

# 프로세스 결함 검출을 위한 OGSA 기반 그리드 서비스의 설계 및 구현

## A Grid Service based on OGSA for Process Fault Detection

강윤희

천안대학교 정보통신학부

Kang Yun-Hee

Dept. of Computer & Communication,  
Cheonan University

### 요약

네트워크 및 소프트웨어 하부구조의 진보에 의해 이형의 컴퓨팅 자원 집합으로 구성된 환경에서 그리드 컴퓨팅 기술은 확산되고 있다. 그리드 컴퓨팅은 광대역으로 연결되어진 분산 컴퓨터들의 조정에 의한 사용을 요구한다. 그리드 컴퓨팅을 위한 다양한 시스템과 그리드 응용 서비스의 증가에 따라 그리드 운영환경은 네트워크 분리 및 노드 결함 등에 의해 높은 결함율을 가질 수 있다. 결함 검출은 그리드 응용 시스템의 강건성을 위해 시스템 설계 및 구현시의 필수적인 요소이다. 본 논문에서는 낮은 네트워크 트래픽 환경에서 높은 신뢰성 정공을 위한 OGSA(Open Grid Service Architecture) 기반의 자원 결함 검출 서비스를 제안한다.

### Abstract

With the advance of network and software infrastructure, Grid-computing technology on a cluster of heterogeneous computing resources becomes pervasive. Grid computing is required a coordinated use of an assembly of distributed computers, which are linked by WAN. As the number of grid system components increases, the probability of failure in the grid computing is higher than that in a traditional parallel computing. To provide the robustness of grid applications, fault detection is critical and is essential elements in design and implementation. In this paper, a OGSA based process fault-detection services presented to provide high reliability under low network traffic environment.

## I. 서론

그리드 컴퓨팅(Grid Computing)은 방대하고 광대역의 자원 공유와 고성능 처리를 위한 분산 시스템의 설계, 구현 및 관리를 위한 기술이다[1,2]. 최근 그리드 컴퓨팅을 위한 다양한 시스템이 증가하고 있으며 그리드 응용 서비스의 구성요소의 증가에 따라 결함(fault)의 가능성은 높아지고 있다. 결함 검출(fault detection)은 그리드 응용 시스템의 강건성

(robustness)을 위해 시스템 설계 및 구현시의 필수적인 요소이다[3,4,5]. 그러나 현재 웹 서비스 환경의 확장인 OGSA(Open Grid Service Architecture) 기반의 그리드 운영 환경에서는 결함을 처리하기 위한 구성요소가 미들웨어에서 제공되지 않고 있다[7].

본 논문에서는 OGSA기반의 그리드 미들웨어인 GT3(Globus Toolkit Version 3) 상에서 결함 포용을 갖는 높은 가용성의 보장을 위한 결함 검출 그리

드 서비스 설계 및 구현을 기술한다.

본 논문의 제 2절에서는 관련연구를 기술하고 제 3절에서는 결합 검출의 설계를 기술하고 제 4절에서는 결합 검출기의 구현을 마지막으로 제 5절에서는 결론을 기술한다.

## II. 관련 연구

본 절에서는 OGSA 기반 그리드 참조 구현인 GT3 미들웨어의 개요 및 결합 검출 기법의 특징을 기술한다.

### 1. OGSA 그리드 아키텍처

OGSA는 소프트웨어 아키텍처(software architecture) 모델로서 OGSI 표준의 참조 구현(reference implementation)으로 서비스 지향 아키텍처(Service Oriented Architecture, SOA)의 일종인 웹 서비스를 기반으로 한다[6].

OGSA를 위한 프로그래밍 하부구조의 표준인 OGSI는 위치 투명성(location transparency), 프로토콜 독립성(protocol independence) 및 서비스 지향 아키텍처의 특징을 제공한다. OGSI에서는 WSDL을 연산(operation)과 서비스 데이터 요소(Service Data Element, SDE)를 기술하기 위해 사용함으로써 자기 기술적(self-describing)이고 응용간의 통신에 적합한 그리드 서비스를 개발을 용이하게 한다. OGSI는 임시 서비스 인스턴스(service instance) 생성, 상태 공개(status exposure), 생명주기(life cycle) 관리, 통지(notification), 등록(registration) 및 팩토리 패턴(factory pattern)을 제공한다[7].

