

다중서버 방식의 새로운 인스턴트메시지 서비스 구조

장춘서*, 이기수*

A New Architecture of Instant Message Service with Multiple Servers

Choonseo Jang*, Ky-Soo Lee*

요 약

인스턴트메시지(Instant Message) 서비스에서 서버는 인스턴트메시지와 사용자들에 대한 다량의 프레즌스 자원을 담은 요청 메시지를 처리해야하고 등록된 프레즌스 자원에서 발생하는 다량의 통지 메시지도 처리해야 한다. 본 논문에서는 사용자 수의 증가에 따라 서버의 부하를 효율적으로 분산하여 처리 할 수 있고 확장 호처리 언어를 적용하여 사용자에게 다양한 기능을 제공 할 수 있는 새로운 다중 서버 방식을 사용한 인스턴트메시지 서비스 구조를 제안하였다. 서비스 사용자는 프레즌스 정보 및 자신이 원하는 기능을 기술한 호처리 언어 스크립트를 서버에 등록하며, 서버는 인스턴트메시지의 처리 및 프레즌스 서비스와 호처리 언어 스크립트 실행 기능을 제공한다. 이를 위하여 확장된 프레즌스 정보 데이터 구조가 제안되었고 인스턴트메시지 서비스 처리에 필요한 호처리 언어 동작 태그들이 새롭게 추가되었다. 따라서 본 시스템에서는 시스템의 확장성이 높아지고 프레즌스 서비스와 호 처리를 조합한 다양한 서비스를 사용자에게 제공할 수 있게 된다. 또 서버에 인스턴트메시지 처리 모듈을 통합하여 시스템 전체의 SIP(Session Initiation Protocol) 메시지의 발생 양을 줄여 시스템의 효율을 더욱 높일 수 있도록 하였다. 제안된 시스템의 성능은 실험을 통하여 분석하였다.

▶ Keywords : 세션시작프로토콜, 인스턴트메시지, 호처리 언어

Abstract

In Instant message service, the server should process instant messages and request messages which include many presence resources for users. And it also preprocesses massive notification messages generating from the subscribed presence resources. In this paper a new architecture of instant message service with multiple servers which can distribute loads efficiently as the number of users increases has been suggested. It also provides various functions to users using extended

•제1저자: 장춘서, 교신저자: 이기수

•투고일 : 2013. 8. 10, 심사일 : 2013. 9. 6, 게재확정일 : 2013. 10. 17.

* 금오공과대학교 컴퓨터공학과(Dept. of Computer Engineering, Kumoh Inst. of Technology)

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

call processing language. The user subscribes presence information and call processing language script which describes user's functions. The server processes instant messages, presence services and call processing language scripts. New extended presence information data structure has been suggested and new call processing language operation tags have been added. Therefore extendability of the system can be increased and various services which combine presence service and call processing can be provided in this system. Furthermore instant message processing module has been integrated in the server to decrease the amount of SIP(Session Initiation Protocol) messages, and it also improves system efficiency. The performance of our proposed system has been analysed by experiments.

▶ Keywords : Session Initiation Protocol, Instant Message, Call Processing Language

I. 서 론

인스턴트메시지 서비스[1][2]는 SIP(Session Initiation Protocol) 프레즌스(presence) 서비스를 통하여 동작에 필수적인 사용자의 온라인 상태나 연결 주소 및 현재 위치 등 사용자 관련 정보를 얻는다[3]. 이때 사용자들의 프레즌스 정보를 처리하는 프레즌스 서버에 호처리 언어를 적용시키면 인스턴트메시지 서비스에서 다양한 사용자의 서비스 요구 사항을 제공 할 수 있다[4]. 호처리 언어는 VoIP 서비스에서 사용자 통화를 처리하는 방식을 기술하고 제어할 수 있는 XML 기반의 언어이다.

따라서 본 논문에서는 호처리 언어 기능을 확장하여 인스턴트메시지 서비스에서 다양한 사용자의 서비스 요구 사항을 프레즌스 서비스와 사용자 호 처리와의 조합으로 가능하도록 하였다. 인스턴트메시지 서비스에서의 서버는 다량의 프레즌스 자원의 주소를 담은 요청 메시지를 처리해야하고 등록된 프레즌스 자원에서 발생하는 다량의 통지 메시지도 처리해야하므로 사용자 수가 증가 할수록 서버의 부하는 급격히 증가하게 된다[5][6].

