동적 가상조직 구성 환경에서 티켓기반의 상세 권한 위임 서비스

김병준[O] 홍성제 김종
포항공과대학교 컴퓨터공학과
{caleb80[O], sjhong, jkim}@postech.ac.kr

Ticket-based Fine-Grained Authorization Service in the Dynamic VO Environment

Byung Joon Kim[O], Sung Je Hong, Jong Kim
Department of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH)

요 약

그리드 컴퓨팅에서 가상조직은 분산되어 있는 여러 자원들과 사용자들로 구성된다. 가상조직을 구성하는 일은 그리드 컴퓨팅에서 중요한 일이다. 가상조직은 목적에 따라 동적으로 구성되며 목적을 완성하고 사라진다. 기존의 가상조직을 위한 권한 위임 구조들은 하나의 가상조직 환경에 중점을 두며 동적인 가상조직 환경을 고려하지 못하고 있다. 또한 자원 제공자, 가상조직 관리자, 그리고 사용자 모두를 위한 상세한 권한 위임 기법을 제공하지 않는다. 이 논문에서 우리는 동적인 가상조직에서 티켓을 이용한 상세한 권한 위임 서비스인 TAS 아키텍처를 제안한다. 기존의 아키텍처와 다르게 TAS 아키텍처는 티켓을 사용하여 여러 가상조직들을 위한 상세한 권한 위임 서비스를 제공한다.

1. 서 론

그리드[1] 컴퓨팅은 분산된 도메인에서 자원과 사용자를 모아서 가상조직(Virtual Organization)을 구성한다. 이러한 가상조직은 그 사용에 따른 목적, 분야 및 영역, 크기, 기간, 구조, 공동체 그리고 사회성을 가진다. 가상조직은 독자적인 내부 정책을 가지며 이를 통해 지역적으로 분산되어 있는 조직의 데이터, 프로그램, 계산 노드들을 통합된 형태의 계산 환경으로 가상조직의 사용자에게 제공된다.

그리드 환경에 있는 자원은 여러 개의 서로 다른 가상조직에게 자원을 제공 할 수 있다. 또한 가상조직은 제공 받은 자원을 가상조직에 속한 사용자들에게 제공하게 된다. 여기서 특정 사용자가 특정자원에 대해서 특정가상조직의 멤버로서 사용할 수 있도록 권한부여(Authorization) 하는 과정이 필요하게 된다. 자원 제공자는 가상조직에 따라 제공하는 자신의 자원을 관리 할 수 있기를 원하며 가상조직 관리자는 가상조직의 사용자에 따라서 제공하는 자원을 관리 할 수 있기를 원한다. 사용자는 자신에게 필요한 만큼의 자원을 제공 받기를 원한다. 이와 같이 자원의 양을 제한하여 필요한 만큼의 자원을 제공하는 것을 상세 권한부여(Fine-Grained Authorization)이라고 한다.

자원에 대한 권한부여에는 보안이 요구된다. 자원제공자가 특정 가상조직에게 제공한 자원을 다른 가상조직에서 사용될 수 없어야 하며 자원을 제공 받은 가상조직이 그 양을 위조 할 수도 없어야 한다. 또한 가상조직이 특정 사용자에게 제공한 자원에 대해서도 도용과 위조가 발생해서는 안 된다.

본 논문은 동적인 그리드 환경에서 자원제공자, 가상조직관리자 그리고 사용자를 위한 안전하고 상세한 권한부여를 실시하는 Ticket based Fine-Grained Authorization Service (TAS) 아키텍처를 제안한다. TAS 아키텍처는 티켓 기반의 상세한 권한부여 방식이다. TAS 아키텍처는 티켓을 이용하여 자원이 가상조직에게 가상조직이 사용자에게 그리고 마지막으로 사용자가 다시 자원에게 상세한 권한 위임을 할 수 있도록 한다. 본 아키텍처는 글로버스 툴킷으로 구성된 그리드 환경에서 구현했다.

