

Globus Toolkit V.3를 사용한 OGSA 기반 서비스 데이터 수집기 서비스 구현

강윤희*

요약

이 논문에서는 OGSA 기반 그리드 서비스의 특징을 기술하고 서비스 데이터 요소(Service Data Element, SDE) 정보 수집을 위한 그리드 서비스를 기술한다. 그리드 서비스의 구성을 위해서는, 주요한 시스템 컴포넌트와 이들 간의 상호작용을 표현하는 시스템의 고수준의 소프트웨어 아키텍처로의 그리드 서비스를 구성하기 위한 체계적인 접근이 고려되어야 한다. 이 논문의 목적은 SDE를 수집단위로 하는 서비스 데이터 수집 서비스의 설계 및 구현으로 이를 위해 GT3의 서비스 데이터 수집기 서비스를 자원 및 서비스의 종류에 따라 SDE를 영속적으로 유지할 수 있도록 저장 스킴을 구성하고 XML DBMS인 Xindice 사용하여 확장하였다. 서비스 데이터 수집 서비스는 인터넷과 같은 광대역 환경에서의 효율적인 수행을 위해 통지 메커니즘에 의해 비동기적으로 작동한다.

Implementation of a Service Data Aggregator Service based on OGSA By Using Globus Toolkit V.3

Yun-Hee Kang*

Abstract

This paper describes the main characteristics of Grid Services based on OGSA and a Grid service for aggregating service data element(SDE)s. In order to build a Grid Service, it needs to consider a systematic system building approach from the high-level software architecture that represents the main system components and their interactions. The purpose of this paper is to design and implement an extended service data aggregator service in which SDE is a basic unit for collecting resource information. A GT3 based service data aggregator service is extended to apply the multiple collections based storage scheme for maintaining persistently SDEs with a XML DBMS Xindice. To provide efficient aggregating service for service data elements, which is running under wide area environment like Internet, the aggregator service is asynchronously operated by notification mechanism.

Key words : OGSA, Service Data Aggregator Service, SDE

1. 서론

그리드 컴퓨팅(Grid Computing)은 방대하고 광대역의 자원 공유와 고성능 처리를 위한 분산 시스템의 설계, 구현 및 관리를 위한 기술이다[1-3]. 최근 OGSA(Open Grid Services Architecture) 기반의 그리드 개발 도구인 GT3(Globus Toolkit Version 3)는 XML 스키마, SOAP(Simple Object Access Protocol), WSDL(Web Services Description Language) 등으로 구성된 웹 서비스(Web Services)를 기반으로 한다[4-6,10,11,13]. 그러나 GT3의 그리드 정보 서비스(Grid Information Service) 시스템은 서비스 정보 유지 측면

에서 단일 저장 스킴에 따라 수집된 자원 및 서비스 정보를 저장하는 제약점을 갖는다.

그리드 시스템의 사용자는 자신이 속한 VO(Virtual Organization) 내에서 서비스 및 계산자원들의 정보를 쉽게 검색하고 자신이 필요한 서비스 및 계산 자원을 동적으로 사용하기 원한다. 이를 위해서는 VO 내의 가용한 서비스 및 계산자원들에 관한 정보를 수집하고 정리해서 사용자 또는 다른 응용에 제공하여야 한다[5],[6].

이 논문에서는 OGSA 기반의 서비스 데이터 수집기(Service data aggregator)를 확장하였으며, GT3의 정보 서비스 시스템의 저장구조의 제약점을 해결하기 위해 다중 저장 구조를 갖도록 하였으며, 서비스 및 자원

* 제일저자(First Author) : 강윤희

접수일 : 2005년 2월 1일, 완료일 : 2005년 2월 6일

* 천안대학교 정보통신학부 조교수

yhkang@cheonan.ac.kr

종류에 따라 수집되어진 서비스데이터 정보를 Xindice XML 데이터베이스에 저장한다.

이 논문의 제 2절에서는 웹 서비스 및 OGSA 기반 그리드 서비스의 특징 기술하고 기존의 분산객체 컴포넌트와의 비교를 통해 장단점을 분석한다. 제 3절과 4절에서는 그리드 서비스인 서비스 데이터 수집기의 설계 기법 및 구현을 중심으로 기술한다. 구현된 서비스 데이터 수집기는 인터넷과 같은 광대역 환경에서의 신뢰성 있는 수행을 위해 통지 메커니즘에 의해 비동기적으로 작동한다. 제 5절에서는 결론 및 향후연구를 기술한다.