본 연구에서는 시스템의 부하가 커지면 각 서버의 부하를 효율적으로 분산하여 시스템의 확장성을 아울러 높일 수 있도록 하였다. 이를 위하여 각 서버들은 본 논문에서 새롭게 제안된 확장된 프레즌스 정보 데이터 구조를 사용하여 현재 부하 상태를 주기적으로 교환한다. 이 과정에서 특정 서버의 부하가 지정된 한도 이상으로 증가하는 경우 새로운 서버를 동적으로 생성하거나 부하가 적은 서버들이 이를 분산 처리하도록 하였다.

아울러 본 논문에서는 서버에 인스턴트메시지 처리 기능을 통합하여 인스턴트메시지 처리와 관련된 사용자 등록 및 통지 시에 발생하는 SIP 메시지의 발생 양을 감소시켜 시스템의 효율을 더욱 높일 수 있도록 하였다.

이와 같은 기능을 제공하기 위하여 본 연구에서는 서버의 부하를 제어하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 포맷과 확장 호처리 언어에 필요한 추가 태그들이 제시되었고 이들을 적용한 다중 서버 방식을 사용한 새로운 인스턴트메시지 서비스 구조가 설계되었다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 호처리 언어의 구성에 대해 설명하고 이어서 인스턴트메시지 서비스의 구성 및 기존 연구에 대해 설명한다. III장에서는 본 논문에서 새롭게 제안하는 확장 호처리 언어를 사용한 새로운 다중 서버의 설계 및 구조를 설명하고 이를 적용한 인스턴트메시지 서비스의 구현에 대해 설명한다. 다음 IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 호처리 언어(Call Processing Language)

호처리 언어는 XML 포맷으로 기술된 VoIP 서비스 제어를 위한 언어이며 사용자들이 다양한 VoIP 서비스를 기술 할 수 있도록 텍스트 기반 태그로 구성된다[4]. 호처리 언어 스크립트는 호 처리 동작(Call Processing Action) 부분과 스크립트에 대한 보조 정보 부분으로 구성되어 있다.

이중 사용자 통화 처리 동작 부분은 다시 상위 단계 동작과 하위 단계 동작으로 구분되며 이들은 서버가 실행 할 각종

동작들을 기술하고 있다. 상위 단계 동작은 텍스트 형태의 <incoming> 태그와 <outgoing> 태그를 사용해 서버로 들어 오는 동작과 나가는 동작을 기술한다. 또한 하위에 있는 위치 변경 동작, 스위치 동작, 신호 동작 및 비신호 동작들이 호 처리를 위한 조건과 취해야 할 행동들에 대한 세부적인 기능을 기술하고 있다.

