

그룹 통신을 이용한 서버리스 응용 프로그램의 구현

박재준, 장태무
동국대학교 컴퓨터 공학과

Implementing Serverless Applications Using Group Communication

Jaejun Park, Taemu Chang
Dept. of Computer Engineering, Dongguk University

요 약

기존의 클라이언트/서버 응용 프로그램은 공유 자원을 서버에서만 관리하므로 자원의 가용성(availability)이 떨어지고, 클라이언트의 수가 많아질수록 서버의 과부하로 인해 시스템 전체의 성능이 저하되는 등의 단점을 내포하고 있다. 이는 응용 프로그램을 사용하는 대다수의 클라이언트가 소수의 서버에만 의존하는 시스템 구조에서 비롯되는 문제라고 할 수 있다. 이러한 문제점들을 해결하고자 고성능의 고가형 서버를 도입하거나, 서버의 작업을 클라이언트에 분배하는 등의 연구가 있었지만, 문제의 근본적인 해결책이 될 순 없었다.

본 논문에서는 이러한 문제점들을 해결하고자 그룹 통신을 응용한 서버리스 분산 응용 프로그램을 제안하고, 간단한 문서관리 분산 응용 프로그램을 실제로 구현함으로써 서버리스 응용 프로그램의 특성들을 소개하고자 한다.

1. 개 요

거대한 메인프레임에 터미널들을 연결하여 사용하던 중앙 집중식 시스템에서는 모든 터미널들이 메인프레임 한 대에만 의존하게 되므로 메인프레임에 문제가 생기면 전체 시스템을 사용할 수 없게 되는 경우가 있고, 터미널의 수가 많아질수록 응답 속도가 느려지는 문제점을 가지고 있었다[1].

이러한 문제를 해결하고자 대두된 것이 클라이언트/서버 구조의 응용 프로그램이다. 클라이언트/서버 구조의 응용 프로그램은 중앙 집중식 응용 프로그램에서 서버만 행하던 작업들을 클라이언트 부분과 서버 부분으로 양분하는 역할을 했다. 하지만 클라이언트용 프로그램과 서버용 프로그램이 별도로 필요하게 되어 중앙 시스템을 둘로 나누는 것 이상의 이점을 가져오진 못했다[1].

클라이언트/서버 구조의 응용 프로그램은 중앙 집중식 시스템이 가지고 있는 다음과 같은 문제점들을 고스란히 내포하고 있다. 그 첫 번째가 공유 자원을 서버에서만 관리하게 되어 자원의 가용성이 떨어진다는 것이다. 예를 들어, 응용 프로그램이 실행되는 도중에 서버가 문제를 일으키면 다수의 클라이언트들은 아무 것도 할 수 없게되어 서버가 복구될 때까지 기다려야 한다. 두 번째 문제점은 시스템의 확장성이 떨어진다는 것이다. 응용 프로그램에 참여하는 클라이언트의 수가 많아질수록 서버의 작업 부하가 커져서 과부하가 발생하게 되고 결국엔 이로 인해 전체 시스템의 성능이 떨어진다.

위와 같은 문제점들은 응용 프로그램을 어떻게 구현하느냐에 관한 문제라기보다, 대다수의 클라이언트들이 소수의 서버

에만 의존하는 응용 프로그램의 근본 구조에 관한 문제라고 볼 수 있다.

따라서 이 문제들을 해결하고자 등장했던 고성능, 고가형 서버를 사용한 시스템이나, 서버의 작업을 어느 정도 클라이언트에 분배하는 시스템 등은 근본적인 해결책이 될 순 없다.

이에 본 논문에서는 그룹 통신을 응용한 서버리스 분산 응용 프로그램 구조를 제안하고, 간단한 문서관리 응용 프로그램을 실제로 구현하여 기존의 클라이언트/서버 시스템의 문제점들을 해결해 보고자 한다.

본 개요 설명에 이어서 2장에서는 분산 응용 프로그램에서의 그룹 통신의 응용 및 윈도우즈 플랫폼에서의 그룹 통신 구현에 대해 살펴보고, 3장에서는 서버리스 분산 응용 프로그램의 구현에 대해서 설명하고 위에서 언급한 클라이언트/서버 응용 프로그램의 문제점들을 다시 한번 살펴본다. 마지막 4장에서는 결론 및 향후 연구 방향에 대해 언급하기로 한다.

