

큐 캐싱을 이용한 그리드 자원관리 시스템 설계

함재균^o, 박형우, 이상산
한국과학기술정보연구원
{jaehahm^o, hwpark, sslee}@kisti.re.kr

Design of Queue Caching Grid Resource Management System

Jaegyoon Hahm^o, Hyoungwoo Park, Sangsan Lee
Korea Institute of Science and Technology Information

요 약

그리드 자원관리 시스템은 사용자로 하여금 지역적으로 분산된 고성능의 자원들을 사용하여 단일자원으로서는 계산할 수 없는 커다란 계산을 가능하게 하는 것을 목표로 한다. 이를 위해서 사용자에 대한 인증, 사용자 프로그램 분배 및 실행, 작업 모니터링 등 자원을 할당하기 위한 기능들을 제공해야 한다. 특히 자원의 사용에 있어서 자율적이고 안전한 방법으로 접근할 수 있도록 하는 것이 필요하다. 이러한 요구를 고려하여 본 논문에서는 기존의 자원 할당 시스템보다 사용자에게 편리하고 효율적인 그리드 자원관리 시스템을 제안한다. 고안된 시스템은 원격의 계산 자원에서 실행되는 큐에 대한 사본을 클라이언트에 관리함으로써 작업의 관리 및 재실행에 있어서 효율을 높일 수 있게 하였고, 사용된 큐를 캐싱함으로써 자원할당 과정에서 탐색과정을 줄일 수 있도록 하였다.

1. 서 론

그리드(Grid) 컴퓨팅은 방대한 자원의 공유를 통해 고성능의 선도적인 응용프로그램을 목표로 하고 있다는 점에서 기존의 분산 컴퓨팅과는 차별성이 있는 새로운 개념의 컴퓨팅 환경이다. 그리드는 동적인 자원과 사용자로 이루어지는 컴퓨팅 환경에서 유연하고 안전하며, 조화로운 자원의 공유를 가능케 하기 위한 것이라는 특징을 가지고 있다.[1]

그리드 자원관리 시스템은 이러한 그리드의 목표를 이루기 위해 지역적으로 분산되어 있는 이기종의 고성능 자원들을 사용할 수 있도록 하는 기능을 담당하고 있으며, 이에 필요한 인증과정, 사용자의 작업을 위한 프로세스 생성, 작업관리 등의 문제들을 다룬다. 특히 그리드라는 특수한 환경에서 발생할 수 있는 문제들, 즉 지역 및 운영정책 상의 자율성(site autonomy), 이기종 플랫폼 문제(heterogeneous substrate), 각 응용분야에 맞는 자원관리 정책(policy extensibility), 동시할당(co-allocation), 작업에 대한 온라인 제어(online control) 등의 문제들을 그리드 자원관리 시스템에서 해결할 수 있어야 한다.[2]

사용자들은 단일자원에서는 해결이 불가능한 커다란 계산 수행을 위한 목적으로 더 많은 자원을 얻기 위해 그리드를 사용하며, 그와 동시에 편리하고 효율적인 작업 실행 환경을 원한다. 따라서 그리드 자원관리 시스템은 분산된 자원을 동적으로 사용자의 요구에 맞게 할당하며, 자원의 탐색 및 선정, 동시할당, 사용자 인증, 지역 스케줄러 할당, 모니

터링 등 작업 수행에 필요한 일련의 기능을 제공해야 한다. 이에 더하여, 본 논문에서는 시스템의 성능을 향상할 수 있는 방법을 제안하고자 한다.

2. 관련 연구

2.1 GRAM

GRAM(Globus Resource Allocation Manager)[2]은 미국의 Globus 프로젝트에서 개발한 Globus Toolkit[3][4]의 자원관리 구조에서 가장 하위의 모듈이다. GRAM은 작업 제출, 관리, 종료 등의 API를 제공함으로써 사용자의 작업을 원격에서 실행할 수 있게 한다. GRAM이 동작하는 사이트에서는 gatekeeper를 통해 작업 요구를 받아 인증을 수행하고, 실행을 위해 작업 관리자(jobmanager)를 실행한다. 작업 관리자는 지역 자원을 관리하는 배치 작업 스케줄러에 맞는 작업 스크립트를 생성하여 큐에 제출함으로써 사용자의 작업이 시작되게 하고, 작업에 대한 모니터링을 수행한다.

