 스케줄링

그리드 컴퓨팅에서 사용되는 스케줄링은 크게 전체작업을 분배하는 메타 스케줄링과 한 개의 노드 내에서 작업을 분배하는 작업 스케줄링으로 구분된다. 이러한 작업 스케줄링은 작업을 처리하는 관점에 따라서 Centralized

스케줄링과 Distributed 스케줄링으로 구분된다.

Centralized 스케줄링은 중앙 집중화된 모델 방식으로 작업대기열에 제출된 작업내용을 구분하여 작업분배 및 결과수집을 중앙의 하나의 메타스케줄러에서 모두 처리하는 방식으로 작업관리가 편리하고 노드구성이 쉽다는 장점을 가지고 있다. 단점은 중앙에서 메타스케줄러가 모든 내용을 처리하기 때문에 작업진행 중 네트워크 장애로 인해 통신이 차단될 경우 병목현상이 발생한다[4].

Distributed 스케줄링은 Centralized 방식과는 다르게 중앙의 하나의 메타스케줄러만을 사용하는 것이 아니라 각 노드에 메타스케줄러가 존재하여 작업을 로컬에서 처리하기 때문에 하나의 노드에서 발생하는 장애가 전체 시스템에 영향을 주지 않으며 작업에 따른 병목현상을 축소할 수 있는 장점이 있다. 하지만 구성이 복잡하며 동기화하기 어려운 단점을 가지고 있다[4].

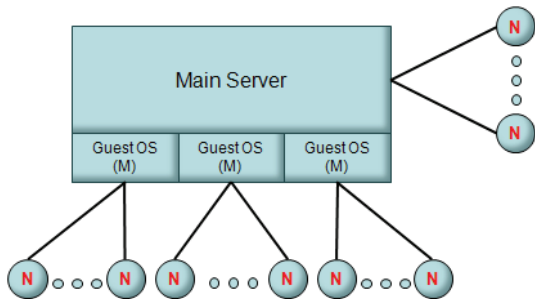
2.2 가상화

가상화 기술은 운영체제에 한정된 자원을 더 효율적으로 사용하기 위해서 발전한 기술로 VMM(Virtual Machine Monitor)을 통해서 실제 물리적인 컴퓨팅 자원과 사용자의 작업영역을 분리하여 사용할 수 있도록 하는 기술이다. 가상화 기술은 크게 OS-Level가상화, 전가상화(full-virtualization), 반가상화(para-virtualization), 가상화를 지원하는 하드웨어 기법으로 나눌 수 있다 [5,6]. 그리드 컴퓨팅은 분산된 이기종 자원을 사용하기 때문에 자원관리 문제가 중요하다. 자원관리에 중요성이

강조되면서 그리드 컴퓨팅에서도 동적 자원 환경의 개념을 적용하여 가상 작업영역을 그리드 아키텍처에 맞게 구현하는 시스템이 연구되었다[7,8].

3. 서버 가상화를 이용한 마스터노드 구성 환경

서버에 가용된 자원을 사용하는 운영체제는 프로세스 생성시 동일한 크기의 자원을 할당하게 된다. 프로세스가 자원을 확보하고 싶은 경우에는 운영체제에 자원 사용권한을 부여받아 자원량을 조절할 수 있다. 운영체제는 여러 개의 프로세스를 관리하기 때문에 전체 자원을 보호해야 한다. 이처럼 자원을 보호하기 위해서 자원사용량을 조절하기 때문에 유휴자원이 발생하게 된다. 그리드 컴퓨팅 또한 단일 운영체제를 사용하기 때문에 유휴자원이 발생한다. 발생된 유휴자원을 사용하기 위해서 가상화를 구성하고 구성된 가상자원에 작업 스케줄링을 대행 할 수 있는 마스터노드를 (그림 1)과 같이 구성한다.