그림 1에 호처리 언어 스크립트의 예를 보였다. 여기서는 <incoming> 태그를 사용한 예이다. 여기서는 스위치 동작을 나타내고 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<cpl xmlns="urn:ietf:params:xml:ns:cpl"
  xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd">
<incoming>
  <address-switch field="origin">
    <address is="sip:user1@test.net">
      <time-switch>
        <time dstart="20130311T090000"
          duration="PT03H" freq="weekly">
          <busy/>
        </time>
      </address-switch/>
    </address/>
  </incoming>
</cpl>
```

그림 1. 호처리 언어 스크립트의 예
Fig. 1. Example of Call Processing Language Scrip

2. 인스턴트메시지 서비스 및 기존 연구

그림 2는 인스턴트메시지 서비스의 기본 동작이다. 여기서는 메시지 서버를 통하여 텍스트 및 멀티미디어 메시지를 교환하며 SIP UA(User Agent)사이의 SIP 다이얼로그(dialog) 생성과정 없이 각 메시지는 독립적으로 처리된다[7]. 이 과정에서 사용자들에 관한 각종 프레즌스 정보[8]가 필요하며 이는 프레즌스 서버가 처리한다. 그림 3에 프레즌스 서비스의 동작을 보였다. 여기서 사용자는 SIP SUBSCRIBE 메시지의 헤더에 프레즌스 자원의 주소를 넣어 프레즌스 서버에게 보내며 프레즌스 서버는 이후 프레즌스 자원 목록에 지정된 사용자의 온라인 상태 등 각종 프레즌스 정보를 처리한다. 그림 3에서 등록된 프레즌스 자원을 나타내는 PUA(Presence User Agents)는 프레즌스 정보의 변화가 발생하면 SIP PUBLISH 동작[9]을 통하여 이를 프레즌스 서버에게 전송하고 프레즌스 서버는 해당 프레즌스 자원을 등록된 사용자에게 통지 메시지를 전송한다.

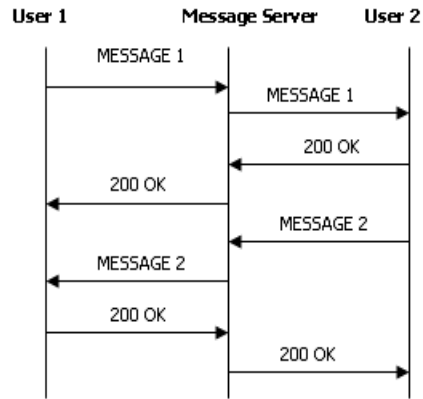


그림 2. 인스턴트메시지 서비스 동작
Fig. 2. Operation of Instant Message Service

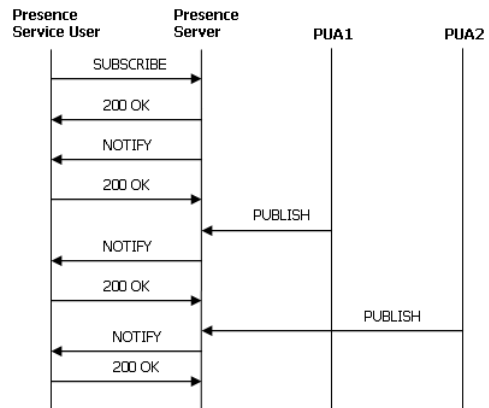


그림 3. 프레즌스 서비스 동작
Fig..3. Operation of Presence Service

인스턴트메시지 서비스에서 서버는 다량의 프레즌스 자원의 등록 및 리프레시에 대한 처리, SIP PUBLISH 동작 처리와 각 사용자들에 대한 통지 메시지 처리가 필요하므로 인스턴트메시지 서비스의 사용자 수의 증가에 따라 서버의 부하는 크게 증가한다.

기존의 인스턴트메시지 서비스 관련 연구는 서버의 부하 감소보다는 메시지 전송 과정에서의 효율을 높이는 데 치중하고 있으며 다중 서버 구조에 대한 연구가 충분하지 않다[10]. 본 논문에서는 인스턴트메시지 서비스의 사용자 수의 증가에 따라 특정 서버의 부하가 지정된 한도 이상으로 증가하는 경우 새로운 서버를 동적으로 생성하거나 부하가 적은 서버들이 이를 분산 처리하도록 하였고 이때 서버에 확장 호처리 언어 기능을 적용하여 프레즌스 서비스와 호 처리를 조합한 다양한 서비스가 가능한 새로운 방식을 제안 하였다.

