

사이트 자율성 보장을 위한 그리드 컴퓨팅 사용자 정보 서비스 아키텍처

김법균*, 안동언**, 정성중**

*전북대학교 공학연구원

**전북대학교 컴퓨터공학과

e-mail: kyun@chonbuk.ac.kr

Grid Computing User Information Service Architecture for Site Autonomy

Beob Kyun Kim, Dong Un An, Seung Jong Chung
Dept. of Computer Engineering, Chonbuk National University

요 약

그리드 환경에서 동작하는 소프트웨어의 가장 중요한 필수 요구사항들 중 중요한 것으로 언급되는 것들이 사이트 자율성과 확장성의 보장이다. 그리드 컴퓨팅 사용자들의 정보를 위한 서비스 또한 마찬가지이며 이를 고려하여 설계하여야 한다. 본 논문에서는 사이트 자율성을 보장하고 확장성이 있는 그리드 컴퓨팅 사용자 정보 서비스를 위한 아키텍처를 제안하고 구현한다. 구축된 사용자 정보 관리 상의 사이트 자율성을 보장하기 위해 2종류의 정보 서비스를 구축하고 각 사이트 관리자는 자신의 관리 영역 내에 위치한 정보 서비스를 관리하고 포털 관리자는 다수의 로컬 정보 서비스와의 연결을 통해 유연한 구조를 유지한다.

1. 서론

그리드 소프트웨어의 가장 중요한 필수 요구사항들 중 중요한 것으로 언급되는 것들이 사이트 자율성과 확장성의 보장이다. 그리드 어플리케이션을 제작할 때는 현재 및 미래의 수요를 예측할 수 없으므로 항상 사용자의 수요와 공급자의 공급상의 변화에 적용할 수 있는 확장성(Extensibility)을 갖추어야 한다. 또한 그리드 어플리케이션은 자원 소유자가 자신의 자원을 제어하는 능력을 제한해서는 안된다. 자원 소유자가 자신의 자원을 제어하는 능력을 보장받지 못한다면 대규모의 그리드 시스템의 구축은 불가능할 것이다. 즉, 각 사이트의 자율성(Site Autonomy)을 보장해 주어야 하며 이는 그리드 시스템의 복잡성을 줄여줌으로써 대규모 그리드 시스템의 구축을 가능하게 한다.

본 논문에서는 그리드 컴퓨팅 사용자의 정보를 구축하고 서비스하기 위한 구조를 제안하고자 하며 이 구조는 사이트 자율성을 보장하고 확장성이 있는

유연한 구조를 가지고 있다. 이를 위해 2종류의 정보 서비스를 구축하게 되는데 사이트 관리자는 자신의 관리 영역 내에 위치한 정보 서비스의 관리를 통해 정보 관리상의 사이트 자율성을 보장받게 되며 포털 관리자는 다수의 로컬 영역 내에 있는 정보 서비스들과의 관계를 유지함으로써 유연한 구조를 획득하게 된다.

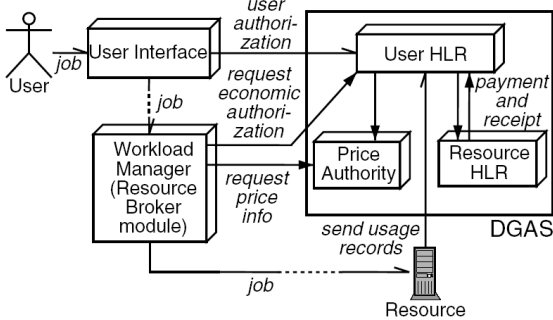
2. 관련 연구

전 세계적으로 수많은 그리드 프로젝트들이 진행되고 있으나, 각 프로젝트마다의 독자적인 어카운팅 시스템을 구축하고 다른 시스템과의 호환성 등에서 문제점을 보이고 있다. 대표적인 사례로 DGAS와 VUS 등을 설명한다.