GT3는 웹 서비스가 갖는 stateless 연결 및 persistent 서비스의 문제점을 개선한 그리드 운영환경으로 OGSI(Open Grid Services Infrastructure)는 그리드 컴퓨팅을 위한 프로그램 모델이다. 그리드 서비스는 자신을 기술하는 서비스 데이터 집합

(Service Data Set)를 갖는다. 서비스 데이터 집합은 그리드 서비스와 연관되는 정보의 구조화되어진 집합으로 0 개 또는 다수의 SDE로 구성된다. SDE는 질의 되어질 수 있으며 필요시 동적으로 생성되어 외부 그리드 서비스에서 접근할 수 있다. SDE는 그리드 서비스의 상태 정보를 유지하기 위해 사용되며 특성에 따라 분류되어지고 색인되어 질 수 있다. 즉, 서비스 데이터는 그리드 서비스를 색인하기 위해 사용하며 서비스 데이터는 서비스의 특성 및 제공 정보에 따라 검색(search) 또는 통지(notification) 기법으로 사용한다.

### 2. 결합 검출기(Fault Detector)

그리드 컴퓨팅 시스템은 전통적인 분산 시스템에서와 같이 네트워크 분리(network partition) 및 노드 결함(node crash) 등에 의해 높은 결합율을 가질 수 있다[5].

결함은 시스템의 명세에서 정한 서비스를 더 이상 제공하지 못할 때 발생하며, 결합 포용 분산 시스템은 시스템의 구성요소의 결함에도 신뢰할 수 있고 지속적인 서비스를 제공할 수 있는 기능이 설계되어야 한다. 결합 포용(fault tolerance)는 그리드 시스템을 구성하는 계산 서비스의 종속성(dependability)을 증가시키기 위한 주요한 수단이다[5]. 그리드 수행 환경은 동적으로 컴퓨팅 자원 및 서비스의 상태가 변화되어지는 분산 환경이다. 이러한 그리드 컴퓨팅 환경에서 효율적인 작업 수행을 위해서는 환경의 변화에 적응하기 위해 정확하고 빠른 상태 정보 수집은 필수적이다.

## III. OGSA 기반 결합 검출기의 설계

본 절에서는 그리드 정보 서비스의 결합 포용을 위해 필수적인 OGSA 기반 결합 검출기의 설계를 기술한다.

## 1. 설계시스템 아키텍처

결합 검출기는 주기적인 모니터링을 수행하는 에이전트(monitoring agent)와 자원 상태 수집기(status aggregator)로 구성된다. 모니터 에이전트는 상태 모니터링의 대상에 따라 프로세스 모니터, 프로세서 모니터 및 네트워크 모니터로 나누어질 수 있다. 와 결합 상태 수집기로 구성된다.

<그림 1>은 설계되어진 결합 검출기의 아키텍처를 보인 것으로 것이다. 프로세스에 대한 모니터링 정보는 자원 정보에 대한 전달과정에서 piggy-backing을 통해 특정 프로세스에 대한 상태 정보를 전달함으로써 수집에 요구되는 네트워크 부하를 줄일 수 있다. 성능에 대한 정보는 브로커가 서비스 제공자에 대한 선택을 결정하는 과정으로 중요한 요소로서 활용할 수 있다. 설계된 검출기는 수집하는 자원 또는 서비스의 형태에 따라 수집의 전달 단위 및 회수를 조정함으로써 GT3의 자원 정보 수집기의 문제점을 해결하고 높은 성능을 보장할 수 있다.

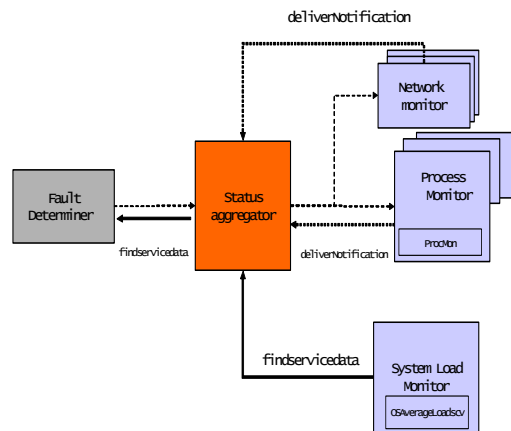
## 2. 상태 정보 수집 메카니즘

상태 수집기가 변화된 정보를 얻기 위한 방법은 pull 기반과 push 기반 수집으로 나눌 수 있다. Pull 기반 수집은 주기적으로 클라이언트가 서버에게 변경되어진 것을 요청하고 서버는 요청된 정보를 제공하는 방법이다. 그러나 pull 기반 수집 방법은 호출간의 간격이 작은 경우 네트워크 트래픽과 CPU 사용을 증가시키고, 다수의 클라이언트 요청이 있는 경우 시스템을 블로킹 시키는 문제를 발생시킨다.

