

# 웹 클러스터 환경을 위한 신뢰성 있고 확장성 있는 이벤트 채널 서비스의 설계와 구현

안형근, 이용환, 민덕기  
건국대학교 컴퓨터·정보통신공학과  
E-mail:{h96ahg, yhlee, dkmin}@konkuk.ac.kr

## Design and Implementation of Scalable and Reliable Event Channel Service in Web Cluster Environment

Hyunggeun An, Yonghan Lee, Dugki Min  
Department of Computer Science and Engineering, Konkuk University

### 요약

인터넷이 확산되고 일반화되면서 가용성, 확장성이 우수한 저가형 웹 클러스터 시스템이 많이 사용되고 있다. 웹 클러스터 시스템을 효과적으로 관리하기 위해서는 클러스터 시스템을 구성하는 다양한 컴포넌트들에서 발생하는 이벤트를 직접 또는 간접적으로 전달해 주는 이벤트 전달 서비스가 필요하다. 본 논문은 웹 클러스터 환경에서 컴포넌트들 간에 발생하는 이벤트를 간접적으로 전달하는 이벤트 채널 서비스를 제안한다. 이 이벤트 채널 서비스는 이벤트 생산자와 소비자가 채널을 통하여 Push 또는 Pull 방식으로 이벤트를 전달 할 수 있는 구조를 제공하며, 전달 시 필요에 따라 다양한 QoS 정책과 Filtering 기법을 적용할 수 있는 유연한 구조로 설계되었다. 또한 웹 클러스터의 구조와 적절히 결합되어 시스템 장애에도 불구하고 계속적으로 동작할 수 있으며 멀티 클러스터 구성 시 확장성이 우수하다.

### 1. 서론

최근 인터넷의 사용의 폭발적 증가로 인하여 대용량의 분산 웹 서버와 데이터베이스 서버, VOD 서버 및 고성능의 컴퓨팅 파워를 필요로 하는 서비스가 증가되었다. 이러한 서비스들의 증가로 저가의 컴퓨터들을 연결하여 하나의 대형 컴퓨터처럼 사용하는 클러스터 시스템의 사용이 점차 확산되고 있다. 이러한 클러스터링 기술은 가용성(availability), 관리성(manageability), 확장성(scalability)과 같은 장점들을 지니고 있다

클러스터 시스템의 관리를 효과적으로 제공하고 장애예방을 하기위하여 시스템에서 발생하는 이벤트들을 안정적으로 전달하는 통신 서비스가 필수적이다. 시스템 컴포넌트 간에 이벤트를 전달하는 방식에는 두 가지가 있을 수 있다. 첫 번째는 이벤트 공급자(Supplier)와 소비자(Consumer)간 직접 연결하

는 방식(Direct Communication)이며, 두 번째는 중재자를 경유하는 간접통신 방식(Indirect Communication)이다. 공급자와 소비자를 직접 연결하는 방식은 빠른 실시간 전달을 필요로 하는 경우 적당하다. 하지만 공급자와 소비자를 서로 모르게 분리시킬 필요가 있는지 Pull 방식의 전달이 필요한 경우에는 간접 통신 방식이 적당하다. 본 논문에서 고려하는 이벤트 채널 서비스는 공급자와 소비자 간에 간접적인 통신 수단인 채널을 제공한다. 이 채널은 다중 공급자와 다중 소비자 간의 통신을 제공하는 중재자 역할을 제공한다.

본 논문은 웹 클러스터 환경에서 컴포넌트들 간에 발생하는 이벤트를 간접적으로 전달하는 이벤트 채널 서비스를 제안한다. 이 이벤트 채널 서비스는 이벤트 생산자와 소비자가 채널을 통하여 Push 또는 Pull 방식으로 이벤트를 전달 할 수 있는 구조를

제공하며, 전달 시 필요에 따라 다양한 QoS 정책과 Filtering 기법을 적용할 수 있는 유연한 구조로 설계되었다. 또한 웹 클러스터의 구조와 적절히 결합되어 클러스터 환경에서 노드(공급자와 소비자)들이 동적으로 추가 삭제되는 환경과 다양한 시스템 장애 환경에도 불구하고 신뢰성 있는 통신 서비스를 제공하며 멀티 클러스터 구성 시 확장 가능하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 클러스터 환경에서의 이벤트 채널 서비스의 필요성을 설명하고 제안하는 이벤트 채널 서비스의 구조는 3장에서 기술한다. 4장에서는 제안한 이벤트 채널 서비스와 클러스터 환경과의 결합구조를 제시한다. 5장에서 결론을 맺는다.