GRAM과 더불어 Globus Toolkit의 자원관리 구조를 이루고 있는 또 하나의 모듈이 있는데, DUROC(The Dynamically-Updated Request Online Coallocator)이라는 모듈은 MPI 프로그램과 같은 병렬 프로그램을 여러 GRAM 사이트에서 동시에 수행되도록 하는 동시할당 기능을 알고 있다.

GRAM은 자원관리의 기본 기능을 충실히 구현하고 있으며, 현재 많은 그리드 프로젝트에서 자원 할당의 기본 모듈로

서 사용되고 있다. 그러나 GRAM이 담당하는 부분은 지역 자원에서의 자원관리 부분일 뿐이고, 사용자의 편의를 위해서는 GRAM을 이용한 자원 브로커를 구현해야 한다.

2.2 Condor-G

Condor-G[5]는 앞에서 설명한 GRAM을 활용하는 자원 브로커의 하나로써, 사용자로 하여금 그리드에 넓게 분산되어 있는 여러 자원들 위에서 작업을 수행하고 관리할 수 있도록 하기 위해서 만들어졌다. Condor-G는 미국의 위스콘신 대학에서 만든 스케줄러인 Condor를 Globus로 확장한 것이다. Condor-G는 Globus를 사용하여 여러 자원들 사이에서 인증, 원격 실행, 데이터 전송 등을 구현한다. Condor-G는 Globus의 globusrun과 같은 작업 수행 기능 외에도, 그리드 상에 입력 파라미터를 바꾸어 계산하는 수많은 작업을 한꺼번에 제출할 수 있도록 하고, 그것을 관리할 수 있는 기능을 제공한다. 그리고 작업의 수행 과정과 종료 등에 대한 로그를 사용자의 디렉토리에 저장함으로써 작업에 대한 알림 기능을 제공한다. 그리고 작업 수행 도중 프록시가 만료되었을 때 미리 작업을 중지시켜 Globus 크리덴셜(credential)을 유지하도록 한다.

2.3 GRMS

GRMS(Grid Resource Management System)[6]은 유럽의 GridLab 프로젝트의 WP9에서 연구되고 있는 그리드 자원관리 시스템으로서, 그리드에서의 분산 스케줄링과 자원관리를 위한 자원 브로커를 개발하는 것을 목표로 하고 있다. 현재 그리드 자원관리를 위한 요구분석과 시스템의 스펙 정의를 마치고, 자원관리 라이브러리들을 구현하고 있는 단계이다.

3. MRMS(Moredream Resource Management System) 설계
MRMS는 기본적으로 클라이언트-서버 구조를 가지고 있다. 클라이언트에서는 사용자의 작업요구를 가지고 필요한 자원의 탐색과 스케줄링 및 각 자원에서의 작업제출을 담당하고, 서버에서는 작업 요구를 받아 지역 자원 관리자인 작업 스케줄러에 제출하는 일을 담당한다. MRMS의 구성 요소는 서버에서 사용자의 작업이 돌고 있는 스케줄러의 큐를 캐싱하는 새도우 큐(shadow queue), 지역 자원 큐의 상태를 새도우 큐에 업데이트 하기 위한 이벤트 리포터(event reporter)와 작업 모니터(job monitor), 사용자 작업에 대한 로그 매니저(log manager), 그리고 자원을 사용자의 요구에 맞게 자동으로 찾고 할당하는 에이전트 등이 있다. 시스템의 구조는 그림 1에 나타나 있다.