III. 시스템 설계 및 구현

1. 인스턴트메시지 서비스 구조 설계

그림 4에 본 논문에서 설계한 확장 호처리 언어를 사용한 새로운 다중 서버 방식의 인스턴트메시지 서비스의 구조를 보였다. 여기에서 각 서버들은 인스턴트메시지 서비스와 프레즌스 서비스를 통합하여 제공하며 이를 위해 인스턴트메시지 처리 모듈, 호처리 언어 모듈, 부하 제어 모듈 및 SIP 메시지 처리 모듈로 구성된다. 또 이들 동작에 필요한 데이터베이스로 프레즌스 정보 데이터베이스와 호처리 언어 스크립트 데이터베이스를 가진다.

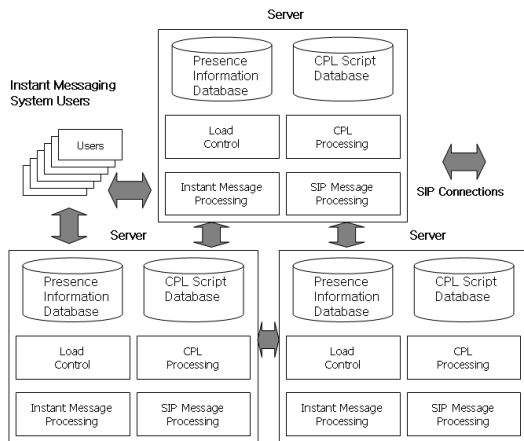


그림 4. 설계된 인스턴트메시지 서비스 구조
Fig. 4. A Designed Architecture of Instant Message Service

여기서 인스턴트메시지 처리 모듈은 사용자들 간의 텍스트 및 멀티미디어 메시지가 서로 교환 될 수 있도록 처리하며 상대방이 오프라인 상태인 경우는 전송된 메시지를 일시 저장하고 이 메시지들의 대기 상태 및 메시지 요약 정보를 프레즌스 정보 데이터베이스에 기록하는 기능을 한다. 기존 시스템에서는 메시지 서버와 프레즌스 서버를 분리하여 구성하므로 저장된 메시지의 상태 정보를 처리하기 위한 별도의 이벤트 패키지 과정이 추가로 필요하지만 본 구조에서는 하나의 서버에서 이 과정을 통합하여 처리하므로 전체적인 시스템의 동작 효율을 높일 수 있다.

프레즌스 서비스에 사용되는 프레즌스 정보 데이터 포맷도 본 연구에서 새롭게 구현한 부하 제어 기능이 추가된 구조이

므로 각 서버들은 현재 부하 상태를 주기적으로 교환하여 시스템의 부하가 커지면 각 서버의 부하를 효율적으로 분산하거나 새로운 서버를 동적으로 생성할 수 있다.

인스턴트메시지 서비스 사용자가 주 서버에게 SIP SUBSCRIBE 메시지를 보내면 이 서버는 프레즌스 정보 데이터베이스에서 현재 각 서버의 부하 상태를 나타내기 위하여 인스턴트메시지 서비스 사용자가 초당 발생하는 메시지 양과 프레즌스 자원에서 초당 발생하는 SIP 메시지 양의 전체 합을 계산하여 설계된 'sys-load-level' 요소의 값을 조사하여 주 서버가 기준값 이하이면 해당 사용자에게 대한 서비스 처리를 시작하고 기준값을 초과하는 경우엔 이 'sys-load-level'에서 각 서버를 구분하기위해 설계된 'sys-server-id' 속성값에 따라 가장 부하가 적은 서버를 선택한다.

다음 이 서버를 가리키는 SIP Redirection 메시지를 사용자에게 보내어 이 서버에게 요청을 하도록 한다. 'sys-load-level' 요소의 모든 서버의 부하 상태가 기준 값을 초과하는 경우 시스템의 부하에 따라 새롭게 추가 될 수 있는 서버들의 URI를 가지고 있는 'sys-uri-server' 요소 값을 데이터베이스에서 추출하여 이 목록에서 새로운 서버를 선택 후 이 서버를 인스턴트메시지 서비스에 참가시키기 위해 SIP INVITE 메시지를 보낸다. 이 서버는 주 서버로부터 전체 프레즌스 정보 데이터를 받아 사용자들에 대한 시스템 서비스 처리를 분담한다.

2. 확장 호처리 언어 및 프레즌스 정보 데이터 포맷 설계

