

분산 구조에 의한 세션 기반 인스턴트 메시지 시스템

장춘서* · 이기수

A Session-based Instant Message System by Distributed Architecture

Choonseo Jang* · Ki-Soo Lee

Department of Computer Engineering, Kumoh Inst. of Tech. Gumi 730-701, Korea

요 약

본 논문에서는 SIP 세션 기반의 인스턴트 메시지(Instant Message) 시스템에서 확장성을 높이기 위하여 복수개의 서버를 사용하여 시스템의 부하를 효율적으로 분산 처리하는 새로운 구조를 제안하였다. 세션 기반 인스턴트 메시지 서버는 SIP 세션을 관리하여 멀티미디어 데이터를 포함한 인스턴트 메시지를 각 사용자들에게 전달해야 하고 각 사용자들에서 발생하는 다량의 프레즌스 정보 처리도 해야 함으로 사용자 수의 증가에 따라 부하를 적절히 분산하여야 한다. 이를 위하여 본 논문에서는 사용자 수의 증가에 따라 특정 서버의 부하가 커지면 전체 인스턴트 메시지 서버 중 가장 부하가 적은 서버로 사용자 요청을 분산하거나 새로운 인스턴트 메시지 서버를 생성하여 메시지 처리를 할당함으로써 동적으로 부하를 분산하도록 한다. 이와 같은 동작을 위하여 인스턴트 메시지 서버의 부하를 제어하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 형식이 설계되었고 인스턴트 메시지 서버들 사이 및 서버와 사용자 사이의 SIP 메시지 교환 절차가 제시되었다. 제안된 시스템의 성능은 시뮬레이션 실험을 통하여 분석하고 성능이 향상됨을 보였다.

ABSTRACT

In this paper we have proposed a new distributed architecture of session-based instant message system, in which multiple servers can distribute loads effectively to improve system extendability. The server of session-based instant message system should manage SIP sessions to transfer instant message with multimedia data to each users, and it should process large amount of presence informations from each users. Therefore load of the server should be distributed as the number of users increases. In our system, when server load increases according to large user numbers, the load are distributed by selecting a server with the lowest load value, or new server is created to allocate the message processing. Therefore the overall load of instant message system can be dynamically distributed. A new presence information data format with added elements for controlling server load has been designed for this purpose, and SIP messages exchanging procedure between servers and users has been also suggested. The performance of our proposed system has been analysed by simulation, and showed improvement of performance.

키워드 : 인스턴트 메시지, 프레즌스 서비스, SIP, 정보데이터형식

Key word : Instant Message, Presence Service, SIP, Information Data Format

접수일자 : 2014. 06. 03 심사완료일자 : 2014. 06. 20 게재확정일자 : 2014. 07. 04

* **Corresponding Author** Choonseo Jang(E-mail:csjang@kumoh.ac.kr, Tel:+82-54-478-7521)

Department of Computer Engineering, Kumoh Inst. of Tech., Gumi 730-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2014.18.9.2169>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

기본적인 인스턴트 메시지(Instant Message) 시스템 [1-3]은 페이지 단위로 동작하는 SIP(Session Initiation Protocol) MESSAGE 메서드를 사용한다. 그러나 이 방식은 다양한 멀티미디어 데이터 전달과 텍스트 기반의 컨퍼런스 동작에는 적합하지 않다. 반면에 SIP 다이얼로그를 사용해 사용자 사이에 세션 연결을 하여 동작하는 세션 기반 인스턴트 메시지 시스템에서는 이러한 문제를 해결 할 수 있으나 사용자 수의 증가에 따라 인스턴트 메시지 서버의 부하가 페이지 방식에 비해 빠르게 커지게 되어 확장성에 제한을 받게 된다.

인스턴트 메시지 시스템은 SIP 프레즌스(presence) 서비스[4-6]를 통하여 사용자의 온라인 상태나 접속 주소 등 동작에 필수적인 사용자 관련 정보를 얻는다. 이를 위하여 사용자는 SIP SUBSCRIBE 메시지를 사용해 프레즌스 서비스 등록을 하며 이후 상대방의 프레즌스 정보 변화를 NOTIFY 메시지를 통해 전달 받을 수 있게 된다.

