

통합통신서비스를 위한 SIP Servlet 응용 사례연구

김종규
티맥스소프트 연구소

A case study on developing SIP Servlet application

Kim, Jong-Kyou
TmaxSoft Research Center
E-mail : jkkim@acm.org

요 약

통합통신서비스는 차세대 정보서비스산업에서 중요한 위치를 차지할 것으로 기대되고 있다. 전화, 이메일, 인스턴트 메시징, 팩스, 음성사서함과 같은 여러 종류의 통신매체를 통합한 서비스를 제공하기 위해서는 각매체를 연결하는 통일된 통신 프로토콜이 필요한데, 현재 가장 각광받고 있는 통신프로토콜은 인터넷 기술에 기반한 SIP (Session Initiation Protocol)이고, 이를 활용하기 위한 프로그래밍 인터페이스로는 기존 웹프로그래머에게 익숙한 Servlet 형식을 갖고 있는 SIP Servlet 이 제공되고 있다. 그러나 프로그래머 인터페이스의 유사성과는 별개로 시스템의 처리용량이 증가할 때 발생하는 성능상의 문제를 해결하기 위해서는 SIP 응용프로그램의 특성을 고려하여 진단을 수행하고 운영파라미터를 조정하여야 한다. 또한 응용프로그램설계에 있어서도 전통적인 웹응용프로그램에서는 무시되었던 부분이 중요한 성능방해요소로 작용하기 때문에 주의가 필요하다. 이 논문은 공개된 인터넷망에서의 서비스를 목적으로 개발된 통합통신서비스를 개발하는 과정을 중심으로 성능문제의 진단과 개선 사례를 소개하고 응용프로그램 설계에서의 고려사항에 대해서 논의한다.

1. 서론

통합통신서비스는 (Unified communication service) 차세대 정보서비스산업에서 중요한 위치를 차지할 것으로 기대되고 있다 [1]. 통합통신서비스를 제공하는 플랫폼은 여러가지 방안이 제시되어왔으나 오늘날에는 IP 를 통신 미디어로 사용하고 사용자 정보, 위치 등은 SIP 라는 규약을 통해서 제공하는 방식이 일반화되어가고 있다. SIP

는 HTTP 와 매우 유사한 형태를 띠고 있고 IETF 에 의해서 표준화된 방식으로 최근 Servlet 표준이 제안되어 기존의 기업 전산환경에 잘 융합될 수 있는 방안이 마련되었다 [6].

그러나 SIP Servlet 은 비교적 최근에 적용되기 시작한 기술로 실업무에 적용할 경우 기존의 HTTP Servlet 과는 다른 여러가지 문제점들을 야기한다. 특히 Servlet 규격이 주로 HTTP 와

같은 동기적 통신방식을 가정하고 설계되었기 때문에 SIP 을 Servlet 형태로 다루는 경우 기존 HTTP 기반 Servlet 시스템을 운영할 때와는 달리 병목구간을 파악하는 것이 훨씬 더 어렵다는 문제가 있다. 예를 들어 HTTP 기반 서비스의 경우 요청에서 응답에 소요되는 시간이 긴 서비스를 찾고 그 서비스가 호출하는 함수나 외부서비스의 시간을 측정하는 비교적 단순한 방식을 적용해도 전체 성능에 영향을 미치는 구간을 비교적 용이하게 파악할 수 있다. 그러나 SIP 를 기반으로 한 서비스의 경우 서비스 요청에 대한 응답이 비동기적으로 여러 모듈과 서비스를 통해서 진행되기 때문에 병목구간을 파악하는 것이 용이하지 않다.