본 논문에서는 프레즌스 서비스의 사용자 부가 기능을 위해 확장 호처리 언어를 적용하였다. 이를 위하여 호처리 언어 동작 처리 부분 중 상위 단계 동작에서 서버로 들어오는 동작과 나가는 동작을 제어하기 위한 <sys-incoming> 태그와 <sys-outgoing> 태그가 새롭게 추가되었다. 여기서 <sys-incoming> 태그는 인스턴트메시지 서비스에 등록된 해당 사용자에게 대한 SIP SUBSCRIBE 메시지가 서버에 수신되었을 때 실행되며 <sys-outgoing> 태그는 해당 사용자가 서버에 프레즌스 정보 등록 요청을 보냈을 때 실행된다. 이태그들은 하위 단계 동작을 위해 추가된 <sys_control_sip_subscription> 태그와 기존의 <time-switch>를 사용해 해당 사용자에게 대한 프레즌스 정보 등록 과정에서 특정 시간대를 지정해 프레즌스 정보 등록의 허용 및 금지를 제어하는데 사용된다.

이와 함께 SIP NOTIFY 메시지의 실행 제어를 위하여 해당 사용자에게 대한 프레즌스 정보 통지가 발생했을 때 동작을 제어하기 위한 <sys_incoming_notify_control> 태그와 해당 사용자

가 SIP PUBLISH 메시지로 프레즌스 정보를 발생했을 때 실행되는 <sys_outgoing_notify_control> 태그도 새롭게 추가되어 프레즌스 정보 통지 기능을 사용자가 설정 및 제어할 수 있도록 하였다. 이를 통하여 원하지 않는 시간에 발생하는 SIP 통지 메시지를 억제 할 수 있다. 확장 호처리 언어의 하위 단계 동작을 위해 추가된 <sys_control_sip_subscription> 태그는 SIP SUBSCRIBE 메시지에 대한 제어 처리를 담당한다. 또 호처리 언어 스위치 동작으로 <sys_location_addr-switch> 태그가 추가되어 인스턴트메시지 서비스에 등록된 사용자의 위치나 주소를 조건으로 하여 이에 따른 처리를 할 수 있도록 하였다. 아울러 <sys_call_outgoing> 태그를 추가하여 필요시 등록된 사용자들에 대한 직접적인 호를 발생시킬 수 있도록 하였다.

프레즌스 정보 데이터 포맷 설계에서는 각 서버가 현재 처리하고 있는 사용자수를 나타내기 위한 'sys-count-users' 요소가 추가되었고 각 서버가 현재 담당하고 있는 사용자 리스트인 'sys-list-users'가 추가되었다. 이 요소의 속성으로는 각 서버를 구분하기위한 'sys-identity-server'가 포함되었다.

서버 부하 상태를 파악하여 서버를 제어하기 위하여 'sys-load-level' 요소가 추가되었다. 이 요소는 인스턴트메시지 시스템의 각 서버에서 프레즌스 서비스 처리와 SIP 메시지 처리에 소요되는 부하의 상태를 수치로 표시한다. 'sys-load-level' 값은 인스턴트메시지 시스템의 등록된 각 사용자에게서 초당 발생하는 메시지 양과 등록된 프레즌스 자원에서 초당 발생하는 SIP 메시지 양의 전체 합을 계산하여 정해진다. 이 요소의 속성으로는 최대 부하 레벨을 나타내는 'max-sys-load-level'과 각 서버를 구분하기위한 'sys-server-id'가 포함되었다. 또 현재 인스턴트메시지 서비스의 상태를 표시하기위하여 'sys-messaging-status'가 새롭게 추가 되었다. 이 요소의 하위 요소로 인스턴트메시지 시스템 사용자들에 대한 개별 정보를 나타내기 위한 요소인 'sys-info-users'가 추가되었다.

다음으로 현재 인스턴트메시지 서비스에서 동작 중인 서버의 개수를 나타내는 'sys-count-server'와 시스템의 부하에 따라 새롭게 추가 될 수 있는 서버들의 URI를 가지고 있는 'sys-uri-server'가 추가되었다. 그림 5에 이와 같이 설계된 시스템 동작 과정에 대한 순서도를 보였다.

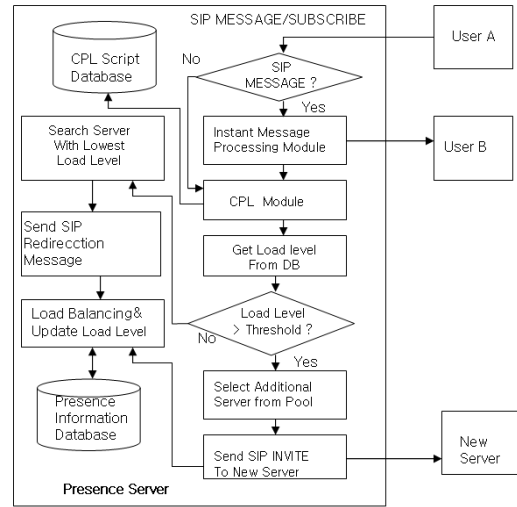