세션 기반 인스턴트 메시지 시스템에서의 서버는 사용자 사이의 SIP 세션 연결 및 유지와 인스턴트 메시지의 교환 및 다량의 프레즌스 자원의 주소를 담은 요청 메시지를 처리해야하고 등록된 프레즌스 자원에서 발생하는 다량의 통지 메시지도 처리해야 하므로 사용자 수가 증가 할수록 서버의 부하는 급격히 증가하게 된다.

본 논문에서는 세션 기반 인스턴트 메시지 시스템의 부하를 복수개의 서버로 효율적으로 분산하여 인스턴트 메시지 시스템의 확장성을 높일 수 있는 방안을 연구하였다. 이를 위하여 각 인스턴트 메시지 서버들은 본 논문에서 새롭게 제안된 확장된 프레즌스 정보 데이터 구조를 사용하여 현재 부하 상태를 주기적으로 교환하고 특정 인스턴트 메시지 서버의 부하가 지정된 한도 이상으로 증가하는 경우 새로운 서버를 동적으로 생성하거나 부하가 적은 서버들이 이를 분산 처리하도록 하였다. 이를 위하여 인스턴트 메시지 서버의 부하를 제어하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 형식과 각 서버들 사이의 SIP 메시지 교환 절차가 제시되었다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 인스턴트 메시지 시스템의 구성 및 기존 연구에 대해 설명하고 III장에서는 본 논문에서 새롭게 제안하

는 분산 인스턴트 메시지 서버의 설계 및 구조를 설명하고 인스턴트 메시지 서버의 부하를 제어하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 형식 설계에 대해 설명한다. 다음 IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

2.1. 세션 기반 인스턴트 메시지 시스템

그림 1은 세션 기반 인스턴트 메시지 시스템의 기본 동작이다. 사용자는 먼저 INVITE 메서드를 사용하여 메시지 서버와 SIP 세션을 연결한다. INVITE메서드의 몸체 부분은 SDP(Session Description Protocol)[7] 타입 데이터이며 교환 할 인스턴트 메시지의 MIME 타입과 TCP 포트번호, 목적지 SIP 주소 등 이 포함된다. 메시지 서버는 응답 메시지의 몸체 부분에 인스턴트 메시지 교환에 사용할 SDP 정보를 담아 보내준다. 이와 같은 과정을 거쳐서 SIP 세션이 구성되고 다양한 MIME 타입 메시지 교환이 이루어진다.

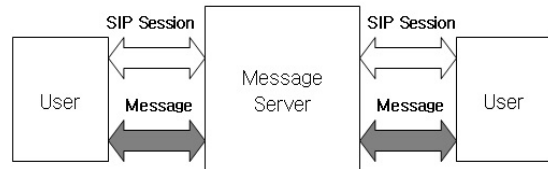


그림 1. 세션 기반 인스턴트 메시지 시스템
Fig. 1 Session-based Instant Message System

2.2. 프레즌스 서비스 동작

인스턴트 메시지 시스템에서는 SIP 프레즌스(presence) 서비스를 통하여 사용자의 온라인 상태나 현재 위치, 사용 가능한 통신 서비스의 종류 등 사용자 관련 프레즌스 정보를 얻는다. 그림 2에 프레즌스 서비스의 동작을 보였다. 여기서 PUA(Presence User Agents)는 프레즌스 정보를 담은 SIP PUBLISH 메시지[8]를 생성하고 이를 전송하는 기능을 한다. PA(Presence Agent)는 SIP 프로시 기능을 가진 프레즌스 서버이며 각각의 PUA로부터 전송된 사용자 프레즌스 정보를 받아 처리한다. 프레즌스 사용자(watcher)는 처리된 내용의 알림 대상이며 프레즌스 정보를 실제로 수신한다.

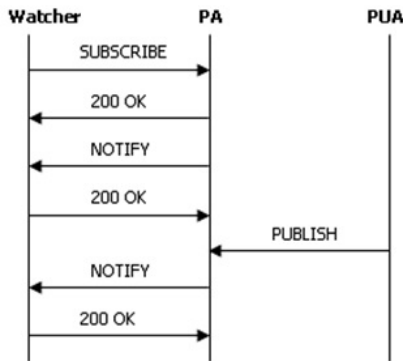


그림 2. 프레즌스 서비스 동작
Fig. 2 Operation of Presence Service

프레즌스 정보의 기본 형식은 PIDF(Presence Information Data Format)[9,10]이다. 프레즌스 정보 형식에서는 최상위 요소로 <presence>가 있고 하위 요소로 <tuple>이 각각의 프레즌스 정보를 구분하기 위하여 사용된다. 각 <tuple>은 필수 하위 요소로 <status>를 가지며 이는 해당 사용자의 현재 상태 정보를 가지고 있고 이외에 여러 추가적인 하위요소를 가질 수 있다.