2. 관련연구

본 절에서는 웹 서비스의 특징 및 제약점을 설명하고 그리드 운용 아키텍처인 OGSA 기반 환경의 특징을 기술한다.

2.1 웹 서비스 개요

웹 서비스는 기존의 CORBA, RMI, DCOM 등의 다른 분산 컴퓨팅 기술을 통합할 수 있는 새로운 클라이언트/서버 응용 개발 환경을 제공한다[5],[6]. 그리드 컴퓨팅을 위한 미들웨어인 GT3의 개발 환경은 웹 서비스를 기반으로 하고 있다[4]. 웹 서비스는 CORBA와 DCOM과 같은 분산 컴퓨팅 기술을 웹 기술에 차용하여 필요한 서비스를 웹에서 동적으로 찾은 후 서비스를 요청하고 그 결과를 사용하는 URI 기반의 접근 가능 소프트웨어 에이전트(software agent)를 총칭한다[8-11].

웹 서비스는 SOAP을 사용한 메시지 기반 표준 및 WSDL을 사용한 서비스 정의 기술 명세를 적용하므로 높은 상호운용성(interoperability)을 제공한다[10],[11]. SOAP은 분산 환경에서 정보 교환에 사용하는 경량(lightweight)한 프로토콜로서 텍스트 기반의 XML을 프로토콜로 사용한다[12]. WSDL은 웹 서비스가 제공하는 서비스에 대한 명세를 기술하기 위한 XML 문서로서 웹 서비스가 제공하는 기능, 접근을 위한 주소 및 호출 방법을 기술하기 위해 사용한다. WSDL은 메소드의 기술(description), 인자 유형(argument type) 및 리턴 값(return value)을 기술하는 면에서 CORBA의 IDL과 비교할 수 있다[8-10].

2.2 OGSA 구성

OGSA는 소프트웨어 아키텍처(software architecture) 모델로서 OGSA는 동적으로 구성되는 VO 내에서의 협력적인 자원 공유와 지속적인 정보 서비스를 제공한다[13,14]. OGSi는 그리드 컴퓨팅을 위한 프로그램 모델로서 임시 서비스 인스턴스(service instance) 생성, 상태 공개(status exposure), 생명주기(life cycle) 관리,

통지(notification), 등록(registration) 및 팩토리 패턴(factory pattern) 등의 특징을 갖는다. OGSA 기반 미들웨어인 GT3는 웹 서비스가 갖는 비상태유지 연결(stateless connection)의 문제점을 개선한 그리드 운영 환경을 제공한다[12]. 그리드 서비스 설계를 위해 확장된 WSDL 형식인 GWSDL이 연산(operation)과 서비스 데이터 요소(Service Data Element, SDE)를 기술하기 위해 사용된다.

SDE는 그리드 서비스의 상태 정보를 유지하고 필요시 다른 그리드 서비스에 상태정보를 제공하기 위해 사용한다. 또한 SDE는 특징에 따라 분류되고 색인될 수 있다. 즉, SDE는 그리드 서비스 정보를 색인하기 위해 사용하며 서비스의 특성 및 제공 정보의 종류와 특성에 따라 질의(query) 또는 통지(notification) 기법으로 사용된다.

3. 서비스 데이터 수집기 서비스 설계

서비스 데이터 수집 그리드 서비스 설계는 WSDL 설계 후 해당 서비스를 구현하는 WSDL로부터 코드를 구현하는 하향식(top-down) 접근 방법을 통해 이루어진다. 이를 위해 WSDL의 확장 버전인 GWSDL을 사용한다. WSDL 1.1은 portType의 상속 및 내재된 확장을 제공하지 못한 반면 GWSDL의 표준 명세인 WSDL 1.2는 OGSi 명세를 반영하여 설계됨으로써 상속 및 내재된 확장을 제공한다. 또한 연산 제공자(operation provider)를 기반으로 한 위임 모델(delegation model)은 WSDL 연산들의 다양한 구현을 쉽게 플러그인 할 수 있는 설계를 가능하게 한다.