2.1 DGAS

EDG(European DataGrid)의 WMS(Workload Management System)의 일부분인 DGAS(DataGrid Accounting System)은 자원의 교환을 원활하게 하

기 위한 목적으로 설계 및 구현되었다. 사용자는 자신의 HLR(Home Location Register)을 통해 자원 사용 권한을 획득한 후 WMS를 통해 작업을 수행한다. 중앙 집중화된 리소스 브로커를 통해 작업의 수행이 이루어짐으로써 자율성 보장에 한계가 있으며 구조의 유연성 및 호환성에서 문제가 있다.



(그림 1) DGAS

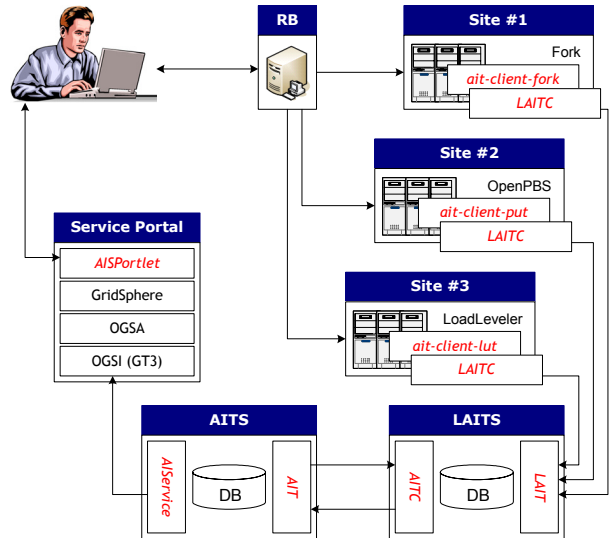
2.2 VUS

VUS(Virtual User's accounts System)은 폴란드 슈퍼컴퓨터 네트워크 센터에서 구축하였으며 Kerberos 시스템과 유사한 형태의 강력한 사용자 관리를 제공한다. 환경 내의 자원 이용은 VUSD에 의해 제어되고 모든 기록은 각 머신의 VUM을 통해 수집 및 구축된다. DGAS와 마찬가지로 정보 및 권한이 집중되며 그리드의 본래 특징인 분산 컴퓨팅과는 상반된 특징을 가지고 있다.

3. 그리드 컴퓨팅 사용자 정보 서비스 아키텍처

사이트의 자율성을 확보할 수 있는 그리드 컴퓨팅 사용자 정보 수집 및 서비스 아키텍처를 제안한다. (그림 2)과 같이 그리드 환경에서 사용자 정보의 수집 및 서비스를 위한 아키텍처를 제안한다. 그리드 사용자가 리소스 브로커를 통해 자원을 선택하여 작업을 실행시키면 각 자원에서는 작업에 대한 사용자 정보 추출이 이루어진다. 각 사이트마다 플랫폼에 맞는 모듈(ait-client-fork, ait-client-put, ait-client-lut 등)이 동작하여 사용자 정보의 추출이 이루어지며, 이 과정에서는 기존에 사용되어 이미 검증된 도구들이 생산하는 작업 수행 정보들이 수집된다. 수집된 사용자 정보들은 각 사이트와 관련된 LAITS(Local Accounting Information Tracking Service)에 로컬 플랫폼에서 사용하는 형식과 내용 그대로 저장된다. 이 정보들은 일정 시간마다 표준

형식으로 변환되어 AITS(Accounting Information Tracking Service)에 전송되며 OGSi 호환 서비스인 AIService를 통해 정보 서비스가 이루어진다. 웹 기반의 서비스 포털을 통해 서비스될 경우 포털에서 사용하는 포틀릿(Portlet)을 통해 서비스될 수도 있다.