세션 기반 인스턴트 메시지 서버는 MIME 타입 인스턴트 메시지의 처리 외에도 각 사용자들과의 SIP 세션 연결 및 세션 유지에 필요한 리프레시 처리를 하여야 한다. 아울러 다량의 프레즌스 자원의 등록 및 등록 유지를 위한 리프레시 처리, SIP PUBLISH 동작 처리와 각 사용자들에 대한 통지 메시지 처리가 필요하므로 사용자 수의 증가에 따라 인스턴트 메시지 서버의 부하는 크게 증가하여 시스템 확정성에 제약을 주게 된다. 그러나 세션 기반 인스턴트 메시지 시스템에서 복수개의 서버를 사용하여 확장성을 높이는 연구 결과는 아직 발표되고 있지 않다.

본 논문에서 제안한 분산 구조에 의한 세션 기반 인스턴트 메시지 시스템은 이와 같은 문제점을 해결하기 위하여 사용자 수의 증가에 따라 특정 인스턴트 메시지 서버의 부하가 지정된 한도 이상으로 증가하는 경우 새로운 서버를 동적으로 생성하거나 부하가 적은 서버들이 이를 분산 처리하도록 하는 새로운 방식을 제안하였다. 이를 위하여 프레즌스 정보의 기본 형식을 확장한 새로운 프레즌스 정보 형식을 설계하였다.

III. 시스템 설계 및 구현

3.1. 인스턴트 메시지 서버 구조 설계

본 논문에서 설계한 새로운 세션 기반 인스턴트 메시지 서버의 구조를 그림 3에 보였다.

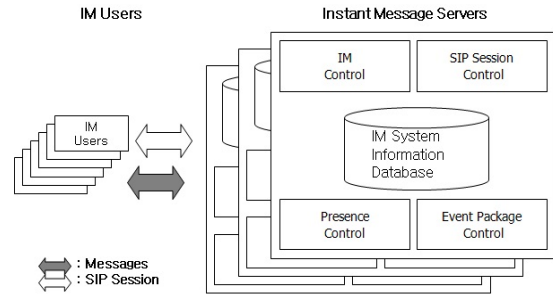


그림 3. 세션 기반 인스턴트 메시지 서버 구조
Fig. 3 Architecture of Session-based Instant Message Server

이 그림에서 각 서버는 시스템 정보 데이터베이스를 중심으로 하여 SIP 세션 제어 모듈, 인스턴트 메시지 제어 모듈, 프레즌스 제어 모듈 및 이벤트 패키지 제어 모듈로 구성된다. 시스템 정보 데이터베이스는 인스턴트 메시지 시스템 동작에 필수적인 각 사용자들의 프레즌스 정보 및 각 서버들의 부하 상태 등 전체 시스템 정보를 저장하고 처리한다.

SIP 세션 제어 모듈은 사용자들 사이에 인스턴트 메시지가 세션 기반으로 동작 할 수 있도록 한다. 이를 위하여 SIP INVITE 메시지를 사용하여 세션을 설정하고 이때 SDP 프로토콜을 사용해 메시지의 목적지 주소, 메시지의 MIME 타입 등을 서로 교환하며 세션 유지를 위한 리프레시 동작도 처리한다. SIP 세션 제어 모듈을 통하여 각 사용자 간의 세션 관리가 통합적으로 이루어져 텍스트 기반의 컨퍼런스 동작도 가능하게 된다.

인스턴트 메시지 제어 모듈은 시스템 정보 데이터베이스를 통하여 메시지가 전달될 상대방 관련 프레즌스 정보를 얻어 사용자들 간의 다양한 멀티미디어 메시지가 실시간으로 서로 교환 될 수 있도록 처리한다. 이벤트 패키지 제어 모듈은 서버들 및 서버와 사용자 사이에서 프레즌스 정보 데이터에 대한 변동 사항을 알려주고 교환하는 기능을 한다.