GT3에서 서비스 정의 구성요소는 WSDL 파일 형식의 확장인 GWSDL에 기술한다. 그림1은 설계된 서비스 데이터 수집기의 스키마 파일인 service_data_aggregator_port_type.gwsdl 내에 포함되는 portType 태그를 보인 것이다. portType 태그는 서비스 접근점(end-point)에서 지원하는 연산(operation)의 집합의 추상화된 정의로서 서비스에 의해 공개된 메소드의 집합이다. portType 태그의 name 속성은 객체인 ServiceDataAggregatorPortType을 표현한다. 해당 객체의 연산은 서비스 구현 시에 Java 클래스 내부의 메소드으로써 구현되게 된다. 연산 addSubscription의 입력은 AddSubscriptionInputMessage()를 사용하고, 출력은 AddSubscriptionOutputMessage()를 통해 얻어진다. 서비스 데이터 수집기의 각 메시지는 해당되는 데이터 타입을 가질 수 있다. 메시지의 name 속성 값 AddSubscriptionInputMessage는 연산 AddSubscription의 입력 메시지 tns:AddSubscriptionInputMessage를 정의한 것으로 각 message 태그는 서비스 호출 파라미터를 나타낸다.

```

<gwsdl:portType
name="ServiceDataAggregatorPortType"
extends="ogsi:NotificationSink">
<operation name="addSubscription">
<input
message="tns:AddSubscriptionInputMessage"/>
<output
message="tns:AddSubscriptionOutputMessage"/>
</operation>
<operation name="removeSubscription">
<input
message="tns:RemoveSubscriptionInputMessage"/>
</operation>
</gwsdl:portType>
    
```

그림 1. 서비스 데이터 수집기 서비스 스키마 파일

4. 서비스 데이터 수집기 서비스 구현

4.1 그리드 서비스 구현 환경 및 도구

본 서비스 데이터 수집기 서비스는 GT3 기본 서비스인 Service Data Aggregator를 확장하여 통합 그리드 정보 서비스 시스템을 구축하였다. 또한 수집된 서비스 데이터는 XML 전용 데이터베이스 관리 시스템인 Xindice를 이용하여 저장한 후 정보 접근이 가능하도록 데이터베이스 설계하고 구축하였다. 본 시스템의 구현환경으로는 Java 기반의 호스팅 환경을 제공하는 그리드 미들웨어인 GT3.0.2를 사용하여 Linux 및 Windows XP에서 구동하도록 하였다[12].

그리드 서비스 응용 개발을 위해서는 GT3.0.2의 그리드 서비스 개발 지원 도구인 GSDL2Java, GenerateBinding 및 GWSDL2WSDL를 사용한다. GWSDL2WSDL 도구는 서비스의 호출을 위해 사용되는 전송 프로토콜과 요청 방식을 기술하는 binding 정보, 실제 구현과 관련된 연결 포트(port)를 요소로 갖는 서비스 및 외부에서의 인터페이스를 위한 portType에 관련된 파일을 생성한다. WSDL 1.2의 확장 기능인 extends는 GWSDL2WSDL 도구에 의해 WSDL 1.1 형식으로 변환된다. 다음은 GWSDL2WSDL 도구를 사용하여 3.2 절의 GWSDL 파일을 자동 생성한 후의 생성된 WSDL 파일이다.

- service_data_aggregator_bindings.wsdl
- service_data_aggregator_service.wsdl
- service_data_aggregator_port_type.wsdl

gwsdl2wsdl 도구를 사용하여 생성된 wsdl 파일인 service_data_aggregator_port_type.wsdl은 ogsi의 NotificationSink 인터페이스 내의 연산인 deliverNotification이 ServiceDataAggregatorPortType 내에 포함된다.

4.2 ServiceDataAggregator 서비스 구현

GT3.0.2에서는 Index 서비스에서 XindiceServiceData

Set 클래스를 초기화하여 하나의 컨테이너(container) 인스턴스를 만들고 이것을 이용하여 데이터를 저장하는 방식을 사용하였으나 서비스 데이터의 종류별로 서로 다른 컬렉션을 생성하고 구분하여 저장함으로써 데이터 베이스의 활용 효율을 높이고자 한다.

구현한 다중 컬렉션 구조는 VO에 기반으로 구성된다. GT3.0.2에서 하나의 컬렉션만을 생성할 수 있는 구조를 변경하기 위하여 본 연구에서는 Xindice 데이터베이스에 정보를 저장하기 위한 메서드 호출이 발생하는 시점에 컬렉션을 위한 XindiceServiceDataSet 컨테이너를 VO 이름에 기반을 두어 생성한다. VO 이름과 더불어 각 서비스 이름도 컨테이너 생성에 사용된다. 이미 존재하고 있는 VO 이름이나 서비스 이름인 경우에 해당 컨테이너가 생성되지 않고 그대로 사용된다. 그림2는 다중 컬렉션의 계층 구조를 보여준다.