(그림 1) 가상화를 이용한 마스터노드 구성 환경

4. 서버 가상화를 이용한 작업 스케줄링

가상환경을 통해서 마스터노드를 구성하게 되면 기존 그리드 컴퓨팅에서 사용하는 글로벌 스케줄링과 로컬 스케줄링을 변경해야 한다. 제안된 스케줄링은 마스터노드에서 실행하는 작업 스케줄링과 마스터노드를 관리하는 메타 스케줄링으로 구분한다.

4.1 마스터노드 관리를 위한 메타 스케줄링

메타 스케줄링은 연결된 노드를 마스터노드 중심으로 스케줄링하는 방식이다. 작업분배시 작업을 처리 할 수 있는 마스터노드를 검색하고 검색 결과에 따라서 작업을 분배하게 된다. (그림 2)는 최적 마스터노드 검색 알고리즘이다. 최적 마스터노드 선택 알고리즘은 작업 수행에 따른 워크플로우 정보(WorkflowInfo WFIF), 서버에 연결된 작업노드 정보(WorkNodeList WNL), 서버에 구성된 마스터노드 정보(MesterNodeList MNL)를 입력 값으로 갖는다. 마스터노드에 작업노드를 연결하기 전에 MNL 정보를 WFIF와 비교하여 예약이 가능하다면 예약(Requisition(MNL))을 접수하고 작업을 진행한다. 예약이 불가능할 경우는 WNL을 순서대로 WFIF와 비교(NodeSorting(WFIF,WNL))하여 그룹화 대상일 경우 그룹노드(GN)에 저장한다. 저장이 완

료되면 가상연결(VirtualConnection(WNL))을 수행한 후 MNL을 갱신한다.

```

AllocateMasterNodeCluster( WFIF,WNL,MNL )
① INPUTE: WorkflowInfo WFIF, WorkNodeList WNL
           ,MesterNodeList MNL
② GroudNode GN
③ WHILE ( MNL not empty ) DO
④   if( MNL == WFIF )
⑤     Requisition( MNL )
⑥     break
⑦   end if
⑧ END WHILE
⑨ if( !RequisitionCheck( MNL ) )
⑩   WHILE ( WNL not empty ) DO
⑪     if( NodeSorting( WFIF, WNL ) )
⑫       GN = WNL
⑬       VirtualConnection( WNL )
⑭     end if
⑮   END WHILE
⑯ end if
⑰ Update( MNL, GN )
    
```

(그림 2) 최적 마스터노드 검색 알고리즘

4.2 마스터노드 작업 스케줄링

작업을 할당 받은 마스터노드는 연결된 작업노드에 작업 분배 및 결과수집을 수행한다. 결과수집을 통해서 작업이 완료되면 메인서버에 작업결과를 갱신한다. 작업 스케줄링은 (그림 3)과 같이 마스터노드 작업 스케줄링 알고리즘을 통해서 실행된다. 마스터노드 작업 스케줄링 알고리즘은 워크플로우 정보(MasterWorkInfo MWIF), 연결된 작업노드 정보(GroudNode GN)를 입력 값으로 갖는다. 작업 확인(Work(GN))을 통해서 작업을 확인하고 완료된 작업을 MWIF에 저장한다. 작업이 미완료일 경우 미완료저장(AddICQueue(MWIF,GN))을 사용하여 미완료 작업 Queue에 저장하고 작업실행을 갱신한다. 모든 작업이 완료되면 마스터노드 예약접수(RequisitionCheck(MNL))를 확인하고 예약이 되어 있다면 작업실행(AddCQueue(MWIF,GN))을 사용하여 작업을 재시작한다. 예약 작업이 없을 경우는 서버복귀(NodeReturn(GN))를 실행하여 갱신한다. 마스터노드는 작업실행, 작업예약확인, 작업대기를 반복하면서 작업분배 및 결과수집을 수행한다.

```

MasterWorkCluster( MWIF,GN )
① INPUTE: MasterWorkInfo MWIF, GroundNode GN,
②      MesterNodeList MNL
③ WHILE ( Work( GN ) not empty ) DO
④   if ( Completion( GN ) )
④     MWIF += GN
⑤   else
⑥     AddICQueue( MWIF, GN )
⑦   end if
⑧   if ( !RequisitionCheck( MNL ) )
⑦     AddCQueue( MWIF, GN )
⑨     break
⑦   end if
⑧ END WHILE
⑨ NodeReturn( GN )
    
```