프레즌스 제어 모듈은 본 논문에서 새롭게 구현한 인스턴트 메시지 서버의 부하를 복수개의 서버로 효율적으로 분산하는데 필요한 기능이 추가된 프레즌스 정보 데이터 형식을 처리한다. 각 서버들은 이와 같이 확장된 프레즌스 정보 형식을 사용하여 현재 부하 상태를 주기적으로 교환하고 시스템의 부하가 커지면 각 서버의 부하를 효율적으로 분산하거나 새로운 서버를 동적으로 생성할 수 있다.

서버의 프레즌스 제어 모듈에서는 확장된 프레즌스 정보 형식 중 각 서버의 현재 부하 상태를 나타내는 요소인 <current-svr-load-index>의 값을 사용하여 주어진 기준값을 초과하는 경우 현재 부하 상태가 가장 낮은 서버를 시스템 정보 데이터베이스에서 찾아 이 서버에게 사용자의 처리를 분산하도록 한다. 모든 서버의 부하 상태가 기준값을 초과하는 경우에는 시스템 정보 데이터베이스의 <candidate-svr-list> 항목에서 새롭게 추가 가능한 서버를 선택하여 이 서버와 SIP 세션을 연결해 시스템의 부하를 분담하도록 한다. 이와 같은 동작 과정에 대한 순서도를 그림 4에 보였다.

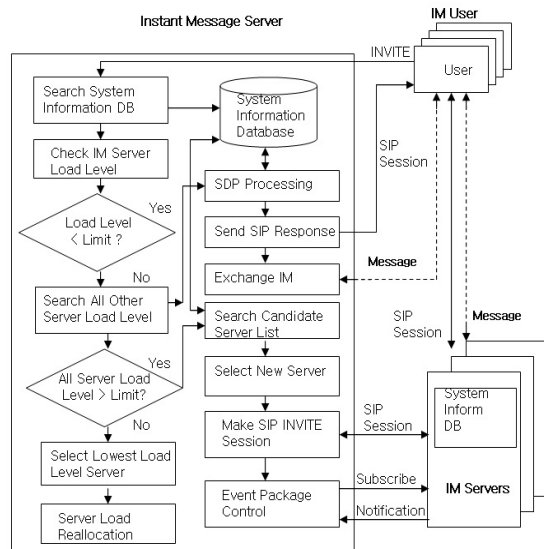


그림 4. 서버 동작 과정에 대한 순서도
Fig. 4 Flowchart of Server Operation Procedure

여기서는 사용자로부터 SIP INVITE 요청이 들어오면 시스템 정보 데이터베이스에서 서버의 부하 인덱스 값을 조사하여 허용된 값 이하이면 해당 사용자에게

SDP 파라미터를 적용한 응답 메시지를 보내어 SIP 세션 설정을 한다. 현 서버의 부하 인덱스 값이 허용된 값을 초과하면 다른 서버들의 부하 상태를 조사하여 가장 부하가 낮은 서버를 선택하여 해당 사용자의 요청을 처리하도록 한다. 모든 서버의 부하 인덱스 값이 기준치 이상이면 시스템 정보 데이터베이스의 서버 후보군을 나타내는 <candidate-svr-list> 항목에서 다른 서버를 선택하여 이 서버와 SIP 세션 연결을 맺는다. 다음 이 서버와 SIP SUBSCRIBE 및 NOTIFY 메시드에 의한 프레즌스 이벤트 패키지를 사용하여 시스템 정보 데이터베이스 내용을 전송 후 이 서버에게 사용자 처리를 분산하게 한다.

3.2. 프레즌스 정보 데이터 형식 설계

본 논문에 설계된 프레즌스 정보 데이터 형식에서는 복수개의 서버에 의한 부하의 분산 처리가 가능하도록 다음과 같은 요소들이 추가되었다. 먼저 최상위 요소인 <presence>의 하위 요소로 서버의 부하 상태를 나타내는 <current-svr-load-index>가 설계되었다. 이 요소는 각 서버에서 세션 설정과 인스턴트 메시지 처리 및 프레즌스 정보 처리에 소요되는 부하의 레벨을 나타낸다. 이 값은 서버와 사용자 사이에 SIP 세션을 설정하고 유지하는데 필요한 SIP 메시지 양과 사용자에게서 발생하는 초당 인스턴트 메시지 양 및 프레즌스 자원에서 발생하는 초당 SIP 메시지 양을 기준으로 하여 계산된다. 이 요소는 각 서버를 구분하기 위한 'im-svr-label'과 최대 부하값을 나타내는 'max-svr-load-index' 및 현재 동작 중인 서버의 개수를 나타내는 'current-active-svrs'를 속성으로 가진다.