따라서 SIP Servlet 기반 응용프로그램의 성능을 측정하고 분석하기 위해서는 기존 HTTP Servlet 기반 서비스의 성능 측정 및 분석과는 다른 새로운 프레임워크가 필요하다. 이를 위해서 먼저 SIP Servlet 을 소개하고 HTTP 통해 SIP 서비스를 호출하는 통합통신서비스를 설명한 후 성능측정방식과 병목구간의 파악을 위해서 측정값을 설명한다. 마지막으로 병목을 일으킨 원인과 증상을 파악하는 데 도움이 되는 지표를 설명하도록 한다

2. SIP Servlet

SIP 는 멀티미디어 통신을 수행하기 위해서 상대방을 찾고, 연결하고, 통신을 종료하는 각종 신호를 전달하는 데 필요한 규약이다 [4]. SIP 는 IP 를 기반으로 설계되어 HTTP 와 같은 다른 많은 인터넷 기반 프로토콜들과 마찬가지로 텍스트를 기반으로 하고 있다. 특히 HTTP 와의 유사성은 인터넷 프로그래밍 환경에 친숙한 대다수 프로그래머가 쉽게 이해하고 사용할 수 있다는 장점을 제공한다. SIP 는 현재 대표적인 인터넷 멀티미디어 응용 시스템인 VoIP 나 비디오 컨퍼런싱 등에 가장 빈번하게 적용되고 있다.

비록 SIP 기술이 HTTP 와 많은 유사성을 갖

고 있기는 하지만 HTTP 프로그래밍 환경을 그대로 이용할 수 있는 것은 아니다. HTTP 기술의 경우 많은 응용 시스템 개발 경험을 통하여 컴포넌트 기반의 효율적인 개발을 지원하는 Servlet 기술이 보편적으로 보급되어 있다 [3]. 이에 반하여 SIP 응용시스템의 개발을 지원하는 시스템은 찾기 어려웠다. SIP Servlet 은 HTTP Servlet 기술과 많은 부분을 공유하여 기존의 Servlet 프로그래머가 쉽게 적응할 수 있도록 설계된 응용시스템 개발기술이다 [6,5].

SIP 을 위한 Servlet 기술은 Sip Servlet 이라는 이름으로 불리고 HTTP Servlet 과 마찬가지로 Java 언어에 기초하고 있다 [6]. 기업환경에서 Java 언어는 기업의 전산환경을 통합하는 사실상의 표준이 되어가고 있다 [2]. VoIP 나 비디오통화, 통합통신서비스와 같은 진보된 멀티미디어 통신에 대한 요구가 대체로 기업에서 제기되고 있기 때문에 SIP Servlet 이 기존의 기업정보환경의 구축에 널리 보급된 Java EE 환경과 잘 융합된다는 점은 기업 전산관리자의 관점에서도 매력적인 부분이다.

SIP Servlet 은 RFC 3261 에 정의된 기본규약 뿐 아니라 다양한 확장규약을 지원한다. 기본 규격은 세션을 생성하고, 정보를 전달하고, 세션을 종료하는 기본 기능을 구현하는 데 필요한 내용을 담고 있고, 확장 규격에서는 각종 이벤트를 전달하고 다른 세션과 연결하는 등의 일을 수행하는 데 필요한 규약을 정의하고 있다. 단순히 IP 상의 무료 통화를 중재하는 것과 같은 간단한 VoIP 시스템 외에는 대부분 확장 SIP 규약이 활용된다. 따라서 SIP Servlet 은 이러한 확장규격을 포함하여 정의되어 있다.

3. 통합 통신 서비스

통합통신서비스는 여러가지 통신미디어를 통합하여 사용자에게 좀더 진보된 서비스를 제공하는 것이다. 예를 들어 사무실로 부재 중 전화가 왔을

때 녹음된 내용을 e-mail 로 전송하게 되면 좀더 신속하게 해당 상황을 파악할 수 있을 것이다. 또 전화회의를 개최하고자 하는 경우에도 대상자에게 일일이 연락해서 별도의 세션을 여는 것 보다는 웹페이지로 작성된 주소록에서 통화대상자를 복수로 선택해서 바로 연락하는 편이 더 편리할 것이다.