## 2. 웹 클러스터 환경과 이벤트 채널 서비스

이 장에서는 고려하는 웹 클러스터 환경에 대해서 설명하고 여기서 왜 이벤트 채널 서비스가 필요한지에 대해서 설명한다.

### 2.1 웹 클러스터 환경

웹 서비스 로드 밸런싱 클러스터 시스템은 4가지 종류가 있다. Layer 4 스위치를 사용하는 하드웨어 레벨, Linux Virtual Server와 같이 OS 기반에서 제공하는 시스템 레벨, WAS(Web Application Server)를 사용하는 미들웨어 소프트웨어 레벨과 DNS(Domain Name Server)를 사용하는 애플리케이션 소프트웨어 레벨이 있다. 우리의 이벤트 채널 서비스는 시스템 소프트웨어 레벨을 기반으로 개발된 웹 클러스터 로드밸런싱 시스템 환경에서 개발되었다. 그림 1은 웹 클러스터 로드 밸런싱 시스템의 구성이다.

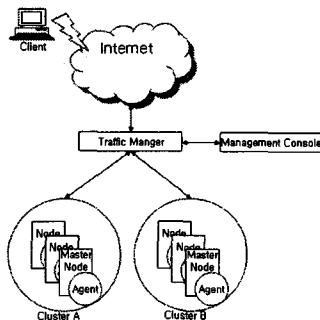


그림 1 클러스터기반의 웹 서비스 로드 밸런싱

그림 1에서 Cluster A, Cluster B는 웹 서비스를 제공하는 웹 클러스터이다. 클러스터내의 노드(Node)들은 웹 서비스를 제공하는 서버들이다. 그림

의 Traffic Manager는 각 Cluster A, Cluster B 내의 Master Node와 통신하면서 각 클러스터 내의 노드들 간의 로드밸런싱을 한다. Master Node는 클러스터 내의 노드들에 대한 상태정보를 유지하며 Traffic Manager와 정기적으로 통신하며 Cluster 내의 노드들 정보를 Traffic Manager와 동기화한다. Management Console은 Management System에 위치하고 있으며 전체 클러스터에 대한 정보를 가지고 있으며 클러스터 관리(클러스터 생성, 노드 추가, 삭제 등)를 책임진다.

### 2.2 웹 클러스터 환경에서의 채널 서비스 필요성

웹 클러스터 환경에서 채널 서비스의 필요성은 세가지가 있다. 첫번째는 이벤트를 제공하는 공급자와 이벤트를 사용하는 소비자가 동적으로 추가, 삭제되어야 한다. 두 번째는 클러스터와 클러스터 외부에 방화벽과 같은 어떤 요인에 의해서 직접 통신할 수 없는 경우이다. 마지막으로 이벤트 전달에 있어서 Pull, Push 모드 지원과 Qos, Filtering 지원이다. 첫번째는 웹 클러스터내의 노드(공급자, 소비자)들은 수시로 클러스터에 합류하거나 탈퇴한다. 이러한 동적인 환경에서 이벤트를 주고받기 위해서는 등록 기반으로 이벤트를 전달하는 채널 서비스가 필요하다. 두번째는 클러스터 내부의 노드들이 사설 IP로 되어 있거나 클러스터들 사이에 방화벽이 존재하는 경우에는 서로 다른 클러스터에 속한 노드들 간에는 직접 통신을 사용할 수 없다. 이러한 경우에는 클러스터 외부에 존재한 전역채널(Global Channel)을 사용할 수 있다. 전역 채널의 주소와 포트를 방화벽에 등록하여 클러스터 간에 간접통신으로 이벤트를 전달할 수 있다. 세번째는 웹 클러스터 내에서 발생하는 이벤트들은 시스템 성능에 직접적인 영향을 미치는 실시간으로 전송해야 할 이벤트들도 있지만 시스템 Performance Reporting에 필요한 이벤트 같이 시스템 운영이나 성능에 필요하지 않은 이벤트들도 있다. 긴급하지 않은 이벤트들은 채널로 전송해서 채널이 제공하는 다양한 모드, QoS, Filter를 통해서 채널에 등록된 소비자에게 전송된다. 즉 채널 서비스를 사용하게 되면 이러한 다양한 부가적인 기능들이 가능하다.

### 3. 이벤트 채널 서비스

이 장에서는 이벤트 채널 서비스 구조를 설명하고 각 구성요소들의 기능들을 설명한다.

### 3.1 이벤트 채널 구조

우리가 제시하는 이벤트 채널 모델에서는 이벤트 공급자, 소비자와 Proxy, Admin, Channel이 존재한다. 그림 2는 우리의 이벤트 채널 서비스 모델이다.