(그림 2) 사용자 정보 서비스 아키텍처

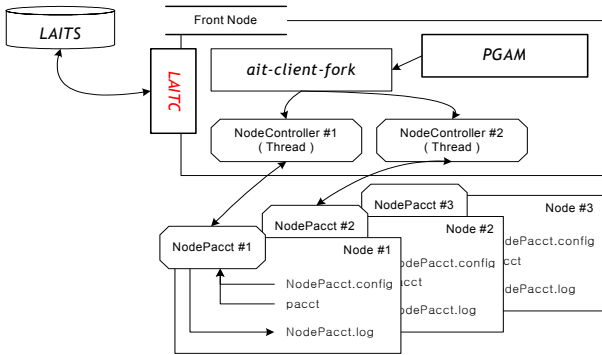
제안된 아키텍처는 선택적으로 로컬 사용자 계정에 대한 그리드 사용자의 책임소재를 밝힐 수 있도록 PGAM[-]과 같은 시스템의 도움을 받을 수 있도록 설계하였다. 물론, 사이트 자율적으로 CAS와 같은 다른 서비스와의 연동도 약간의 수정을 통해 쉽게 적용할 수 있다.

3.1 사용자 정보의 수집

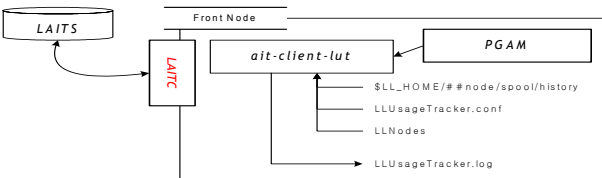
그리드 환경의 특성상, 그리드 환경에서의 어플리케이션은 최소한의 개입으로 각 사이트의 자율성을 보장하면서 최대의 효과를 얻도록 설계되어야 한다. 특히 사용자 정보는 자원 제공자 및 소비자 양측에서 매우 민감하게 반응할 수 있는 정보를 포함하므로 신뢰받을 수 있는 정보로 유지되어야 한다. 따라서 직접적인 개입을 통해 새로운 방법을 통한 정보의 수집 보다는 이미 검증된 도구를 통한 안정적인 정보의 수집이 선호되고 있다.

일반적으로 사용자들은 자신의 작업을 수행하기 위해 직접 프로세스를 발생시키는 경우와 작업 스케줄러를 이용하는 방법을 선택할 수 있다. 그 결과 발생하는 작업 수행정보는 작업을 실행시킨 형태에 따라 달라지므로 정보의 수집 시스템 또한 프로세스

단위의 정보와 작업 단위의 정보로 구분하여 수집할 필요가 있다. (그림 3)과 (그림 4)는 각각 프로세스 단위의 작업 수행 정보와 작업 스케줄러를 통해 생성된 작업 수행 정보의 추출을 위한 구조를 보여주고 있다.



(그림 3) 프로세스 단위의 정보 수집 아키텍처



(그림 4) 작업 단위의 정보 수집 아키텍처

프로세스 단위와 작업 단위의 정보 수집 모두 동일한 인터페이스인 LAITC를 통해 LAITS에 정보를 전달하며 시스템 의존적인 부분은 ait-client-put 이하의 부분이다. 동일한 LAITC 인터페이스를 통해 연결되므로 LAITS는 하위의 정보 수집 모듈과는 무관하게 독립적으로 동작할 수 있다.

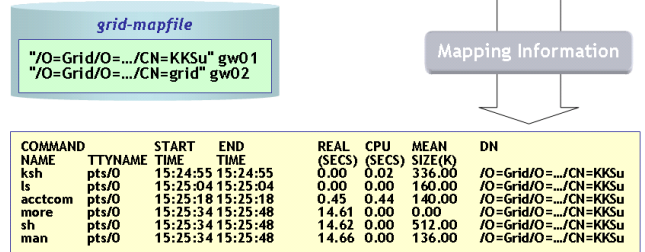
3.2 그리드 인식 정보로의 변환

각 로컬 사이트에서 수집된 정보는 그리드 환경과는 무관하게 기존의 자원에서 사용하던 방식의 정보이므로 그리드 환경에서 유효한 형태의 정보로 변환시켜주어야 하며 이 정보를 발생시킨 그리드 사용자를 규명하는 것으로 이를 해결할 수 있다.

