Jini를 이용한 웹 서버 클러스터의 자율 관리 기법

이철, 박규호
한국과학기술원, 전자 전산학과
전화 : 042-869-5425 / 핸드폰 : 016-592-3205

Autonomous Management of a Web Server Cluster Using Jini Technology

Chul Lee, Kyuho Park
Dept. of EECS, KAIST
E-mail : chullee@core.kaist.ac.kr

Abstract

In this paper, we consider issues related to the autonomous management of a server cluster. Server clustering technique is widely used for high available Internet servers such as a web server, because it provides very scalable and reliable service. However, it is very burden-some to manage the cluster system. For instance, installation or configuration of a new node requires skilled administrators. Also, when a node is failed, it should be recognized and recovered by a human. The management complexity gets higher while increasing the number of machines for higher throughput. Here, we are going to alleviate the management complexity by building an autonomous cluster server with Jini ™ technology from Sun Microsystems

I. 서론

여러대의 PC를 클러스터링하여 고가용성 고확장성 서버 시스템을 제작하는 기법은 매우 성공적이었다. 특히, 최근의 인터넷 서버 시스템과 같이 높은 확장성을 필요로 하는 분야에 많이 적용되고 있다. 그러나, 여러대의 컴퓨팅 기계들을 소수의 관리자가 관리해야 하는 경우, 기술적으로 숙련되어 있다 하더라도 실수를 피하기 어려우며 이로 인한 치명적인 오류도 발생할 수 있다. 본 논문에서는 관리자의 간섭을 최소화하며 자율적인 관리 기능을 가지는 자율 클러스터 컴퓨팅을 제안한다. 관리작업은 크게 소프트웨어의 설치, 설정, 최적화, 복구등으로 분류될 수 있으며, 하드웨어 설치를 제외한 소프트웨어 관리 요소들을 자동화 한다.

Jini[1]는 이동성 기기들을 위한 네트워크 인프라를 제공하는 미들웨어로써, 자바 언어와 RMI 기법을 이용하여 기기 및 서비스 검색 및 접근제어 인프라스트럭처, 리스를 통한 고장감내, 트랜잭션 지원등에 필요한 프로그래밍 모델 제공한다.

서버 클러스터에서 각 구성요소들의 출입이 고장 및 추가 업데이트등으로 인하여 빈번하다고 볼 수 있으며, 이러한 측면에서 이동성 기기/서비스를 위한 Jini 네트워크 기술을 적용 할 수 있다고 본다.

II. Background

2.1 Jini

Jini는 Sun Microsystems에서 개발하였고, 동적인 환경 하에 산재하는 이동성 기기들간의 네트워크를 형성하기 위한 인프라 및 프로그래밍 모델을 제공한다. 이동성 기기의 사용이라는 측면에서 '기기'와 '서비스'를 개념적으로 혼용하여 사용하며, 모두 Java 객체로 표현 될 수 있다. Jini 는 JVM 환경에서 동작하며, 표 1 에서와 같이 Java 의 기반기술을 사용하면서 이동성 기기간의 네트워크를 위한 별도의 인프라 및 프로그래밍 모델을 제공한다.

Jini 에 대응하는 기술들

이동성 기기간의 네트워크를 형성하기 위한 기반 기술

은 하드웨어와 미들웨어를 포함한다. 블루투스가 그 대표적인 경우로, PDA, 노트북, 이동전화 등을 포함하는 다양한 이동성기기들 간의 초근거리 주파수를 제공하며, 기기들의 이동에 따른 기기 검색 및 서비스 접근 제어등을 위한 네트워크를 형성한다. HAVi는 가전 정보 기기들간의 홈 네트워크를 구성하기 위한 기술정의이다. 기기들의 이동에 따라 네트워크가 자동으로 설정되며, 음향기기들간의 협동을 큰 목표로 하고 있다. uPnP 는 Jini 의 목표와 가장 유사한 경쟁상대로써, MS의해 주도적으로 표준화 및 지원이 이루어지고 있다. 독립적인 PC에서의 플러그앤플레이의 개념을 네트워크를 통한 이동 기기에 적용 시킨 예이다.

Jini 의 구성요소

Jini 가 가정하는 분산 환경에서는 크게 3가지 요소가 있다. 사용자, 서비스 및 기기, 색인 서비스가 그것으로써, 표2에 그 역할을 설명하였다.