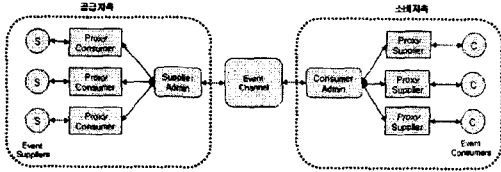


그림 2 이벤트 채널 모델

공급자(S)는 이벤트를 발생시키며 채널 서비스의 ConsumerProxy로 전송한다. ConsumerProxy는 이벤트를 Channel로 전송하며 채널은 이벤트를 소비자(C)측의 Supplier Proxy들은 채널의 이벤트들을 소비자에게 전송한다. 우리의 이벤트 채널 모델에서는 한 채널 당 여러 개의 공급자와 소비자가 존재할 수 있다. 공급자와 소비자측에는 1개의 Admin이 존재하며 Supplier Admin은 Supplier Proxy를 Consumer Proxy은 Consumer Proxy를 관리한다. Proxy들은 Pull또는 Push 타입을 제공한다.

그림 3은 이벤트 채널 서비스의 구조를 설명한다. 이 그림의 Administrator, Supplier는 CommunicationManager를 통해서 채널 서버와 통신을 한다. 이때 CommunicationManager는 내부의 Communication Adapter를 변경함으로써 다양한 통신 프로토콜(UDP, TCP, RMI)을 동적으로 변경할 수 있다. 채널 서버는 초기화되면서 Default Channel을 생성한다. 다음으로 Proxy들을 그룹 형태로 관리할 SupplierAdmin, ConsumerAdmin을 생성하고 난 후에 공급자 쪽에 Push기반의 Default Proxy를 생성하고 소비자 쪽을 위해서 Push 기반의 Default SupplierProxy를 생성한다. 이러한 Default Proxy들은 단지 Push기반의 공급자나 소비자와 같은 Default Proxy에서 정한 QoS 사용할 경우에 사용하고 나머지 Pull기반이나 QoS가 다른 경우에는 공급자나 소비자당 하나의 Proxy를 생성해서 채널을 통해서 이벤트를 전달한다.

채널서버 안의 큐의 자료 구조는 QoS의 Ordering에 대한 Policy에 따라서 FIFO일 경우에는 Circular Queue를 사용하고 이벤트의 Priority별로 처리를 해야 할 경우에는 LinkedList기반의 Queue를 사용했다. 이와 같이 채널서버는 Policy에 따라서 언제든지 자료 구조를 변경할 수 있는 유연한 구

조이다. 그리고 채널 안의 Job Scheduler가 하는 역할은 Push기반의 ConsumerProxy의 수가 많을 때 Proxy안의 Buffer안에 Structured Event가 쌓여 있을 때 어떤 Proxy를 먼저 가져올 것인가. 또한 가져올 때 어떤 주기 형태로 가져올 것인가에 대한 Job Scheduler 역할을 수행한다. 마찬가지로 Push기반의 SupplierProxy안의 Queue에 이벤트를 전달할 때 어떤 순서 별로 할 것인가에 대한 스케줄 작업을 한다.

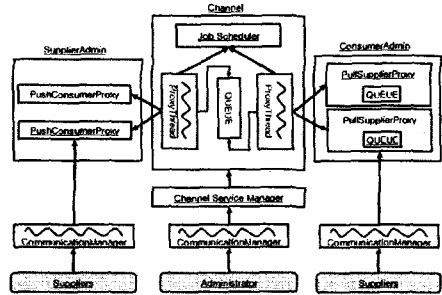


그림 3 Event Channel Service 구조

### 4. 웹 클러스터와 이벤트 채널 서비스 적용

이 장에서는 클러스터 환경에서 실제로 이벤트 채널을 어떻게 적용하고 동작되는지에 대해서 설명한다.

#### 4.1 웹 클러스터내의 채널 서비스

기존 채널 서비스들은 소비자와 공급자가 채널을 찾기 위한 방법으로 Directory Service를 제공하고 있다. 그러나 마스터 노드가 존재하는 웹 클러스터 환경에서는 Directory Service가 필요하지 않다. 그림 4는 클러스터 상황에서 채널 서비스의 동작을 나타낸다.