2. 그룹 통신

2.1. 그룹 통신과 분산 응용 프로그램

1 대 all 통신을 기본으로 하는 방송(broadcast) 프로토콜과, 1 대 n 통신을 기본으로 하는 멀티캐스트 프로토콜을 통틀어 그룹 통신이라고 일컫는다. 이 중 방송 프로토콜은 LAN, 정지 위성, 기지국 기반의 라디오 시스템에서 제공하고 있고, 에더넷이나 링 구조의 네트워크에서는 멀티캐스트 프로토콜도 사용할 수 있다. 방송 프로토콜은 고용량의 멀티미디어 데

이터를 사용하는 응용 프로그램에 많이 응용되고 있으며, 향후 기가비트 단위의 고속 LAN에서도 방송 프로토콜을 제공할 전망이다[2].

이러한 그룹 통신을 응용 프로그램에서 사용하면 통신에 필요한 메시지 수를 줄일 수 있으며, 특히 다음과 같은 사항들을 요구하는 분산 응용 프로그램인 경우에는 그룹 메시지를 유용하게 사용할 수 있다[2][3].

- 자원 중복(replication)을 이용한 고장 감내성 향상 : 여러 대의 서버 사이에 자원을 중복시켜 그룹 통신을 이용해서 관리하게 되면, 한 서버가 이상을 일으켜도 클라이언트들은 계속 서비스를 받을 수 있다.
- 자원 중복을 이용한 성능 향상 : 공유 데이터가 갱신된 내용을 응용 프로그램에 참여하고 있는 모든 클라이언트들에게 그룹 메시지를 통해 알릴 수 있다.
- 분산 서비스에서 서비스 객체의 위치 결정 : V 시스템에서는 그룹 메시지를 사용하여 서버 그룹에 찾고자 하는 자원을 요청하여 자원을 획득한다.
- 다중 갱신(multiple update) : 어떤 이벤트가 발생했을 때, 관련 그룹에 속한 프로세스들에게 그룹 메시지를 사용하여 알릴 수 있다[3][4][5].

하지만 운영체제에서 그룹 통신을 지원하는 경우가 드물기 때문에 응용 프로그램에서 사용하기가 쉽지 않다. 실제로, 현재 사용되고 있는 많은 수의 분산 응용 프로그램이 네트워크에서 제공하는 그룹 통신을 충분히 활용하지 못하고 있어서 네트워크의 가용 자원의 활용도를 극대화시키지 못하고 있으며, 또한 1대1 통신을 위주로 하여 네트워크의 대역폭을 낭비하고 있는 실정이다.

2.2. 윈도우즈 플랫폼에서의 그룹 통신 구현

분산 응용 프로그램에서 그룹 통신을 이용할 경우 얻을 수 있는 이점이 많지만 운영 체제에서 그룹 통신을 제공하는 경우는 드물기 때문에 상용 시스템에서 그룹 통신을 사용한 예는 많지 않다. Kaashoek와 Tanenbaum은 Amoeba 분산 운영 체제에서 그룹 통신을 할 수 있는 프리미티브들을 구현한 바 있고[2][5], 이 외에도 Isis 시스템, V 시스템 등에서 그룹 통신을 이용했지만 윈도우즈 플랫폼을 기반으로 하는 응용 프로그램에서 그룹 통신을 사용한 예는 찾기 힘들다[2].

윈도우즈 플랫폼에서 그룹 통신을 사용할 수 있는 방법에는 윈도우즈 소켓을 이용하는 방법이 있다. 소켓을 이용하여 UDP(User Datagram Protocol)나 IPX(Internet Packet eXchange)를 통하여 그룹 메시지를 전송할 수 있다.

또 다른 방법은 윈도우즈에서 제공하는 메일슬롯을 이용하는 방법이다. 메일슬롯은 실제로는 파일 시스템을 사용하고 있지만 리다이렉터를 통해 원격 메시지도 교환할 수 있는 방법이다. 네트워킹 구성 요소 없이 마치 파일을 다루듯이 손쉽게 그룹 메시지나 1대1 메시지를 교환할 수 있다. 사용하기가 쉬울뿐더러 메시지 전송률도 좋기 때문에 윈도우즈NT의 시스템 서비스에서 사용되고 있다[6].