2. 관련연구

CAS[2], VOMS[3], Akenti[4], PERMIS[5], SHARP[6] 그리고 WAS[7]은 그리드에서 권한 부여를 하기 위해 연구 되어온 기존 방법들이다. CAS는 한 가상조직의 구성원 모두가 신뢰하는 하나의 중앙 서버를 통해 권한 부여를 실시한다. 사용자는 자신이 얻을 수 있는 권한을 CAS 서버에게 요청하고 CAS 서버는 그 권한을 인증서의 형태로 사용자에게 제공한다. VOMS 역시 CAS와 같이 중앙의 하나의 서버를 통해 권한부여를 실시한다. VOMS는 사용자의 속성 인증서를 발급하며 이에는 사용자의 소속 그룹의 이름이 포함 되어 있다. 자원은 이러한 속성 인증서를 확인한 후 소속 그룹에 따른 권한부여를 실시한다. Akenti에서는 사용자의 권한과 가상조직의 권한이 분산되어 있다. Akenti는 작업의 요청을 받은 자원이 그 사용자의 권한을 찾아오는 방식을 취한다. PERMIS 역시 Akenti와 같이 자원이 사용자의 권한을 찾아오는 방식이다. PERMIS는 X.509 속성 인증서를 이용하며 역할 기반으로 권한관리를 하며 자원에 대한 권한 부여를 한다. SHARP는 1:1 신뢰관계의 P2P기반이다. 자원으로부터 발행된 티켓이 사용자에게 전달되어 권한 부여가 이뤄진다. 마지막으로 WAS는 Work flow기반의 상세한 권한 위임 기법이다. WAS는 CAS를 확장한 형태의 서비스로 사용자는 자신이 사용할 프로세스의 Work flow을 먼저 WAS 서버에게 인증 받고 그 인증서를 통해 실행 흐름에 따른 권한 위임을 실행한다.

기존의 권한 부여 모델은 하나의 가상조직의 관리자가 정적으로 정한 권한을 사용자가 사용 하는 것을 허용한다. 하지만 이러한

모델은 많은 자원, 많은 가상조직 그리고 많은 사용자가 있는 동적인 가상조직 환경 지원에 여러 문제점을 가지고 있다. 먼저 동적인 가상조직 환경 지원의 효율 성 문제가 있다. CAS, VOMS, WAS는 병목 현상문제와 확장성의 문제를 가진다. 이들은 하나의 가상 조직을 위해 하나의 관리 시스템을 요구한다. Akenti, PERMIS는 확장성이 좋으나 사용자가 먼저 자신이 사용할 수 있는 권한을 알 수 없는 문제가 발생한다. SHARP는 자원과 사용자 모두가 사용할 자원의 양을 제한하는 것을 지원하지만 이는 완전하게 분산되어 있는 환경과 1:1의 신뢰 관계를 기반으로 하여 가상조직에 따른 정책 모델을 적용 할 수가 없다. 두 번째로 이 모델은 자원 제공자, 가상조직관리자 그리고 사용자를 위한 상세한 권한 위임 기법을 가지고 있지 않다. 기존 모델에서 사용자는 동적으로 변하는 자원에 대해 동적인 권한 위임을 할 수 없다. 그리고 사용자는 자신이 필요한 만큼의 자원을 제한하여 상세한 권한 위임하는 것을 할 수 없다. 마지막으로 기존의 모델은 권한을 제공받은 자원의 가용성을 보장하지 않는다. 기존모델은 자원 제공자와 가상조직 관리자의 정적인 계약에 따라서 사용자에게 자원을 제공한다. 그러므로 제공된 자원의 가용성을 보장하지 못하며 다른 모델들을 이용하여 가용성 보장을 받아야 한다.

3. 제안하는 아키텍처

3.1 요구사항

그리드는 자원과 사용자 그리고 이들을 묶고 있는 여러 개의 가상조직들로 구성된다. 자원은 서로 다른 여러 개의 가상조직에게로 자원을 제공할 수 있고 가상조직은 서로 다른 사용자에게 가상조직안의 자원을 제공할 수 있다. 사용자는 서로 다른 여러 개의 가상조직에 소속될 수 있으며 소속되는 가상조직에 따라서 서로 다른 정책을 적용받게 된다.