<그림 2>는 push 기반 자원 정보 수집을 위한 시스템의 논리적 구성요소의 상호작용을 시퀀스 다이어그램으로 보인 것으로 facilitator는 상태 정보를 수집하고 Source1과 Source2는 각각 프로세서 및 프로세스에 대한 정보의 모니터링 에이전트의 역할을 수행한다. 통지 서비스는 그리드 서비스 내에서 발생되어지는 변경을 모든 구독자에게 전달하기 위해 사용한다. 상태수집기는 클라이언트는 전체 시스템의

상태를 모니터링하기 위해 변경 통지된 정보를 사용할 수 있다.

그리드 서비스들은 사용자에게 그리드 환경의 정보를 제공하는 도구이기 때문에 오류 없이 정상적으로 언제나 동작해야 한다. 만약 오류로 인해 정상적인 서비스를 수행할 수 없을 때에는 재시작을 통해 지속적인 서비스를 수행하도록 하여야 한다.



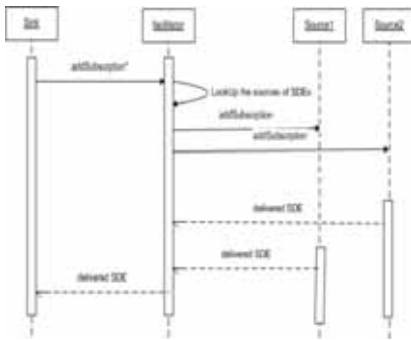
▶▶ 그림 1. OGSA 기반 결합 검출기 아키텍처

<그림 2>에서와 같이 상태 정보는 에이전트에 의해 수집기로 비동기적으로 상태정보를 제공하는 Push 기반 메카니즘을 사용하여 상태정보 수집을 하도록 모니터링 에이전트와 수집기를 설계한다. 또한 모니터링 에이전트는 프로세스가 구동하기 시작하면 작동한다. 그리고 주기적으로 그리드 서비스의 생존 여부를 정보 수집기에게 보낸다. 정보 수집기는 얻어진 정보를 바탕으로 정보수집 가시화를 통해 사용자에게 나타낸다.

## IV. 결합 검출기 구현

본 절에서는 프로세스의 결합 검출을 위한 정보 서비스의 생산자인 모니터링 에이전트 및 소비자인 상태 수집기에 대한 결합 발견 검출기의 구현을 기술한다.

<그림 3>은 현재 프로세서 모니터인 HeartBeat 모니터링 수행을 통해 프로세스 상태 정보를 보인 것이다. ProcessStatus() 연산과 수집된 정보를 유지하기 위한 ProcessStatus SDE를 사용한다. <그림 3>은 GT3의 서비스 브라우저를 통해 가시화되어진 시스템의 부하 정보를 모니터링하는 HeartBeat 서비스의 처리 내용을 보인 것으로 AbstractPortTypePanel 클래스를 확장한 HBbasePortTypePanel 클래스에서 처리되어진다.



▶▶ 그림 2. Push 기반 자원 정보 수집

### V. 결론

이 논문에서는 OGSA(Open Grid Service Architecture) 기반의 그리드 미들웨어인 GT3 상에서 자원 결합 검출을 통해 결합 포용을 갖는 높은 가용성의 보장을 위한 결합 검출 그리드 서비스 설계 및 구현을 기술하였다. 이를 위해 기존의 HeartBeat 서비스를 확장하여 프로세스 상태 정보를 수집하는 프로세스 정보 모니터와의 상호작용을 통해 정의된 프로세스의 상태를 주기적으로 수집하도록 하였다. 향후 시스템의 결합 처리를 구현한 예정이다.

Kaufmann, San Fransisco, CA, 1999.

- [2] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit" Intl. J. Supercomputer Applications, 11(2):115-128, 1997
- [3] Paul Stelling, Ian Foster, Carl Kesselman, Craig Lee, Gregor von Laszewski, "A Fault Detection Service for Wide Area Distributed Computations", Cluster Computing, Vol. 2, No. 2, PP. 117-128 1999.
- [4] Tushar Deepak Chandra, Sam Toueg: Unreliable Failure Detectors for Reliable Distributed Systems. J. ACM 43(2): 225-267 (1996)
- [5] Barry W. Johnson, An Introduction to the Design and Analysis of Fault-Tolerant Systems, in Fault-Tolerant Computer System Design, Dhiraj K. Pradhan, Prentice Hall, Inc., 1996.
- [6] W3C, Web Services Description Language (WSDL) 1.1,2001, <http://www.w3c.org/TR/wsdl>
- [7] T. Sandholm, R. Seed, and J. Gawor, "Globus toolkit 3 core-a grid service container framework," tech. rep., Globus, January 2003



▶▶ 그림 3. 수집된 프로세스 상태 정보

#### ■ 참고문헌 ■

- [1] I. Foster and C. Kesselman. The Grid: Blueprint for a New Computing Infrastructure Morgan