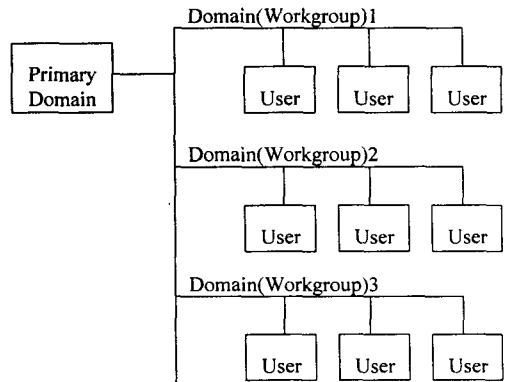
본 논문에서 구현하고자 하는 서버리스 응용 프로그램에서도 이 메일슬롯을 사용하였는데, 응용 프로그램에서 사용할 메일슬롯 프리미티브들은 다음과 같이 구성되었다.

- CreateGlobalSlot(domain) : "domain"에 그룹 메시지를 보내는데 사용될 메일슬롯을 생성한다.
- CreateLocalSlot : 그룹 메시지나 1대1 메시지를 읽어올 메일슬롯을 생성한다.
- BroadcastMsg(domain, msg) : "domain" 그룹에 그룹 메시지 "msg"를 전송한다.
- GetSlotInfo : 시스템의 지역 메일슬롯에 저장된 메시지에 대한 정보를 얻는다.
- ReadBroadcastMsg(buf) : 시스템의 지역 메일슬롯에 저장된 메시지를 "buf" 메모리 영역으로 읽어온다.
- SendMsg(domain, computer, msg) : "domain"그룹의 "computer" 사용자에게 "msg" 메시지를 전송한다.
- DestroyMsg(slot) : 사용이 끝난 그룹 메일슬롯, 혹은 지역 메일슬롯을 시스템에서 제거한다.

3. 서버리스 응용 프로그램의 구현

앞서 설명한 그룹통신 라이브러리를 이용하여 간단한 문서 관리 응용 프로그램을 구현해 보았다.

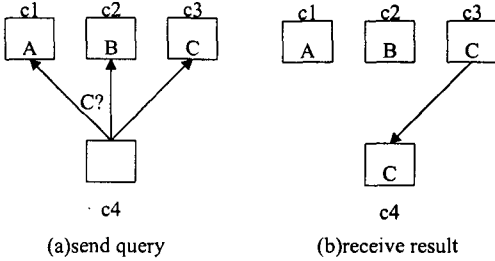
응용 프로그램에 참여하는 클라이언트들은 도메인(혹은 작업그룹)과 사용자들로 구성된 윈도우즈의 논리적 네트워크 계층을 기반으로 구성된다[그림1]. 응용 프로그램의 필요 작업에 따라 그룹 메시지와 1대1 메시지를 선택하여 사용한다. 예를 들어, 문서를 검색할 때에는 검색 조건을 작업 그룹 내에서 그룹 메시지를 이용하여 전송하고, 검색 결과는 해당 정보를 소유한 클라이언트와 검색 조건을 전송한 클라이언트 사이의 1대1 메시지를 사용하여 전송한다[그림2].



[그림 1]윈도우즈 네트워크의 논리적 계층도

사용자의 인증 과정도 그룹 통신을 이용하게 된다. 사용자 식별자와 암호로 인증을 요구할 경우, 해당 사용자의 정보가 지역 데이터베이스에 저장되어 있다면 아무 문제없이 인증이

이루어지지만 다른 클라이언트에 등록된 사용자인 경우에는 그룹 통신을 이용한 별도의 인증 과정이 필요하게 된다. 이 경우에는 사용자 정보를 그룹 메시지를 통해 전송하면 각 클라이언트들은 사용자 정보를 검색하여 그 결과를 되돌려 줌으로써 인증이 이루어진다.



[그림 2] 검색 과정

응용 프로그램에서 사용하는 메시지는 메시지의 송/수신자의 식별자와 도메인 이름, 메시지 식별자, 메시지 내용 등이 포함된다. 메시지 식별자는 일련의 번호를 갖게 되며, 네트워크 장애 등의 원인으로 인해 분실된 메시지가 발생한 경우 재전송을 요구할 수 있다.