그림 5. 설계된 시스템 동작 과정에 대한 순서도
Fig. 5. Flowchart of Designed System Operation

여기서는 인스턴트메시지 서비스 사용자로부터의 요청이 발생 할 경우 SIP MESSAGE인지 프레즌스 정보 등록을 위한 SIP SUBSCRIBE인지를 판단하여 SIP MESSAGE인 경우 인스턴트메시지 처리 모듈에서 이를 처리한다. SIP SUBSCRIBE인 경우 호처리 언어 스크립트 데이터베이스를 액세스하여 사용자에게 대한 호처리 언어 스크립트를 조사하고 이에 대한 처리를 한다.

다음 'sys-load-level' 값을 조사하여 각 서버의 현재 부하 레벨을 확인 한 후 모든 서버의 부하 상태가 높아 새로운 서버의 추가가 필요한 경우 서버 추가 과정으로 들어간다. 이를 위하여 새롭게 추가 될 수 있는 서버들의 URI를 가지고 있는 'sys-uri-server' 값에서 서버를 선택 후 이 새로운 서버에게 SIP INVITE 메시지를 보내어 인스턴트메시지 서비스에 참가하도록 한다.

이 새로운 서버는 SUBSCRIBE 메시지를 주 서버에게 보내어 NOTIFY 메시지를 통해 시스템의 전체 프레즌스 정보 데이터와 호처리 언어 스크립트를 받게 되고 이 후 사용자들에 대한 시스템 서비스 처리를 분담한다. 새로운 서버의 추가를 필요로 하지 않는 경우 서버의 현재 부하 레벨이 가장 낮은 서버를 선택하여 SIP Redirection 메시지를 사용하여 이 서버에게 사용자 처리를 분담시킨다.

IV. 성능 분석

본 논문에서 제안한 인스턴트메시지 서비스의 성능 분석을

위하여 두 대의 PC를 시스템 서버로 사용하였고 일반 사용자 용으로는 20대의 PC가 사용되었다. 서버의 운영체제로는 커널 버전 2.6인 리눅스를 사용하였다. 서버 및 일반 사용자용 PC는 하나의 LAN 세그먼트로 구성하였고 네트워크 인터페이스 속도는 100Mbps이다.

본 논문에서 제안한 다중 서버 방식의 인스턴트메시지 서비스와 기존 방식과의 처리시간을 비교 측정하였다. 이를 위하여 인스턴트메시지 서비스 각 사용자들이 온라인 상태가 되었을 때 이들에게 프레즌스 정보를 보내는데 걸리는 시스템의 전체 처리 시간을 사용자 수를 증가시켜 가면서 측정하였다. 이때 사용한 프레즌스 정보는 30K바이트 크기의 PDF 포맷의 프레즌스 문서이며 각 사용자 당 70개씩의 프레즌스 문서를 반복하여 보내도록 하였다. 이와 같이 프레즌스 정보 크기와 각 사용자 당 반복해서 보내는 프레즌스 문서 양을 크게 하여 시스템 전체 사용자 수를 늘린 시뮬레이션 효과를 내었다. 또 인스턴트메시지 서비스 사용자 수 4명까지는 하나의 서버를 사용하고 이를 초과 할 때부터 서버가 추가되도록 하였다.

그림 6에 이의 결과를 보였다. 여기서는 사용자 수가 4명 일 때는 제안된 방식에서도 1개의 서버만을 사용하므로 기존 방식과 같은 처리 시간을 보이지만 8명 일 때부터는 본 논문에서 제안한 방식에 의해 서버가 추가되어 시스템의 전체 처리 시간이 크게 단축됨을 보여준다. 사용자 수가 8명 일 때는 제안된 시스템에서의 처리 시간이 35.5% 감소하고 참가자 수가 12명 일 때는 39.8%, 참가자 수가 16명 일 때는 38.9%, 참가자 수가 20명 일 때는 43.4% 각각 감소한다. 이와 같이 본 논문에서 제안된 방식의 처리 시간 개선 효과는 시스템 사용자가 수가 증가할수록 높아짐을 보여준다.

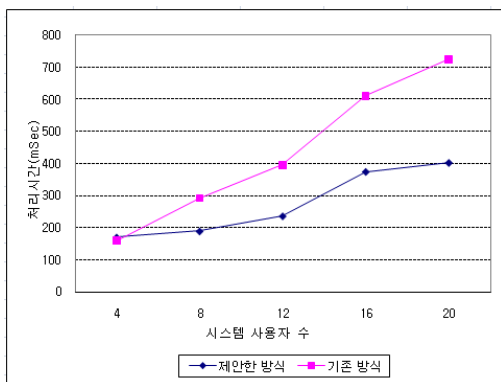


그림 6 통지 메시지 처리 시간 비교