Globus Toolkit의 경우 이 작업에 필요한 정보를 담고 있는 파일이 취약한 구조를 가지고 있으며 실제 손상된 상태로 유지되는 경우가 자주 발생하므로 별도의 접근 제어 시스템을 통해 유지하거나 'grid-mapfile'의 정보를 이용한 실시간 정보 매핑이 필요하다. (그림 5)는 'grid-mapfile'을 이용한 실시간 정보 매핑을 이용할 경우 정보의 변환을 도식화한 것이다. 이때 'grid-mapfile' 내의 그리드 사용자

와 로컬 계정간의 1:1 매핑이 유지되지 않는다면 이 과정은 사실상 불가능하므로 주의가 필요하다.

COMMAND NAME	USER	TTYNAME	START TIME	END TIME	REAL (SECS)	CPU (SECS)	MEAN SIZE(K)
ksh	gw01	pts/0	15:24:55	15:24:55	0.00	0.02	336.00
ls	gw01	pts/0	15:25:04	15:25:04	0.00	0.00	160.00
acctcom	gw01	pts/0	15:25:18	15:25:18	0.45	0.44	140.00
more	gw01	pts/0	15:25:34	15:25:48	14.61	0.00	0.00
sh	gw01	pts/0	15:25:34	15:25:48	14.62	0.00	512.00
man	gw01	pts/0	15:25:34	15:25:48	14.66	0.00	136.00



(그림 5) 'grid-mapfile'의 매핑 정보를 이용한 실시간 그리드 사용자 정보 변환

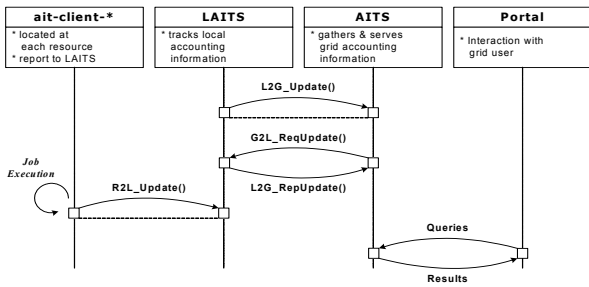
3.3 LAITS

LAITS(Local Accounting Information Tracking Service)는 각 로컬 사이트에서 수집된 정보를 동일 관리 도메인 내에서 서비스하는 역할을 한다. 그리드 사용자에게 관한 정보를 포함한다는 점을 제외하고는 원형에 가까운 형태로 저장되며 사이트 관리자가 전권을 행사할 수 있도록 함으로써 사이트 자체의 자율성을 보장하도록 하였다. LAITS는 일정 도메인 내에 존재할 수 있으며 이렇게 배치함으로써 관리 도메인 내에서의 어카운팅 정보를 바탕으로 한 시스템 및 사이트 사용 패턴 분석, 자원 관리 정책 수립 및 적용 등에 사용할 수 있다.

3.4 AITS

AITS에 보고되는 정보들은 이미 UR-WG에서 제안한 Usage Record 형식으로 변환된 것으로 다른 그리드 서비스에서도 언제든 활용 가능한 형태이다. 즉, AITS에서는 레코드 자체에 대한 변환 작업보다는 정보 자체의 효과적인 관리 및 서비스에 초점을 두고 있고 레코드들이 가지고 있는 정보를 바탕으로 다른 정보를 추출해 내는데 초점을 두고 있다.

(그림 6)은 지금까지 설계된 각 모듈들의 상호작용을 도식으로 표현한 것이다. 각 어카운팅 정보는 실행 머신 내의 어카운팅 레코드 번호와 시간 정보를 더하여 고유한 식별자(<urwg: RecordIdentity>)를 부여받게 되며 이를 통해 LAITS와 AITS의 정보들의 동기화가 이루어진다.



(그림 6) 모듈간의 정보 동기화

● R2L_Update()

수시로 정보를 수집하여 LAITS에게 보고한다. 설정에 따라 거의 실시간에 가까운 정보 수집을 할 수도 있다.