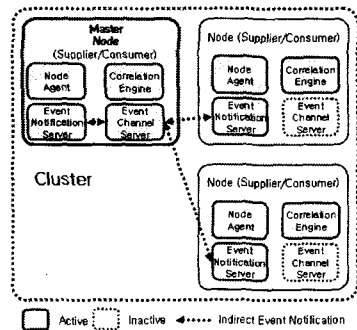


그림 4 클러스터내의 Event Channel Service 동작 클러스터내의 채널서버는 각 노드에 Node Agent, Event Notification Server와 함께 존재한다. 그러나

채널 서버는 마스터 노드에서만 동작한다. 클러스터가 생성된 후에 마스터 노드 선출 알고리즘에 의하여 마스터 노드가 결정되면 마스터 노드는 자신의 시스템에 존재하는 채널서버를 실행한다. 그리고 마스터 노드는 자신이 마스터 노드이고 채널 서버라는 것을 나머지 노드들에게 통지한다. 클러스터가 생성된 이후에 관리시스템과 마스터 노드간에 시스템에 장애(마스터 노드 과부하, 시스템오류, 네트워크 오류)로 통신이 불가능하다면 새로운 마스터 노드를 선출한다. 마스터 노드 선출 알고리즘에 의해 새로 선출된 마스터 노드는 채널서버를 가동하고 자신이 마스터 노드라는 것을 나머지 노드에게 통지한다. 마스터 노드 이외의 노드들은 새로운 마스터 노드가 선출되었다는 통지를 받으면 이벤트 통지 서버(Event Notification Server)는 채널 서버의 주소로 마스터 노드의 주소로 변경한다.

#### 4.2 클러스터간의 전역 채널 서비스

웹 클러스터에서는 어떠한 이유에 의해 각 클러스터간에 직접 통신이 불가능 할 수 있다. 이러한 경우에는 전역 채널(Global Channel)을 간접 통신을 사용해야 한다. 그림 5에서는 전역채널을 이용한 클러스터간의 간접 통신 예를 보여주고 있다.

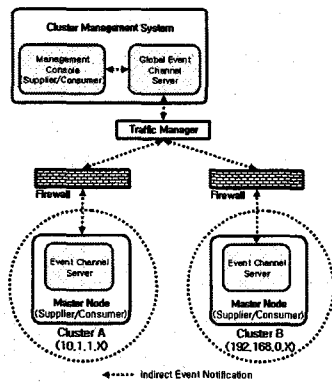


그림 5 Global Channel을 이용한 Indirect Communication

그림 5와 같이 Cluster A, B 사이에 방화벽이 존재하거나 클러스터 안의 노드들이 사설 IP로 되어있는 경우에는 각 클러스터 안의 노드들은 직접 통신이 불가능하다. 이런 경우 Cluster Management System 안에 있는 Global Event Channel의 Public IP와 Port를 방화벽에 등록해서 클러스터 간에 이벤트 전달이 가능하다. 클러스터 노드들이 사설 IP로

되어 있는 경우에도 Global Channel을 경유해서 다른 클러스터의 노드로 이벤트를 전달 할 수 있다.

#### 5. 결론

본 논문에서 제시한 채널서버는 클러스터 환경에서 마스터 노드 선출 알고리즘과 연동하여 어떠한 장애 상황에도 견고하게 서비스를 할 수 있으며 또한 한 클러스터 내의 노드의 수가 가변적인 상황에서도 적용할 수 있는 확장성 있는 구조를 가지고 있다. 그리고 각 노드들의 다양한 QoS나 다양한 통신 프로토콜을 수용할 수 있는 상당히 유연한 구조이다. 또한 클러스터간의 이벤트 전달을 할 수 있으며 클러스터가 추가되는 상황에서도 쉽게 클러스터 간 이벤트 채널을 통한 통신을 지원 할 수 있는 확장성이 있는 채널 시스템이다.

#### 6. 참고 문헌

- [1] Rene Meier "State of the Art Review of Distributed Event Models" Department of Computer Science, University of Dublin.
- [2] Kanaka Juvva and Raj Rajkumar "A middleware Service for Real-Time Push-Pull Communications" Real-Time and Multimedia Laboratory Carnegie Mellon University.
- [3] Yeturu Aahlad "Asynchronous Notifications Among Distributed Objects" SunSoft Inc.
- [4] Greg Eisenhauer, Fabian E. Bustamante "Event Services in High Performance Systems" College of Computing Georgia Institute of Technology
- [5] Subrata Mazumdar, Isabelle Rouvellou "Design of Filterable Event Channel: Building Block for CORBA-Based Notification Services" Bell Laboratories, IBM T.J. Watson Research Center.
- [6] Douglas C. Schmidt, Steve Vinoski "Object Interconnections" Department of Computer Science Washington University, IONA Technologies, INC.
- [7] Sergio Duarte, Jose Legatheaux Martins "DEEDS - a Distributed And Extensible Event Dissemination"
- [8] Xavier Defago "Reliability Issues with CORBA Event Channels"
- [9] Eunmi Choi, Dugki Min "Performance Test and Analysis for an Adaptive Load Balancing Mechanism on Distributed Server Cluster System" Handong Global University, Konkuk University