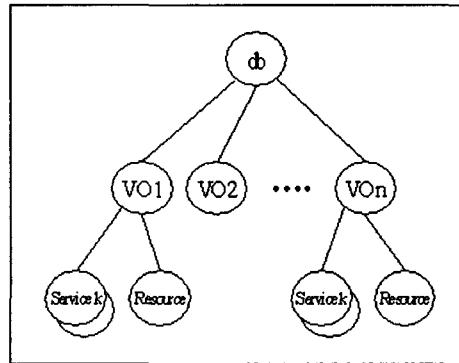


그림 2. 다중 컬렉션을 갖는 xindice DB 스키마

전체적인 서비스 데이터 수집 서비스는 IndexSvc, ServiceDataAggregator, XindiceSvc와 RIPS의 4가지 모듈로 구성된다. RIPS는 자원에 대한 상태 정보의 모니터링을 수행하며 observable로서의 기능을 수행한다. IndexSvc는 서비스 데이터 전달을 위한 접근점으로서 RIPS의 자원 상태 변경을 통보 받는 observer의 기능을 수행하고 XindiceSvc는 데이터의 영속 저장을 수행한다. IndexSvc는 사용자 및 다른 서비스의 전반부(front-end)로서 작동한다.

RIPS 모니터는 주기적으로 시스템의 자원 정보를 모니터링한다. 자원 정보의 예인 시스템 부하는 OS 스케줄러 내의 준비 큐의 길이로서 정의될 수 있다. RIPS 서비스는 1분, 5분 및 15 분 동안의 시스템 로드를 얻기 후 Cluster SDE에 정보를 유지하는 portType을 갖는다. 상태 수집기는 Cluster SDE를 구독하여 비동기적으로 통지를 수신한다.

서비스 데이터 수집 정보 시스템의 제어흐름은 다음과 같다.

- IndexSvc는 서비스 구동 후 SDE의 영속 저장을 위해 초기 xindice DB를 설정한 후 collection을 초기화
- IndexSvc는 사용자 또는 다른 서비스의 RIPS 서비

스의 SDE 질의 또는 구독 요청에 의해 addSubscription() 메소드가 콜백 형태로 호출

- IndexSvc는 구독 처리의 위임(delegate)을 위해 ServiceDataAggregator를 구독
- RIPS 서비스의 Cluster SDE의 변경이 발생하면 ServiceDataAggregator는 deliverNotification() 메소드가 콜백으로 호출되므로 통지
- deliverNotification에서는 수신된 통지 메시지를 파싱한 후 자료의 형태에 따라 그리드 서비스 정보와 자원 정보를 분류한 후 컬렉션을 선택한 후 저장

본 서비스는 영구적인 그리드 서비스로서 배치되어 글로버스 컨테이너(globus container) 의 시동시점에서 활성화될 수 있다. 또한 통지를 수행하는 OGSi NotificationSource 제공자를 사용하도록 명시한다.

서비스 데이터 수집 서비스의 GUI는 수집하고자 하는 SDE에 대한 구독을 위해 사용한다. 이를 위해 버튼과 사용자 입력을 처리하기 위한 리스너(Listener)를 사용한다. 사용자가 구독을 위한 SDE와 소스를 입력한 후 구독을 위한 리스너를 생성한다. 이를 위해 SDESubscribe가 사용된다. 구독이 성공한 후에는 반환 값인 sink의 URL을 검사하여 구독이 성공하였음을 알 수 있다.

그림 3 은 가시화되어진 시스템의 부하 정보를 포함하는 SDE를 모니터링 하는 ServiceDataAggregator 서비스의 처리 내용을 보인 것이다. 구독의 sink에서는 NotificationSinkCallback 인터페이스와 ListenerBasePortType 인터페이스를 구현하는 형태를 가지며, NotificationSinkManager의 객체를 사용하여 source로부터의 자료 통지를 수신하기 위한 자신의 리스너 쓰레드를 구동시킨다. deliverNotification 메소드는 구독에 대한 메시지를 전달 받은 후에 구동되는 콜백(callback) 메소드로서 내부에서는 전달 받은 메시지에 대한 해석을 수행한다.