다음으로 <candidate-svr-list> 요소가 설계되어 시스템의 부하가 증가 할 때 새롭게 추가 될 수 있는 서버들의 SIP 주소를 가지고 있도록 하였으며 각 서버가 현재 담당하고 있는 사용자 리스트인 <im-svr-users-list>도 추가되었다. 이 요소는 각 서버를 구분하기 위하여 'im-svr-identity'를 속성으로 가진다.

이외에도 현재 인스턴트 메시지 시스템의 상태를 표시하기 위한 요소인 <im-system-state>가 추가 되었고 이의 하위 요소로써 현재 전체 사용자 수를 나타내는 <im-total-users-count>와 개별 사용자들에 대한 정보를 가지고 있는 <im-users-info>가 각각 추가되었다. 본 논문에서 구현한 프레즌스 제어 모듈은 이와 같은 필요

요소들이 새롭게 추가된 프레즌스 정보 데이터 형식을 사용하여 동작한다. 그림 5에 서버 및 사용자 사이의 SIP 메시지 교환 절차를 보였다.

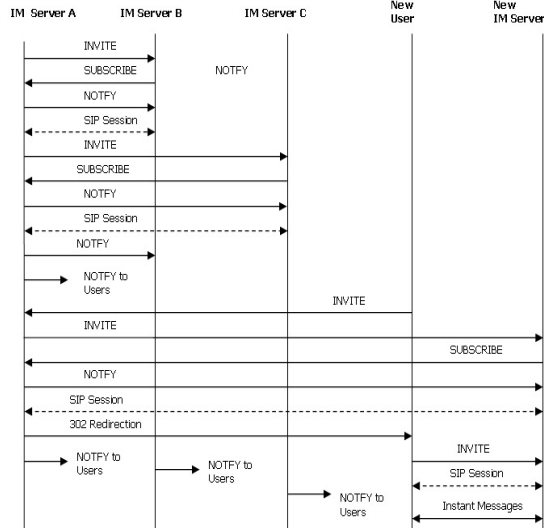


그림 5. 서버 및 사용자 사이의 SIP 메시지 교환 절차
Fig. 5 SIP Message Exchange Procedure between Servers and Users

여기에서 인스턴트 메시지 서버들은 주 서버(Server A)와 SIP INVITE 메서드를 사용해 SIP 세션을 설정하고 SIP SUBSCRIBE와 NOTIFY 메서드를 사용해 프레즌스 이벤트 패키지를 등록하고 프레즌스 정보를 교환한다. 변경된 프레즌스 정보는 전체 사용자에게 통지된다. 새로운 사용자가 서버에게 SIP INVITE 메서드를 사용해 세션 연결 요청을 하면 해당 서버가 시스템의 부하 상태를 조사하여 서버가 추가로 필요하다고 판단하면 시스템 정보 데이터베이스에서 새로운 서버의 SIP 주소를 얻어 이 서버에게 INVITE 요청 메시지를 보낸다.

다음 새로운 서버와 SIP 세션을 연결하고 이 서버는 SIP SUBSCRIBE와 NOTIFY 메서드를 사용해 프레즌스 이벤트 패키지를 등록하고 현 시스템의 프레즌스 정보를 얻는다. 다음 이 사용자에게 SIP Redirection 메서드를 사용해 새로운 인스턴트 메시지 서버와 연결하도록 알려준다, 해당 사용자는 이 새로운 서버와 세션 연결을 하고 인스턴트 메시지 전송을 시작한다.

IV. 성능 분석

본 논문에서 제안한 분산 구조에 의한 세션 기반 인스턴트 메시지 시스템의 성능을 시뮬레이션으로 분석하였다. 이를 위하여 분산 구조에 의한 인스턴트 메시지 시스템과 중앙 집중 방식에 의한 시스템과의 처리 시간을 비교 측정하였다. 먼저 인스턴트 메시지 전송을 위한 세션 연결에 걸리는 전체 시간을 사용자 수를 증가시켜 가면서 측정하였다. 그림 6에 이의 결과를 보였다.