Fig. 6. Comparison of Notification Messages Processing Time

다음 서버에서 SUBSCRIBE 메시지 처리 시간을 비교하기 위하여 각 사용자당 100개씩의 프레즌스 자원을 등록하도록 하였다. 그림 7에 이의 결과를 보였다.

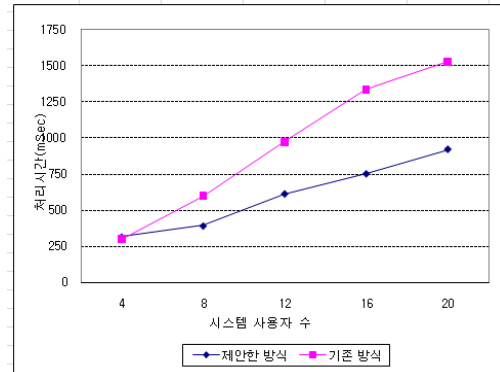


그림 7. SUBSCRIBE 메시지 서버 처리 시간 비교(최대 부하 허용값=200)

Fig. 7. Comparison of Server Processing Time for SUBSCRIBE Message(Max. Load Allowance=200)

여기서는 서버의 최대 부하 허용값을 200으로 하여 현재 시스템 부하 상태를 나타내는 'sys-load-level'이 이 값을 초과하면 서버가 추가되도록 하였다. 본 논문에서 구현한 인스턴트메시지 시스템에서 물리적으로 작은 수인 20개 PC에서 실제 사용자 수가 많은 환경과 유사하도록 각 사용자당 100개씩의 프레즌스 자원을 등록하는 경우 1개 서버만으로 이를 처리 할 때 사용자 수가 4명인 경우 'sys-load-level' 값은 190으로 측정되었다. 따라서 서버의 최대 부하 허용값을 200으로 할 경우 사용자 수 4명에서는 한 개의 서버만을 사용하므로 제안된 방식에서도 기존 방식과 같은 처리 시간을 보이지만 사용자 수가 이를 초과하면 'sys-load-level' 값이 서버의 최대 부하 허용값 200을 넘어서므로 서버가 새롭게 추가되어 부하를 분담하여 큰 성능의 향상을 보여준다. 사용자 수가 8명 일 때는 처리 시간이 34.7% 감소하고 12명 일 때는 36.9%, 16명 일 때는 43.7%, 20명 일 때는 39.8%가 각각 감소함을 보여주고 있다.

그림 8은 서버의 최대 부하 허용값을 400으로 하여 측정 한 결과이다. 이 경우는 사용자 수가 8명 까지는 한 개의 서버만으로 동작하며 이를 초과할 때부터 서버가 추가된 경우이다. 역시 이 경우에도 성능의 향상을 보여주고 있다. 따라서 본 연구에서 제시된 시스템 구조에서는 시스템 관리자가 서버의 최대 부하 허용값을 시스템 동작 상황에 따라 설정하여 동작 가능한 서버 수를 조정 할 수 있어 시스템의 확장성을 높

일 수 있는 것을 알 수 있다.

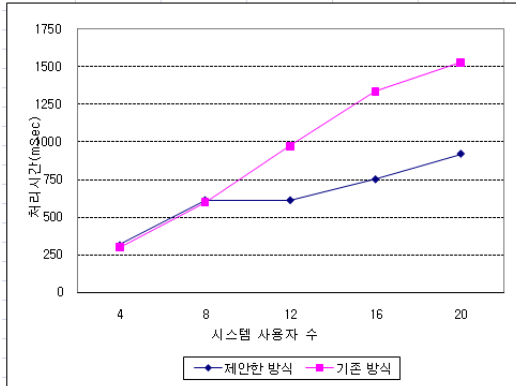


그림 8. SUBSCRIBE 메시지 서버 처리 시간 비교
(최대 부하 허용값=400)
Fig. 8. Comparison of Server Processing Time for SUBSCRIBE Message(Max. Load Allowance=400)

V. 결론 및 향후 과제

본 논문에서는 사용자 수의 증가에 따라 서버의 부하를 효율적으로 분산하여 처리 할 수 있고 확장 호처리 언어를 적용하여 사용자에게 다양한 프레즌스 서비스 기능을 제공 할 수 있는 새로운 다중 서버 방식을 사용한 인스턴트메시지 서비스 구조를 제안하였다. 인스턴트메시지 서비스 사용자는 프레즌스 정보 및 자신이 원하는 내용을 기술한 호처리 언어 스크립트를 서버에 등록하며, 서버는 인스턴트메시지의 처리 및 프레즌스 서비스와 호처리 언어 스크립트 실행 기능을 제공한다. 이를 위하여 서버의 부하를 제어하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 포맷과 새로운 서버 구조가 제시되었다. 또 서버에 인스턴트메시지 처리 모듈을 통합하여 인스턴트메시지 처리와 관련된 SIP 메시지의 발생 양을 줄여 시스템의 효율을 더욱 높일 수 있도록 하였다.