그리드에 참여하는 자원관리자는 자원의 이용성을 최대화 시켜야한다. 자원은 여러 개의 가상조직들과 로컬의 사용자들에게 주어지며 유휴상태를 피해야 한다. 자원관리자는 자원의 작업 양을 동적으로 관리할 수 있어야 하며 가상조직들과 로컬의 사용자 에게 서로 다른 권한을 줄 수 있어야 한다. 확장성을 위해서 자원은 가상조직의 모든 사용자에 대한 정보를 알 필요는 없지만 어떤 가상조직의 어떤 사용자가 어떤 권한으로 자신에게 작업을 요청 하는가를 알 수 있어야 한다.

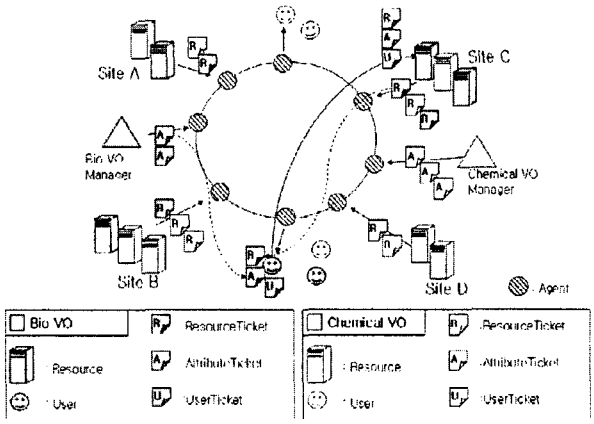
가상조직 관리자는 사용자들에게 정책에 따라 서로 다른 권한의 자원을 제공 할 수 있어야 한다. 가상조직은 자원 제공자들로부터 제공받은 자원의 양을 알 수 있어야 하며 사용자마다 얼마만큼의 자원을 사용했는지를 알 수 있어야 한다. 가상조직은 자원과 사용자 사이의 브로커 역할을 한다. 즉 사용자가 실제 얼마만큼의 자원을 사용 했는가를 알아서 그에 따른 대가를 얻어 자원을 제공한 자원 제공자에게 지불 할 수 있어야 한다. 또한 가상조직은 가상조직 밖의 사용자가 자원을 요청 할 수 없도록 하여 가상조직 사용자의 자원을 보호해야 한다.

사용자는 필요한 양의 자원을 지원 받기를 원한다. 이를 위해서 먼저 사용자는 자신이 사용할 수 있는 권한이 무엇인지 알 수 있어야 하며 자신의 모든 권한을 요청하는 것이 아니라 필요한 만큼의 권한을 요청 할 수 있어야 한다. 불필요한 요청을 하지 않기 위해서 사용자는 자신이 현재 사용 할 수 있는 자원의 양이 얼마 있지 알 수 있어야 한다.

상세한 권한부여를 실행하기 위해서는 위에 제시된 자원, 가상조직 그리고 사용자 각각의 필요를 충족 시켜줄 수 있어야 한다. 결국 상세한 권한 부여를 위해서는 자원, 가상조직 그리고 사용자라는 세 주체들이 자신이 제공할 수 있는 권한과 관계를 기술하고 인증할 수 있는 기법이 필요하다.

3.2 TAS 아키텍처

TAS 아키텍처는 자원 제공자, 가상조직 관리자 그리고 사용자가 발행하는 티켓들을 이용하여 동적인 가상조직 환경에서 상세한 권한 위임을 가능하도록 한다. 그리고 티켓들을 도메인과 가상조직으로부터 독립적인 Structured P2P System을 통하여 관리 공유하여 확장성을 가지도록 한다. 티켓에는 자원 제공자가 발행하는 자원티켓(Resource Ticket), 가상조직 관리자가 발행하는 속성티켓(Attribute Ticket), 사용자가 발행하는 사용자티켓(User Ticket)이 있다. 티켓에는 각각 발행자가 발행 받는 이에게 제한하고 있는 상세한 권한이 기입되어 있다.