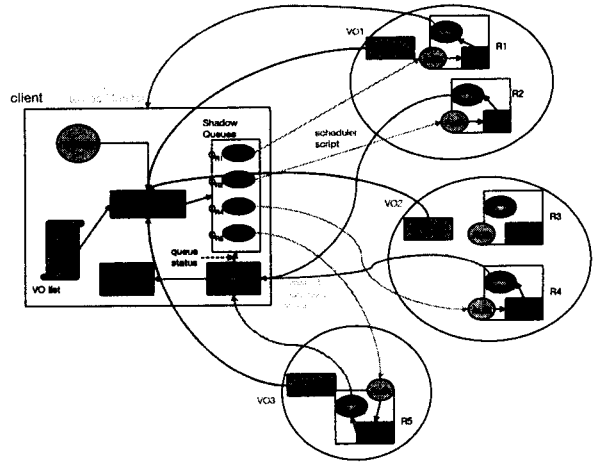


그림 1 MRMS 구조

3.1 새도우 큐(Shadow Queue)

새도우 큐는 사용자의 작업이 실행되고 있는 지역 자원의 배치 스케줄러의 큐를 복사해서 클라이언트에 유지하는 일종의 미러(mirror)이다. 이 새도우 큐는 작업의 수행 도중에, 각 자원에서 이벤트 리포터로부터 오는 큐의 상태변화를 작업 모니터를 통하여 업데이트 됨으로써 항상 최신의 상태를 유지하도록 한다. 작업이 끝난 후에는 클라이언트에 캐시로 남아 다음의 작업에 필요한 정보를 제공하게 된다. 이때 캐싱된 큐의 정보는 작업에 사용된 빈도순으로 정렬되어 할당에 실패할 확률을 낮추도록 하였으며, 나중에 사용될 때에는 정적인 정보만을 사용하게 된다.

3.2 자원의 탐색, 선택 및 작업 제출

에이전트가 사용자의 작업 실행 요구를 받게 되면, 해당 요구를 분석하여 필요한 자원을 찾게 되는데, 에이전트는 먼저 클라이언트에 큐의 캐시를 살펴보게 된다. 이 캐시에는 최근의 작업을 수행한 큐의 당시 상태가 보관되어 있는데, 이 캐시의 정적 정보를 통해 작업에 필요한 큐를 얻을 수 있으면 곧바로 각 자원의 지역 정보 제공자-예를 들어 Globus의 GRIS-를 통해 큐의 현재 상태를 읽어 업데이트 하고, 선택을 수행한다. 이는 자원 정보를 얻기 위해 매번 인덱스 서버를 검색하는 과정을 생략함으로써 시스템의 성능을 높일 수 있게 한다.

만약, 큐 캐시가 비어 있거나 사용자의 작업을 위해 큐의 노드 수가 충분하지 못하다면, 자원정보의 인덱스 서비스를 통해 탐색을 수행한다. 이때, 보통 사용자들이 사용할 수 있는 자원의 범위가 여러 VO(Virtual Organization)에 존재한다고 했을 때, 에이전트는 인덱스 서버를 검색하는데 있어서 하나의 VO 안에서만 그치는 것이 아니라 사용자가

접근 권한을 가진 VO들을 나열한 VO 리스트를 차례로 검색하면서 사용자의 요구를 만족하는 자원을 모으게 된다. 이렇게 함으로써 자원 선택에 있어서 확장성을 가질 수 있다.