간략화한 시나리오를 설명하면 다음과 같다. 먼저 웹페이지에서 대상자를 선정하여 웹서버에 요청을 보내면 이 내용은 어플리케이션 서버로 전달된다. 어플리케이션 서버는 URL 을 검토하여 Servlet 을 선정하고 요청내용을 해당 Servlet 으로 전달한다. 해당 Servlet 에서는 대상자에게 전화를 걸어 회의에 초청하기 위해서 필요한 SIP 메시지를 작성하여 개별단말기에 전달하고, 해당 통화를 관리하기 위한 별도의 세션을 생성한다. 상대방이 통화 확인을 하면 (전화기를 들면) 웹세션과는 별도로 SIP 을 관리하는 세션에서 통화의 나머지 부분을 진행한다.

성능측정은 x86 기반의 Linux 서버에서 JEUS SIP 어플리케이션서버를 활용하여 수행하였다. 순수한 SIP 메시지의 경우 JEUS 는 초당 900 개 이상의 통화를 처리하는 성능을 보여주었고 웹트랜잭션의 경우 2000 회 이상을 처리할 수 있음을 확인하였다. 서버와 단말은 모두 Fast Ethernet (100Mbps) 에 스위칭허브를 사용하고 HTTP 메시지는 TCP 를 활용한 전통적인 방식으로, SIP 메시지 전달은 UDP 를 활용하는 방식을 사용하였다. 측정의 예상치는 적어도 초당 100 개 이상의 요청을 처리하는 것이다.

기존 정보시스템과 통합되어 통신서비스를 제공하는 예제로는 주소록에서 검색한 인물에게 통화를 요청하는 서비스인 Click to Dial 서비스가 가장 대표적이다. 이 어플리케이션의 시나리오를 간략화 하면 다음 그림 1 과 같다. 먼저 웹어플리케이션이 UA1 에 통화를 요청한다. 그러면 SIP 트랜잭션이 시작되어 응답이 오기를 기다리는 상태

가 된다. UA1 에게 메시지가 전달되면 180 Ringing 을 응답으로 보낸다. 사용자가 수화기를 들면 (혹은 통화버튼을 누르면) 200 Ok 메시지가 전달된다. SIP Servlet 은 이 메시지가 도착하면 앞서 보낸 요청메시지의 응답임을 파악하여 다음 UA2 에 비슷한 과정을 거쳐 통화를 요청한다.

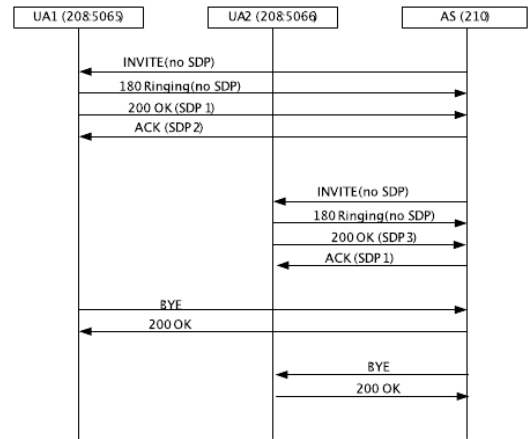


그림 1. 통화흐름도

기존 웹어플리케이션 개발자의 관점에서는 이 시나리오에서 성능상 문제가 될 만한 부분을 발견하기가 쉽지 않다. 각각의 메시지에 대해서 한 차례 데이터베이스에 접근하여 관련 정보를 갱신하는 수준으로 웹트랜잭션이라면 초당 수천회 정도는 무리없이 수행할 수 있을 것으로 기대할 것이다. 그러나 실제 SIPp 라는 테스트 도구를 사용하여 측정한 결과 초당 30 에서 40 통화밖에 지원하지 못하는 것으로 파악되었다. 먼저 네트워크 트래픽을 검토하여 파악한 내용은 다음과 같다.

- 한 통화를 연결하기 위해서는 최소 12 차례 메시지가 교환된다. 이는 HTTP 관점에서 보면 약 12 회의 트랜잭션에 해당한다.
- 12 회의 메시지 교환 이외에도 통화가 유지되는 동안 주기적으로 타이머가 동작한다. 타이머의 동작 시점에는 데이터베이스 접속이 이루어진다. 이 과정은 사실상 하나의 웹트랜잭션에 해당한다.
- 각각의 메시지처리에서 추정된 성능은 초당 100 에서 200 통화가 진행되어야 정상이다.