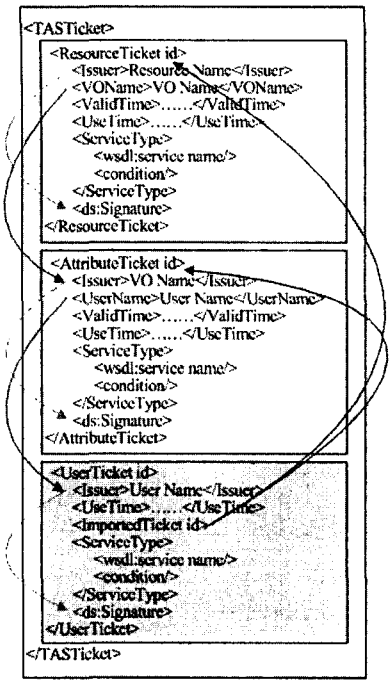


[그림 1] TAS 구조

[그림 1]은 TAS 아키텍처를 단순하게 표현하고 있다. 그림에는 Bio 가상조직(회색) 그리고 Chemical 가상조직(흰색)과 같이 2개의 가상조직이 있고 각각에 소속된 5명의 사용자가 존재한다. 가운데 큰 원은 분산 Structured P2P 기반의 분산 에이전트 시스템(Distributed Agent System)을 나타낸다. 자원제공자가 자신의 자원을 제공하기로 결정 했을 때 여러 가상조직에게 각각의 가상조직에 따른 자원티켓을 발행할 수 있다.

예를 들어, 사이트 B의 자원 제공자는 Chemical 가상조직을 위해 2장의 자원티켓을 발행하며 Bio 가상조직을 위해 1장의 자원티켓을 발행한다. 그림을 보면 Chemical 가상조직 관리자는 3명의 사용자를 위해 속성티켓을 발행했고 Bio 가상조직 관리자는 2명의 사용자를 위해 속성티켓을 발행 했다. 이들이 발행한 티켓들은 분산된 에이전트 시스템을 통해 관리 공유된다. Bio 가상조직의 사용자중 하나는 분산 에이전트 시스템을 이용하여 사용 가능한 자원티켓과 속성티켓을 찾고 자신이 사용할 자원의 양을 적은 사용자티켓을 만든다. 그리고 이 세 티켓을 하나의 TAS 티켓으로 만들어서 자원 제공자에게 자원을 요청한다.

이 모델에서 자원 제공자, 가상조직관리자 그리고 사용자는 티켓을 발행하여 쉽게 가상조직을 구성할 수 있다. 새로운 가상 조직에 참여하기 원하는 자원 제공자는 새로운 티켓을 발행하며, 기존의 참여를 중지하기 원하는 자원 제공자는 기존 티켓 발행을 중단하여 목적을 이루게 된다. 사용자의 추가와 삭제 역시 가상조직 관리자가 단순히 티켓을 발행하는 것에 의해 결정되어 동적인 가상조직 환경을 제공하게 된다.



[그림 2] TAS 티켓

[그림 2]는 TAS 티켓의 형태를 보여 주고 있다. TAS 티켓은 자원티켓, 속성티켓 그리고 사용자티켓으로 이루어 진 것을 알 수 있다. 자원 제공자가 발행하는 자원티켓에는 가상조직에 제공되는 자원에 대한 정보가 기록되어 있다. 자원티켓 속에는 제공되는 가상조직에 대한 정보와, 제공 될 Node의 수, Node의 가용시간, 티켓의 유효시간이 기록되어 있으며 자원제공자의 Digital Signature가 포함되어 있다. 자원 제공자는 내부의 정책에 따라서 서로 다른 가상조직들에게 서로 다른 자원티켓을 만들어서 발행한다. 속성티켓 속에는 제공되는 사용자에 대한정보와 제공 될 수 있는 Node의 수, 시간이 기록되어 있으며 가상조직관리자의 Digital Signature가 포함 되어 있다. 가상조직 관리자는 사용자에 따라서 서로 다른 권한을 기록한 티켓을 만들어 발행한다. 사용자는 자신이 사용할 프로세스에 따라서 제한된 양의 자원을 사용해야 한다. 사용자는 자신이 발행하는 사용자티켓 속에 자신이 실제 사용할 자원에 대한 정보들을 기록한다. 사용자티켓 속에는 사용할 Node의 수와 시간이 기록되어 있으며 자신의 Digital Signature가 포함 되어 있다. [그림 2]와 같이 TAS 티켓은 구조적으로 자원 티켓 안에 기입된 가상조직의 속성 티켓만 사용될 수 있으며 속성 티켓 안의 기입된 사용자의 사용자 티켓만이 사용될 수 있다. 사용자 티켓에는 TAS 티켓이 함께 첨부 되는 자원 티켓과 속성 티켓의 아이디가 포함되어 있다. TAS 티켓 안에 있는 각각의 티켓들은 전자 서명이 되어 있어 도용과 위조가 불가능 하다.