● L2G_Update()

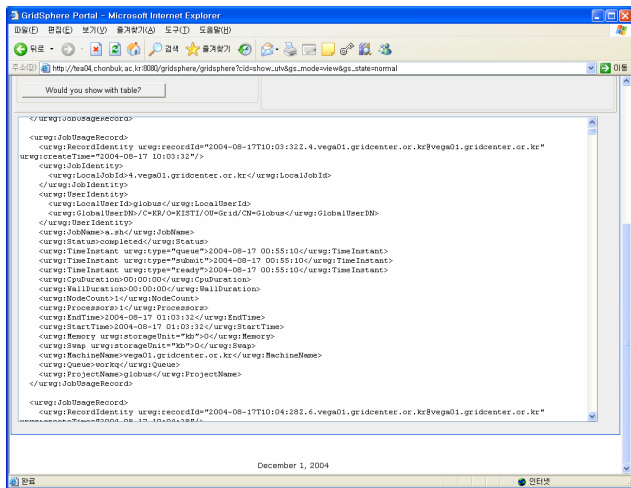
LAITS는 주기적으로 축적된 새로운 정보를 연결된 AITS들에게 보고한다. 각 정보는 <urwg:RecordIdentity> 필드로 구분된다.

● G2L_ReqUpdate()

포탈을 통한 사용자의 요청 또는 일정한 운영 정책에 따라 연결된 LAITS 들에게 정보 요구를 할 수 있다.

● L2G_RepUpdate()

요청받은 어카운팅 정보를 제공한다.



(그림 7) AIService를 위한 웹 기반 서비스 포탈

4. 결론

본 논문에서는 사이트 자율성을 보장하고 확장성이 용이한 유연한 구조를 가지는 그리드 컴퓨팅 사용자 정보 서비스를 위한 아키텍처를 제안하고 구현하였다. 최소한의 개입을 통해 신뢰받는 정보를 추출할

수 있도록 하였고 이 정보의 관리에 있어서 각 사이트의 자율성을 보장하기 위해 로컬 도메인 내에 LAITS를 유지하였으며 대외적인 정보 서비스를 위한 AITS를 다수의 도메인과의 연결이 용이하도록 함으로써 구조상의 유연성을 확보하였다.

참고문헌

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", The International Journal of High Performance Computing Applications, vol. 15, no. 30, pp. 200-222, 2001.

[2] Global Grid Forum, <http://www.ggf.org>

[3] The Globus Alliance, <http://www.globus.org>

[4] GFK, <http://www.gridforumkorea.org>

[5] M. Melani, "Thinking about Accounting," <http://sdm.lbl.gov/srm-wg>, September, 2004.

[6] A. Beardsmore, K. Harley, S. Hawkins, S. Laws, James Magowan, and A. Twigg, "GSAX: Grid Service Accounting Extensions," <http://www.doc.ic.ac.uk/~sjn5/GGF/rus-wg.html>, Resource Usage Service WG, Global Grid Forum, September, 2002.

[7] 김법균 외, "그리드 환경에서의 사이트 자율성 보장을 위한 접근 제어시스템에 관한 연구", 한국정보처리학회 논문지A, vol. 12-A, no. 2, April, 2005.

[8] Usage Record, <http://forge.gridforum.org/projects/ur-wg>

[9] The DataGrid Project, <http://eu-datagrid.web.cern.ch/eu-datagrid>

[10] Sebastian Ho, "GridX System Design Documentation", Bioinformatics Institute, 2002

[11] Thomas J. Hacker, Brian D. Athey, "Account Allocations on the Grid", Center for Parallel Computing University of Michigan, 2000

[12] S. Tuecke et al, "Open Grid Services Infrastructure (OGSI)", (draft), GGF, 2003

[13] B. K. Kim, D. U. An, S. J. Chung, and H. Jang, "Grid Accounting Information Gathering System with Access Control", APWeb 2005, 7th Asia-Pacific Web Conference, Lecture Notes in Computer Science, Springer, vol. 3399, pp. 845-850, 2005.