(그림 3) 마스터노드 작업 스케줄링 알고리즘

5. 성능 평가

제안된 스케줄링 기법을 검증하기 위해 가상화환경의 자원량을 측정하고 마스터노드 수에 따른 작업처리 시간을 측정하는 시뮬레이션을 수행한다.

5.1 가상화환경의 자원량 측정

시뮬레이션을 위한 서버의 성능은 Duo CPU E8400 @3.00GHz와 Memory 2G, HDD 300G이다. 가상화 틀은 VMware를 사용하고 가상화환경의 자원량(CPU,Memory)을 측정한다. CPU측정은 파일(MB/s), 이미지(Mpixels/s), 오디오(KB/s) 정보로 구분하여 압축 및 해체를 동시에 실행시켜 용량을 측정한다. 측정 결과는 <표 1>과 같다.

<표 1> 가상화환경에서 CPU 측정

처리정보 \ 처리시간	가상화전	2개 OS	3개 OS
파일압축(MB/s)	13	25	30
파일압축해제(MB/s)	195	384	472
파일암호화(MB/s)	82	161	176
파일복호화(MB/s)	82	161	196
이미지압축해제(Mp/s)	42	83	91
오디오압축(KB/s)	4006	7914	10006

측정 결과 비가상화 환경보다 가상화환경에서 처리량이 더 증가 하는 것을 알 수 있다.

Memory측정은 최대16MB, 최소4KB 용량을 사용하여 Read, Write, Copy를 실행하여 측정된 결과 값을 MB/s단위로 합

산한다. Memory측정도 구성된 가상환경에서 동시에 실행하며 측정 결과는 <표 2>와 같다.

<표 2> 가상화환경에서 Memory 측정

처리정보 \ 처리시간	가상화전	2개 OS	3개 OS
Read(16MB)	6734	8963	11346
Read(4KB)	44853	88418	111961
Write(16MB)	6668	6977	9816
Write(4KB)	43645	86042	103052
Copy(16MB)	5747	6036	8590
Copy(4KB)	44961	88577	91084

Memory측정 결과 가상화 적용시 환경에서 더 많은 자원을 사용 할 수 있는 것을 알 수 있다. 이처럼 가상화를 사용하지 않을 경우는 유휴자원이 발생하는 것이다. 가상화를 통해서 자원을 분배할 경우 가용된 자원을 더 많이 사용 할 수 있기 때문에 많은 노드를 사용하는 그리드 컴퓨팅에서는 가상화의 중요성을 생각 할 수 있다.

5.2 시뮬레이션

시뮬레이션은 마스터노드 수에 따라 <표 3>과 같은 파라미터 값을 이용하여 작업 처리시간을 계산하였다.