제안된 시스템의 성능 분석을 위하여 복수개의 서버를 갖춘 환경에서 사용자 수를 증가시켜 가면서 서버에서의 처리 시간을 측정하였다. 측정 결과 본 연구에서 제안한 방식이 최소 34.7%에서 최대 44.2% 까지 처리시간의 개선 효과가 있음을 보였다. 향후 과제로는 대역폭이 상대적으로 작은 무선 모바일 환경에서 메시지 발생량을 최소화하고 처리 시간을 감소시킬 수 있는 개선된 인스턴트메시지 서비스 구조를 연구 할 필요가 있고 사용자 수 및 PC를 증가 시킨 대규모 인스턴

트메시지 서비스에서의 성능을 검증 할 필요가 있다.

참고문헌

- [1] D.A. Hour, E. Aoki, "Presence & Instant Messaging Peering Use Cases", RFC 5344, June 2008
- [2] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.
- [3] P. Saint-Andre, "Extensible Messaging and Presence Protocol", RFC 6502, March. 2011
- [4] J. Lennox, Xiaotao Wu, H.Schulzrinne, "CPL: A Language for User Control of Internet Telephony Service," RFC3880, Oct. 2004.
- [5] H. Sugano, G. Klyne, "Presence Information Data Format (PIDF)," RFC 3863, August 2004.
- [6] C. S. Jang, "A New Presence Lists Service in SIP Environment" Journal of The Korea Society of Computer and Information, Vol 15, No.12, pp 67-73, December 2010.
- [7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.
- [8] M. Lonnfors, E. Leppanen, H. Khartabil, J. Urpalainen, "SIP Extension for Partial Notification of Presence Information", RFC 3261, September 2011.
- [9] A. Niemi, M. Lonnfors, E. Leppanen, "Publication of Partial Presence Information", RFC 5264, September 2008.
- [10] H. G. Jo, K. S. Lee and C. S. Jang, "An Efficient Processing of Multimedia Message in SIP-based Instant Messaging Service" Journal of The Korea Contents Association, Vol 15, No.12, pp 152-159, September 2011

저 자 소 개



장 춘 서

1993: 한국과학기술원 공학박사
1981년 ~ 현 재: 금오공과대학교
컴퓨터공학과 교수
관심분야: SIP, 임베디드시스템,
인터넷텔레포니
Email : csjang@kumoh.ac.kr



이 기 수

1982: 서울대학교 대학원 공학석사.
1982년 ~ 현 재: 금오공과대학교
컴퓨터공학과 교수
관심분야: SIP, 임베디드시스템,
데이터베이스
Email : kysoolee@kumoh.ac.kr