2.2 시스템 설정 자동화 기법

LCFG[2] 와 SmartFrog[3]는 많은 수의 시스템에 소프트웨어를 설치하고 관리하는 것을 자동화 할 수 있도록, 설정에 대한 정의와 사용자 인터페이스를 제공한다. 매우 다양한 구성요소들을 포함하고 설정들이 빈번하게 변하는 경우 관리자의 노력을 최소화한다. 그러나, 본 논문에서 의도하는 것과 같이 성능 모델과 모니터링을 통한 시스템 조율 기능은 없다.

III. 관리대상 시스템 및 관리요소들

자율관리를 목표로하는 대상 시스템은 그림 1과 같은 일반적인 웹 서버 클러스터이다. 여러대의 PC들이 Ethernet 과 SAN 으로 연결되어 있다. 각 PC 들은 디스패쳐, 웹 서버, DB 서버, 혹은 공유 파일시스템을 위한 메타데이터 서버로 기능할 수 있으며, SAN을 통하여 데이터 저장장치는 공유된다.

관리자의 역할은 표 3 에서 보는 것과 같이 소프트웨어의 설치, 설정, 최적화, 그리고 복구로 나눌 수 있다. 각 작업들은 관리자에 의하여 처리되어야만 했다.

표 1. Jini 와 Java 의 비교

	Infrastructure	Programming model	Services
Base Java	Java VM RMI 자바 보안	Java API Java Beans	JNDI JTS J2EE Beans
Java + Jini	검색/조인 분산 보안 색인	리스 트랜잭션 이벤트	Printing JavaSpace

표 2. Jini 의 구성요소

요 소	역 할
서비스 Service	이동성 기기 혹은 소프트웨어를 포함하며 Java 객체로 표현된다. 네트워크에 연결될 때, 색인 서비스를 검색하여 사용을 위한 인터페이스를 등록한다.
사용자 Client	서비스를 사용하고자 하는 사람 혹은 다른 기기로써, 원하는 서비스를 사용하고자 할 때 네트워크에서 색인 서비스를 탐색하여, 필요한 서비스가 등록되어 있는지 검색한다.
색인 LUS	주변 네트워크에 존재하는 등록된 서비스들에 대하여 색인 서비스를 제공한다. 사용자에 의해 미리 인지되어 있는 Java 인터페이스를 바탕으로 검색을 진행하며, 리스 기간을 바탕으로 등록된 기기들이 정상 작동중인지 주기적으로 확인한다.

표 3. 관리 요소

관리 요소	설명
설치	소프트웨어가 자동으로 설치되고 업데이트 되어야 한다.
설정	개별 노드의 설정이 자동으로 수행되어야 한다.
최적화	전체 시스템에서 병목지점을 제거하기 위하여 항상 구성요소들을 모니터링 하고 조율해야 한다.
복구	크래쉬 되었을때 관리자의 도움 없이도 스스로 복구 되어야 한다.

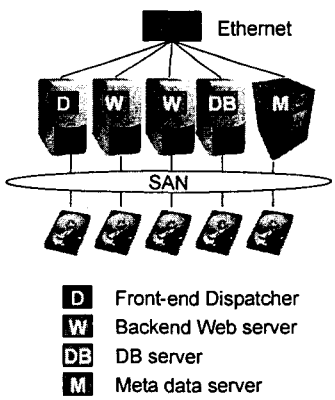

그림 1 관리대상 웹 서버 클러스터 구조

Ⅳ. 자율 관리 클러스터 구조

본 장에서는 웹 클러스터의 구성요소를 Jini의 구성요소에 대응하여 서비스와 사용자를 정의하고 그 구조를 설명한다. 색인 서비스는 별도의 관리 서버를 활용한다.

먼저, 디스패쳐는 웹 서버를 사용하는 사용자로 구분할 수 있다. 새로운 웹 서버가 클러스터에 연합하려면 관리 서버가 제공하는 색인 서비스를 멀티캐스팅을 통하여 탐색하고, 자신을 서비스로써 등록한다. 차후, 디스패쳐는 추가의 웹 서버가 요구될 때, 관리 서버의 색인 서비스를 활용하여, 웹 서버로 사용할 수 있도록 등록된 서비스를 검색한다. 또한, DB서버는 웹서버가 활용할 수 있는 서비스로 정의 할 수 있고, 디스크는 메타데이타 서버가 활용할 수 있는 서비스로 정의 할 수 있다.

그림 2 에서 도시된 바와 같이 서비스(웹서버)가 네트워크에 조인할 때 색인(관리서버:LUS)에 자신을 등록하고, 이를 필요로 하는 사용자(디스패쳐)는 LUS를 검색하여 등록된 서비스를 찾아낸다. 이와 같이 사용자와 서비스를 직접 연결하지 않고, LUS를 탐색/검색/등록 과정을 거치는 것은 환경 설정에 있어서 동적 특성을 제공한다.