에이전트는 발행된 자원티켓과 속성티켓을 보관 하고 있으며 서로 다른 에이전트들과 그 내용을 공유한다. 자원 제공자와 가상조직 관리자는 각각 하나의 에이전트에게 자원티켓과 속성티켓을 발행하며 사용자는 또 다른 하나의 에이전트를 통하여 이들 티켓들을 찾아 올 수 있다. 에이전트는 사용자의 요구에 따라서 해당하는 자원의 자원티켓과 사용자가 소속되어 있는 가상조직의 속성티켓을 찾아준다. 에이전트는 유효기간이 지난 티켓을 자동으로 폐기하여 잘못된 티켓이 전달되는 일이 없도록 한다. 에이전트들은 사이트와 가상조직에 독립적으로 존재한다. 그러므로 특정 가상조직에 상관없이 에이전트로 자원티켓과 속성티켓을 발행 할 수 있으며 사용자 역시 가상조직에 상관없이 에이전트를 통해 발행된 티켓들을 찾아 올 수 있다. 우리는 Structured P2P 아키텍처인 Chord[8]을 이용하여 분산 에이전트를 구현하였다. 그 외에 Pastry, CAN등의 다른 아키텍처를 이용한 구현도 가능하다.

4. 결론 및 향후과제

TAS 아키텍처는 동적인 가상조직 환경에서 자원 제공자, 가상조직 관리자 그리고 사용자를 위한 상세한 권한 위임 기법을 제공한다. TAS는 Structure P2P 아키텍처를 사용하여 큰 규모의 가상조직 환경에서 효율적이고 확장성 있는 서비스를 제공하도록 했다. 그리고 서명이 이뤄진 티켓을 이용하여 안전하게 그리드의 세 주체인 자원 제공자, 가상조직 관리자 그리고 사용자가 상세한 권한 위임을 수행 할 수 있도록 하였다. 또한 티켓의 발행을 통해 자원 제공자와 가상조직 관리자가 효과적으로 가상조직을 구성하며 사용이 가능한 자원에 대해 권한을 부여 할 수 있도록 하였다.

우리는 글로버스 툴킷 3.0 환경에서 TAS 아키텍처를 동작하도록 구현하였다. 향후 우리는 큰 규모의 동적 가상조직 환경에서 TAS가 얼마만큼의 성능을 낼 수 있는지 연구 할 것이다. 또한 실제 적용이 가능한 어플리케이션에 대한 연구를 진행 할 계획이다.

참고 문헌

[1] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physioloby of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG (GGF), June 2002

[2] L. Pearlman, V. Welch, I. Foster, and C. Kesselman, "A Community Authorization Service for Group Collaboration," Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[3] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Å.Frohner, A. Gianoli, K. Lõrentey and F. Spataro, "VOMS, an Authorization System for Virtual Organizations," European Across Grids Conference, pp. 33-40, 2003.

[4] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari, "Certificate-based Access Control for Widely Distributed Resources," Proceedings of the 8th USENIX Security Symposium, pp. 215-227, August 1999.

[5] D. W. Chadwick and A. Otenko, "The PERMIS X.509 role based privilege management infrastructure," Future Generation Comp. Syst. 19(2), pp. 277-289, 2003.

[6] Y. Fu, J. S. Chase, B. N. Chun, S. Schwab and A. Vahdat, "SHARP: an architecture for secure resource peering," SOSP, pp. 133-148, 2003.

[7] S. H. Kim, J. Kim, S. J. Hong, and S. W. Kim, "Workflow-based Authorization Service in Grid," 4th International Workshop on Grid Computing, pp. 94-100, 2003.

[8] I. Stoica, R.Morris, D. L. Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications," IEEE/ACM Transactions on Networking, Vol. 11, No. 1, pp. 17-32, February 2003.