<표 3> 시뮬레이션 파라미터와 값

파라미터	값
노드수	100
작업수	100000
한개의 작업처리량	1000
작업수 증가량	100000

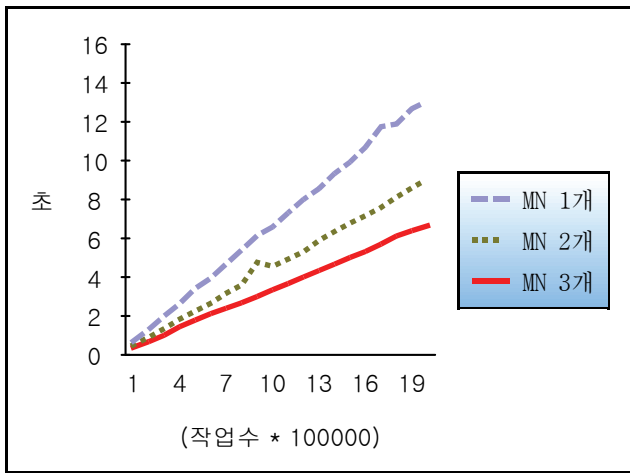
시뮬레이션은 가상환경에서 마스터노드를 3개로 구분하여 작업량을 100000개씩 증가시키면서 총20회 연속으로 측정 한다. 측정된 값은 모든 작업이 완료된 시간이며 작업시 발생하는 오류는 없다고 가정한다. 측정 결과는 <표 4>와 같다.

<표 4> 마스터노드 작업 스케줄링 측정

처리정보 \ 처리시간(s)	마스터 노드 1개	마스터 노드 2개	마스터 노드 3개
작업수 100000	0.71	0.51	0.39
200000	1.31	0.90	0.68
300000	2.00	1.35	1.01
...
...
...
1800000	11.89	8.12	6.12
1900000	12.68	8.60	6.39
2000000	13.07	9.06	6.64

최종 작업을 종료했을 경우 마스터노드가 1개일 경우는

13.07초이고 마스터노드가 3개일 경우는 6.64초 시간이 경과 되었다. 이처럼 마스터노드가 증가 할수록 작업처리 시간이 감소하게 된다.



(그림 4) 마스터노드 작업 스케줄링 측정 그래프

마스터노드 구성 시 (그림 4)에서 볼 수 있듯이 최종 작업 처리시간 결과 마스터노드가 1개일 경우와 2개일 경우 차이는 5초이며 마스터노드가 2개일 경우와 3개일 경우의 차이는 3초인 것을 알 수 있다. 이처럼 마스터노드가 증가 할수록 작업처리 시간의 폭은 감소하게 된다. 그렇기 때문에 가용된 자원에 적합한 마스터노드를 구성해야 가장 효율적인 작업처리가 가능하다.

6. 결론 및 향후 연구 과제

본 논문에서는 높은 성능을 가지고 있는 그리드 서버에서 발생하는 유휴자원 문제를 가상화를 통하여 해결하였다. 또한 구성된 가상환경에 그리드 작업 스케줄링을 대 행 할 수 있는 마스터노드를 구성함으로써 그리드 작업 스케줄링에 효율을 증가시켰다. 가상환경은 유휴자원을 사용하기 때문에 자원 사용률이 높다 하지만 가용된 자원에 부적합하게 가상화를 적용 할 경우 구성된 가상화 성능이 떨어지게 된다. 앞으로 연구에서는 가용된 자원에 효율적인 마스터노드 분배정책 및 서버간 통신을 통해 마스터노 드를 이용 할 수 있는 스케줄링을 연구한다.

참고문헌

[1] I.Foster, C.Kesselman and S. Tuecke, "The Anatomy of the Grid : Enabling Scalable Virtual Organizations", International Supercomputer Applications, Vol.15, No.3, 2001.

[2] I.Foster, C.Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", International J. Supercomputer Application, 11(2):115-128, 1997.

[3] Ian Foster, and Carl Kesselman, "The Grid : Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 1998

[4] V. 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. 191-202, 2000.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warden, "Xen and the Art of Virtualization," ACM in SOSP'03, pp.164-167, 2003

[6] VMWare, Inc. "Understanding Full Virtualization, Para virtualization, and Hardware Assist" <http://www.vmware.com>, 2010

[7] K.Keahey, I.Foster, T.Freeman, X.Zhang, D.Galron, "Virtual Workspaces in the grid", Proceedings of EuroPar 2005, Lisbon, Portugal, September, 2005.

[8] Keahey, K., K. Doering, and I.Foster. From Sandbox to Playground: Dynamic Virtual Environments in the Grid. in 5th International Workshop in Grid Computing. 2004.