상술한 바와 같이 자율 관리 시스템의 네 가지 목표 중 첫 번째 자율 설정 기능이 완성되었다. 웹 서버를 새로 설치할 때 Jini 기반의 자율 관리 에이전트를 실행하기만 하면, 디스패쳐의 설정 파일을 Jini 인프라가 스스로 수정하여, 웹 서버를 클러스터에 연합 할 수 있도로 한다. 디스패쳐는 현재 클러스터 사용량을 모니터링 하며, 추가의 웹서버가 필요한 시점을 스스로 알아내고, LUS 에 문의하여 사용가능한 유휴 웹서버가 있는지 검색한다.

표 4. 웹클러스터 구성요소와 Jini 의 구성요소

Jini 구성요소	웹 클러스터 구성 요소
서비스 (Service)	웹서버, DB서버, 디스크
사용자 (Client)	디스패쳐, 웹서버, 메타데이타 서버
색인 (LUS)	관리서버

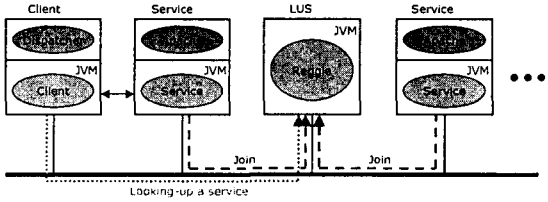

그림 2 클러스터 노드들의 관계 및 시스템 구조

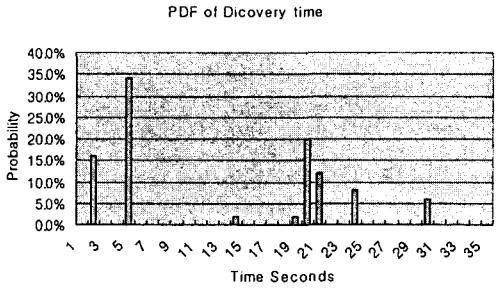

그림 3 서비스 탐색/검색시간의 확률분포도

그림 3은 새로운 웹서버가 LUS에 등록 되었있다고 가정할 때, 디스패쳐가 추가의 웹 서버에 대한 요구를 감지한 시점에서 기 등록된 웹 서버를 클러스터에 연합하기 까지 걸린 시간의 확률 분포를 나타낸 그래프이다. 기기의 이동성을 감안한 네트워크 특성상 서비스를 탐색하는데 멀티캐스팅을 사용하기 때문에 탐색 및 검색이 완료되는 시간의 분포가 6초내에 50%가 위치하며, 20초 이상의 시간이 걸리는 경우도 30%이상이다. 이것은 동적으로 빠르게 환경에 적응해야 할 필요성이 있는 응용 분야에는 적합하지 않으며, 별도의 탐색 프로토콜을 연구해야 할 것이다.

Ⅴ. 결론

본 연구에서는 관리를 스스로 수행하는 클러스터 서버를 그 목표로 하였다. 관리 이슈는 크게 설치, 설정, 최적화, 복구로 나뉠 수 있으며, 본 연구를 통하여 현재 새로운 노드가 설치되어 클러스터에 연합하려 할 때 필요한 일련의 설정 작업을 자율적으로 수행하도록 하였다.

서비스의 탐색 및 검색에 필요한 시간을 측정한 결과 동적이고 빠른 적응력이 필요한 환경에서 Jini를 기반 미들웨어로 채택하는 것은 어려움이 있음을 알게되었다. 이 경우 멀티캐스팅 보다 빠른 응답성을 보이는 별도의 프로토콜을 채택해야 한다.

그밖에 디스크의 설정, DB 서버의 설정을 위한 개발이 필요하며, 자율 최적화와 자율 치료를 위한 기법을 위한 연구가 필요하다. 자율 자율 최적화를 수행하기 위해서, 클러스터 노드들은 성능 병목 지점들을 제거하기 위하여 항시 자신의 구성요소들을 모니터링 하여야 한다. 또한, 다수의 리소스들을 적절히 재분배하여 운영 효율을 높일 수 있는 기법이 향후 제공되어야 한다.

참고문헌(또는 Reference)

[1] Jini Network Technology, http://www.sun.com/software/Jini.xml

[2] LCFG: A Large-scale UNIX configuration System, http://www.lcfg.org/

[3] SmartFrog: Configuration and Automatic Ignition of Distributed Applications, A Technical overview paper from the 2003 HP Openview University Association conference