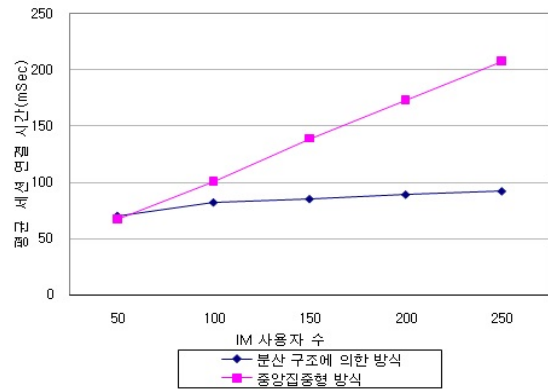


그림 6. 평균 세션 연결 시간
Fig. 6 Average Session Connection Time

여기서 분산 구조에 의한 방식을 사용하는 경우 서버의 최대 부하 허용값을 나타내는 속성인 ‘max-svr-load-index’가 100이 되도록 설정하였다. 이 경우 처음에 하나의 인스턴트 메시지 서버로 동작하다가 최대 부하 허용값을 넘으면 새로운 서버가 추가된다. 사용자 수가 50명일 때는 양쪽 모두 1개의 서버만을 사용하므로 동일한 성능을 보이나 이를 초과하면 제안된 방식의 경우 서버가 추가되어 부하를 분담하게 되므로 큰 성능의 향상을 보여준다. 사용자 수가 100명 일 때는 평균 세션 연결 시간이 20.8% 감소하고, 150명 일 때는 38.8% 감소하고, 250명 일 때는 56.7%가 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 알 수 있다.

다음 전체 사용자에게 인스턴트 메시지를 전달하는데 소요되는 평균 지연시간을 비교 측정하였다. 그림 7에 이의 결과를 보였다.

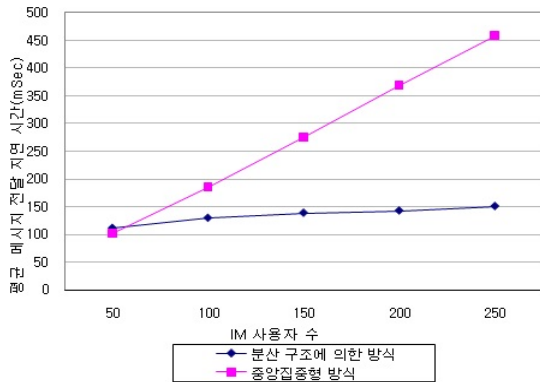


그림 7. 평균 메시지 전달 지연시간
Fig. 7 Average Message Transfer Delay Time

여기서 인스턴트 메시지는 1K 바이트 크기의 텍스트와 50K 바이트 크기의 미디어 데이터의 혼합 형태로 하였다. 처음에 하나의 인스턴트 메시지 서버로 동작하다가 사용자 수가 50명을 넘으면 새로운 서버가 추가되어 부하를 분담하게 된다. 측정 결과 사용자 수가 100명 일 때는 평균 메시지 전달 지연 시간이 29.7% 감소하고, 150명 일 때는 48.4% 감소하고, 250명 일 때는 67.1%가 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 알 수 있다.

그림 8에는 인스턴트 메시지 시스템 각 사용자들에게 프레즌스 정보를 보내는데 걸리는 전체 처리 시간을 사용자 수를 증가시켜 가면서 측정하였다. 이때 사용한 프레즌스 정보는 20K 바이트 크기의 PIDF 형식의 프레즌스 문서이다.

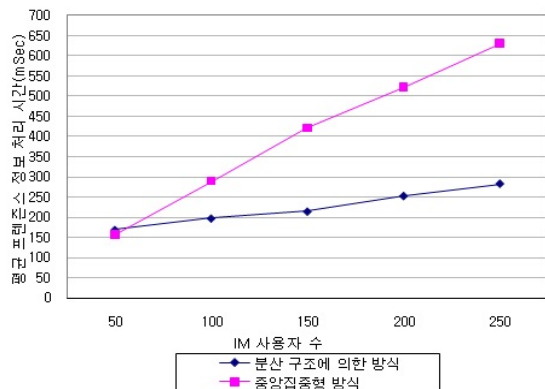


그림 8. 평균 프레즌스 정보 처리 시간
Fig. 8 Average Presence Information Processing Time

여기서는 사용자 수가 50명일 때는 제안된 방식에서도 1개의 서버만을 사용하므로 기존 방식과 거의 동일한 처리 시간을 보이지만 50명을 초과하면 본 논문에서 제안한 방식에 의해 서버가 추가되어 전체 처리 시간이 크게 단축됨을 보여준다. 사용자 수가 100명 일 때는 평균 프레즌스 정보 처리 시간이 31.8% 감소하고, 150명 일 때는 49.2% 감소하고, 250명 일 때는 55.3%가 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 알 수 있다.