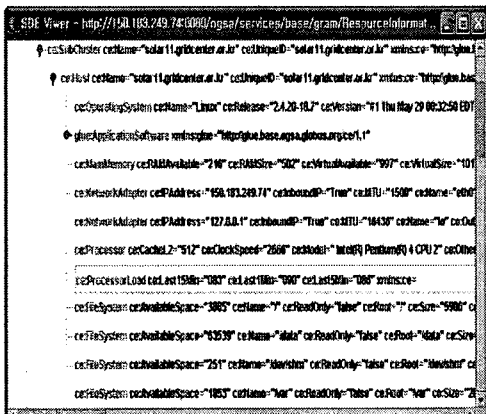


그림 3. 통지 후에 수집된 시스템 부하를 표현하는 서비스 데이터

5. 결론 및 향후 연구

본 논문에서는 그리드 환경을 제공하기 위해 GT3 기반의 서비스 데이터 수집 서비스를 기술하였다. 이를 위해 소프트웨어 아키텍처 측면에서 웹 서비스의 확장 기능을 제공하는 그리드 서비스의 기능을 기술하였다. 또한 확장된 그리드 서비스 설계 과정에서는 그리드 서비스 인터페이스 명세인 GWSDL 명세의 작성을 통해 주요한 시스템 컴포넌트와 이들 간의 상호작용을 기술하는 하향식 개발 방법을 사용하였다. 구현된 서비스 데이터 수집 서비스는 인터넷과 같은 광대역 환경에서의 신뢰성 있는 수행을 위해 통지 매커니즘에 의해 비동기적으로 작동하도록 구현하였다.

향후 연구로는 본 시스템을 확장하여 지속적인 서비스 수행을 위해 결함 포용 기능을 갖는 OGSA 기반 그리드 서비스를 설계할 예정이다.

참고 문헌

- [1] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," Morgan Kaufmann, San Fransisco, CA, 1999.
- [2] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Intl. J. Supercomputer Applications, Vol. 11, No. 2, pp 115-128, 1997.
- [3] S. Fitzgerald, I. Foster, C. Kesselman, G. Von Laszewski, W. Smith, and S. Tuecke. "A directory service for configuring high-performance distributedcomputings," In Proc. 6th IEEE Symp. On High Performance Distributed Computing, pp 365-375, 1997.
- [4] M.P. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing," CACM, Vol. 46, No. 10, Oct 2003.
- [5] I. Foster, C. Kesselman, J. Nick, S. Tuecke., "Grid Services for Distributed System Integration, " IEEE Computer, Vol. 35, No. 6, 2002.
- [6] Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju, Web Services-Concepts, Architectures and Applications, Springer Verlag, 2004.
- [7] P. Baglietto, M. Maresca, A. Parodi and N. Zingirian, "Deployment of Service Oriented Architecture for a Business Community," In Proc. of the 6th International Enterprise Distributed Object Computing (EDOC'02), 2002.
- [8] W3C, Web Services Architecture, <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>
- [9] F. Curbera et. al., "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI," IEEE Internet Computing, Vol. 6, No. 2, pp. 86-93, March/

April 2002.

- [10] W3C, Web Services Description Language(WSDL) 1.1, 2001, <http://www.w3c.org/TR/wsdl>
- [11] W3C, Simple Object Access Protocol (SOAP) 1.1. 2000, <http://www.w3c.org/TR/SOAP>.
- [12] T. Sandholm, S. Tuecke, J. Gawor, R. Seed, T. Maguire, J. Rofrano, S. Sylvester, and M. Williams, "Java OGSI hosting environment design a portable grid service container framework," Tech. Rep., Globus, 2002.
- [13] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.

강 윤 희



1989년 동국대학교 컴퓨터공학과
(공학사)

1991년 동국대학교 컴퓨터공학과
(공학석사)

2002년 고려대학교 컴퓨터과학과
(이학박사)

1991년 - 1994년 한국전자통신연구원(연구원)

1994년 - 1997년 한국문화예술진흥원,
전산개발부 (선임연구원)

1997년 - 2000년 (주)오름정보 개발부(과장)

2000년 - 현재 천안대학교 정보통신학부 조교수

관심분야 : 그리드 컴퓨팅, 분산처리,
에이전트 시스템, 디지털라이브러리