응용 프로그램에서 사용하는 정보들은 가용성을 높이기 위해 중복시켜 관리할 수 있는데, 응용 프로그램의 개발자가 어느 정도로 중복할 것인지 선택할 수 있다. 중복 정도가 클수록 응용 프로그램의 성능이 그만큼 저하되는데, 문서 자료의 성격, 네트워크의 구성 등의 요소들에 근거하여 적당한 값을 선택할 수 있도록 한다. 이렇게 중복된 자료들은 작업 부하를 분산시키는 데에도 이용할 수 있다. 예를 들어, 응용 프로그램을 사용하는 도중에 한 클라이언트에 다른 클라이언트들의 요청이 쇄도하는 경우, 해당 작업 그룹의 중복 자료들이 있는 클라이언트로 요청을 재전송할 수 있다.

문서 관리 응용 프로그램의 경우, 지역성이 강하게 나타나는 특성이 있다. 이는 특정 사용자가 등록, 편집, 삭제한 문서들의 정보는 한 특정 클라이언트에 집중될 확률이 높다는 것이다. 예를 들어, 문서자료를 검색하는 경우, 대상 자료가 클라이언트들 사이에 분산되어 있는 것보다 한 곳에 집중되어 있다면 검색 결과를 전송하는데 필요한 1대1 메시지의 개수를 줄일 수 있다. 이렇듯 1대1 메시지의 개수를 줄이면 네트워크의 대역폭을 훨씬 효율적으로 사용할 수 있는 것이다.

4. 결론 및 향후 연구 방향

기존의 클라이언트/서버 구조의 응용 프로그램이 갖고 있는 문제점을 해결하기 위해 본 논문에서는 윈도우즈 플랫폼 하에서 그룹 통신을 할 수 있는 프리미티브들을 구현하고, 이 프리미티브들을 사용하여 서버가 없는 분산 응용 프로그램의 구조를 제안하고 구현해 보았다.

서버가 없이 클라이언트들끼리 그룹 통신을 이용하여 분산

응용 프로그램에 참여하면 다음과 같은 이점을 얻을 수 있다.

· 현재 행하고 있는 작업에 따라 클라이언트와 서버의 관계가 동적으로 성립된다. : 각 클라이언트는 때에 따라 서버의 역할도 겸하게 되고, 그 확률이 작업 그룹 내에서 비교적 고르게 분포되어 있으므로 작업이 어느 한 클라이언트에 집중되는 경우는 그리 많지 않다. 또한 자료 중복을 이용하여 집중되는 작업 부하를 중복 자료를 이용하여 분산시킬 수도 있다.

· 클라이언트의 수가 많아져도 시스템 전체의 성능이 저하되지 않는다. : 전체 응용 프로그램은 작업 그룹을 단위로 구성되고 실행되므로, 클라이언트의 수가 많아지면 그룹 구성을 새롭게 하는 것만으로도 전체 시스템의 성능을 조율할 수 있다.

· 네트워크 자원의 효율성을 높인다. : 네트워크에서 지원하는 그룹통신을 최대한 이용하고, 1대1 통신보다는 그룹 통신을 많이 사용하여 네트워크의 대역폭을 효율적으로 사용한다.

5. 참고 문헌

- [1] Robert Orfali, Dan Harkey, Jeri Edwards, "The Essential Distributed Objects Survival Guide", John Wiley & Sons, 1995
- [2] Kaashoek, M.F., and Tanenbaum, A.S. : "Efficient Reliable Group Communication for Distributed Systems" (submitted for publication, 1994)
- [3] George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems : Concepts and Design", Second Edition, Addison-Wesley, 1994
- [4] Chi-Chung Hui, Samuel T. Chanson : Efficient Load Balancing in Interconnected LANs Using Group Communication. ICDCS 1997 : 0-
- [5] Kaashoek, M.F., Tanenbaum, A.S., and Verstoep, K. : "Using Group Communication to Implement a Fault-Tolerant Directory Service," Proc. Thirteenth Int'l Conf. on Distributed Computing Systems, IEEE, pp. 130-139, 1993.
- [6] Microsoft, "MSDN On-Line Library : Platform SDK", Microsoft, <http://msdn.microsoft.com/isapi/>