V. 결론

본 논문에서는 SIP 세션 기반의 인스턴트 메시지 시스템에서 복수개의 서버를 사용하여 시스템의 부하를 효율적으로 분산 처리하여 시스템의 확장성을 높일 수 있는 새로운 구조를 제안하였다. 이를 위하여 각 인스턴트 메시지 서버들은 본 논문에서 새롭게 제안된 확장된 프레즌스 정보 데이터 구조를 사용하여 현재 부하 상태를 주기적으로 교환하고 특정 인스턴트 메시지 서버의 부하가 지정된 한도 이상으로 증가하는 경우 새로운 서버를 동적으로 생성하거나 부하가 적은 서버들이 이를 분산 처리하도록 하였다. 이와 같은 동작을 위하여 인스턴트 메시지 서버의 부하를 제어하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 형식이 설계되었고 인스턴트 메시지 서버와 사용자 사이의 SIP 메시지 교환 절차가 제시되었다. 제안된 시스템의 성능 분석을 위하여 인스턴트 메시지 전송을 위한 세션 연결에 걸리는 전체 시간을 사용자 수를 증가시켜 가면서 측정하였고 평균 세션 연결 시간이 20.8%에서 56.7%까지 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 보여 주었다. 다음 전체 사용자에게 인스턴트 메시지를 전달하는데 소요되는 평균 지연시간을 비교 측정하였고 역시 측정 결과로 29.7%에서 67.1%까지 감소하였다. 또 전체 사용자들에게 프레즌스 정보를 보내는데 걸리는 전체 처리 시간도 큰 폭의 개선을 보여주었다. 향후 과제로는 대역폭이 상대적으로 작은 무선 모바일 환경에서 제안된 시스템의 실제 성능을 측정하고 개선된 방식을 연구 할 필요가 있다.

감사의 글

본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

REFERENCES

- [1] P. Saint-Andre, " Extensible Messaging and Presence Protocol (XMPP):Instant Messaging and Presence," *RFC* 6121, March 2011.
- [2] J. Rosenberg, "IETF Specifications for Instant Messaging and Presence Using SIP," *RFC* 6914, April 2013.
- [3] P. Saint-Andre, A. Hourii, "Interworking between the SIP and the Extensible Messaging and Presence Protocol: Instant Messaging", *Internet-Draft*, March 2014.
- [4] P. Saint-Andre, "Instant Messaging and Presence Purpose for the Call-Info Header Field in the Session Initiation Protocol (SIP)," *RFC* 6993, Ju;y 2013.
- [5] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Address Format," *RFC* 6122, March 2011.
- [6] M. Lonnfors, E. Leppanen, H. Khartabil, J. Urpalainen, "SIP Extension for Partial Notification of Presence Information", *RFC* 6261, September 2011.
- [7] F. Andreasen, "Session Description Protocol (SDP) Capability Negotiation," *RFC* 5939, September 2010.
- [8] A. Niemi, M. Lonnfors, E. Leppanen, "Publication of Partial Presence Information", *RFC* 5264, September 2008.
- [9] H. Schulzrinne,H. Tschofenig and M. Thomson, "Dynamic Extensions to the Presence Information Data Formate," *RFC* 5962, September 2010.
- [10] J. Rosenberg, H. Schulzrinne and O. Levin, "Presence Information Data Format (PIDF) Extension for Partial Presence," *RFC* 5262, September 2008.



장춘서(Choonseo Jang)

1993년 8월 : 한국과학기술원 공학박사
현재 : 금오공과대학교 컴퓨터공학과 교수
※관심분야 : SIP, 임베디드 시스템, 인터넷텔레포니



이기수(Ki-Soo Lee)

1982년 2월 : 서울대학교 공학석사
현재 : 금오공과대학교 컴퓨터공학과 교수
※관심분야 : SIP, 데이터베이스