이와 비교하면 약 10 분의 일 정도밖에 성능이 나오지 않는다.

전통적인 웹응용프로그램의 성능분석에서는 어플리케이션서버에서 service() 를 호출하는 시간을 측정하고 이 시간을 호출되는 서비스의 URL 과 함께 출력하면 시간이 오래 소요되는 Servlet 을 비교적 쉽게 찾아낼 수 있다. 일단 서블릿이 확정이 되면 서블릿 수행을 구간별로 측정하여 어디에서 가장 많은 시간이 소요되는 지를 찾아내는 것도 어려운 일이 아니다. 그러나 SIP 과 HTTP 를 통합한 이 응용프로그램에서는 두 가지 기능이 모두 Servlet 의 service() 메소드를 통하여 수행되고 있음에도 불구하고 성능에 영향을 주는 서비스 호출을 찾을 수가 없었다. 따라서 자원활용도를 점검하여 타이머서비스에서 병목이 발생한다는 가설을 세우게 되었다..

이 가설을 확인하기 위해서 먼저 타이머서비스에서 데이터베이스를 접근하는 부분을 제거하였다. 좀더 구체적으로 설명하면 업무요건을 분석하여 해당 데이터를 데이터베이스에 기록하지 않고 단순 화일에 로그형식으로 기록한 후 후처리를 수행하는 방식으로 업무 로직을 변경한 것이다. 이렇게 변경하고 성능을 측정한 결과 이론상 근사치인 초당 150 통화를 처리한다는 사실을 확인하였다.

4. 결론 및 토의

SIP Servlet 은 통합통신시스템을 구축하는 데 효과적으로 활용될 수 있는 기술이다. 실시간 멀티미디어통화를 연결하는 기술인 SIP 기술을 활용하면서 기존의 기업정보환경에 손쉽게 추가할 수 있는 형태로 제안된 기술인 SIP Servlet 기술은 SIP 을 활용한 응용프로그램의 작성과 설치를 용이하게 하였지만 기존 HTTP Servlet 과는 달리 메시지가 비동기적으로 전달되어 타이머 등에 의존한 비동기 통신기술에 많이 의존하고 있다. 실제 응용프로그램에서 이러한 특성이 성능에 어떤 영향을 미치고, 예측치와 차이가 나는 부분을

어떻게 파악할 수 있을지에 대한 개략적인 이해를 얻기 위해서 가장 대표적인 응용프로그램인 Click to Dial 서비스에 대해서 실제 응용에서 요구되는 부가 기능을 추가하여 성능시험을 진행하였다. 초기 성능측적에서는 예측치와 동떨어진 결과가 나왔음에도 기존 service() 메소드 수행시간을 측정하는 HTTP Servlet 의 병목구간 분석방식으로는 병목구간을 파악할 수 없었다.

따라서 향후 SIP 를 활용한 통합통신 응용프로그램에서 성능문제가 발생한 경우 타이머서비스에서 사용하는 자원을 검토할 필요가 있다. 또한 어플리케이션서버 벤더는 이 부분이 병목을 일으킬 수 있는 중요한 부분임을 인지하고 현재 동작 중인 타이머의 수 뿐만 아니라 데이터베이스세션 등 중요자원의 점유 상태를 모니터링 할 수 있도록 기능을 제공하여야 할 것이다.

[참고문헌]

- [1] Steve Blood. SIP deployment emerges in enterprise telephony. Gartner Research Report (G00125672), 2005.
- [2] Roberto Chinnici and Bill Shannon. Java(tm) platform, enterprise edition (Java EE) specification v6. JSR 316 (Public Review), 2009.
- [3] Rajiv Mordani. Java servlet 3.0 specification. JSR 315 (Public Review), 2008.
- [4] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393.
- [5] Sun Microsystems. The SIP Servlet Tutorial, 2009.
- [6] Jarek Wilkiewicz and Mihir Kulkarni. Sip servlet v 1.1. JSR 289 (Final), 2008.