모니터는 새도우 큐를 업데이트 하고, 해당 작업의 변화를 로그 매니저에 기록하게 된다. 이렇게 함으로써 로그 매니저는 사용자의 작업의 실행 궤적을 모두 가지고 있음으로써 작업에 대한 사후 관리가 용이하게 된다.[7]

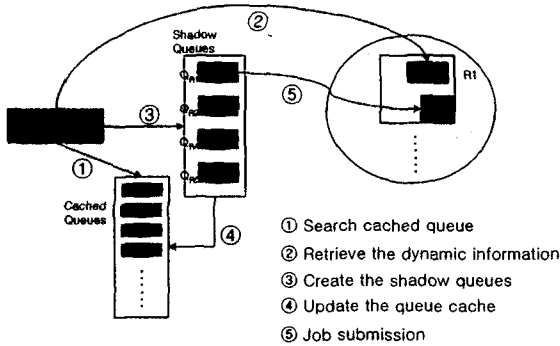


그림 2 자원 할당 과정

이 자원의 탐색 및 선택의 과정에서 에이전트는 선택된 자원들로부터 각 계산 자원을 관리하는 지역 큐의 현재 상태를 질의하여 클라이언트 측에 보관한다. 이 과정에서 기존의 새도우 큐를 업데이트하고, 새도우 큐 히스토리를 사용빈도순으로 재정리를 수행한다.

일단 에이전트가 자원들을 선택하고 나면 에이전트는 선택된 자원의 지역 큐의 종류에 맞는 부분작업들의 실행을 위한 큐 스크립트를 작성하여 지역 자원에 보낸다. 이 큐 스크립트를 받은 각각의 자원들은 사용자 인증의 과정을 거치고 나면 곧바로 자원의 큐에 스크립트를 제출하여 작업이 실행되도록 한다. (그림 2)

3.3 작업 복구 및 자원 재할당

병렬 프로그램이 실행 도중 자원에 문제가 발생해서 중단 되었을 때 기존에는 사용자가 다시 새로운 자원을 찾아서 실행해 주어야 했다. 이 과정에서 사용자는 작업의 처리를 처음부터 다시 똑같은 과정으로 반복해야 한다는 불편이 있었다. 그러나 MRMS에서는 새도우 큐에서 관리하고 있는 큐를 살펴서 여유가 있을 경우에 그 안에서 자원을 재할당할 수 있기 때문에 새롭게 GIS를 검색하는 시간을 절약할 수 있다. 새도우 큐에 충분한 노드가 없을 경우에만 에이전트가 GIS를 통해 새로운 자원을 찾아 재할당하게 된다.

3.4 작업 모니터링

실행중인 작업에 대해서, 그 프로그램이 할당된 지역 자원에서 현재 큐의 상태 변화가 일어날 때마다 이벤트 리포터가 클라이언트에 있는 작업 모니터로 알린다. 그러면, 작업

4. 결론

본 논문에서 제안된 자원관리 시스템은 그리드 환경에서 사용자의 프로그램 실행을 위해 자원 탐색 및 선택, 스케줄링, 동시 할당 및 사용자 인증, 지역 자원 할당 및 모니터링 등을 지원하게 된다. 이 과정에서 작업이 수행되는 지역 자원의 배치 스케줄러의 큐에 대한 사본을 만들어 작업의 실행 도중 모니터링이 가능하도록 하였고, 실행 도중 발생하는 문제로 인한 작업 중단에서 다시 복구될 때의 자원 탐색 과정을 줄일 수 있게 하였다. 특히, 큐 캐싱 개념을 도입함으로써, 자원 탐색 시간을 대폭 줄일 수 있는 방법을 고안하게 되었다.

5. 참고문헌

[1] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International J. Supercomputer Applications*, 15(3), 2001

[2] K.Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke. "A Resource Management Architecture for Metacomputing Systems". *Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, pg. 62-82, 1998

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

[4] I. Foster, C. Kesselman. "The Globus Project: A Status Report". *Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop*, pp. 4-18, 1998.

[5] J. Frey, T. Tannenbaum, I. Foster, M. Livny, S. Tuecke. "Condor-G: A Computation Management Agent for Multi-Institutional Grids", *Cluster Computing*, 5(3):237-246, 2002.

[6] <http://www.gridlab.org/WorkPackages/wp-9>

[7] 함재균, 박형우, "그리드 환경에서의 자원 사용 기록 생성 시스템 설계", 한국정보처리학회 추계학술발표대회, 제9권 제2호, 2002년