MoIM (Mobile Internet Middleware)-Message 시스템 설계 및 구현

정인철[0] 남궁한
이동분산처리연구팀, 컴퓨터.소프트웨어 연구소, 한국전자통신연구원
(jic[0], nghan)@etri.re.kr

Design and Implementation of Mobile Internet Middleware Message

In-Cheol Jeong[0] Han NamGoong
ETRI-Computer & Software Technology Laboratory

요 약

MoIM(Mobile Message Internet Middleware)-Message 시스템은 이러한 메시지 시스템의 요구사항을 만족하는 메시지 시스템으로서 *Point-to-Point* 및 *Publish-Subscribe* 방식을 제공하는 시스템이다. 이 시스템의 통신 방법은 RMI(Remote Method Invocation)을 사용한다. 최근의 인터넷 사용자 수가 급증함에 따라서 메시징 서버를 사용하는 동시 클라이언트의 수도 수천에서 수만으로 이를 것으로 예상되어 대규모의 동시 사용자를 지원하는 연결 기술이 필요하다. 이 논문에서는 이러한 대용량의 데이터를 RMI 통신방법을 사용하여 설계 및 구현한 방법을 기술한다.

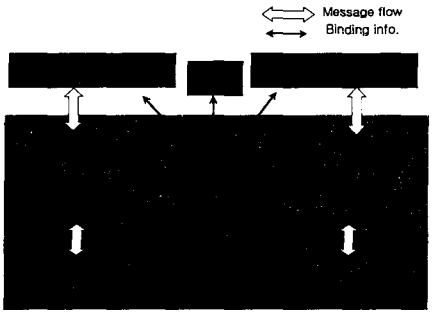
1. 서 론

인터넷의 확산 및 활용 증가와 함께 기업간, 개인 기업간의 정보 교환이 빈번하게 이루어 지고 있으며, 이러한 정보 교환을 지원하는 메시지 시스템의 요구도 점점 증대되고 있다.

MoIM(Mobile Message Internet Middleware)-Message 시스템은 이러한 메시지 시스템의 요구사항을 만족하는 메시지 시스템으로서 Point-to-Point 및 Publish-Subscribe 방식을 제공하는 시스템이다. 이 시스템의 통신 방법은 RMI(Remote Method Invocation)을 사용한다. 이러한 메시지 시스템은 기업간 거래나 개인간의 정보 교환을 위하여 분산 환경에서 소프트웨어 프로세스간 데이터 교환을 위하여 미국 IBM등에서는 메시징 시스템을 별도로 패키지화 하여 상품화하고 있다. 최근의 인터넷 사용자 수가 급증함에 따라서 메시징 서버를 사용하는 동시 클라이언트의 수도 수천에서 수만으로 이를 것으로 예상되어 대규모의 동시 사용자를 지원하는 연결 기술이 필요하다. 이 시스템은 이러한 요구사항을 만족하기 위한 여러 가지 특징을 가지고 있다. 이 논문에서는 이러한 대용량의 데이터를 RMI 통신방법을 사용하여 설계 및 구현한 방법을 기술한다.

2. 관련연구

메시징 시스템들은 조직간 실시간으로 정보를 공유하고 비즈니스의 처리를 통합하기 위하여 필요한 핵심적인 요소이다. 기업들은 메시징 시스템에서 제공하는 시기적절한 정보의 배달과 효율성의 증가로 더욱 큰 이익을 보고있다. 1998년 SUN 마이크로 시스템사는 응용 프로그램들이 메시지를 생성하고 , 전송하고 수신하고, 읽도록 하기 위한 공통의 방법으로 메시징 서비스 (JMS) 표준을 정의하였다.



(그림 1) JMS 구성요소

메시징 시스템의 성능을 결정하는 요소는 다양하고 요소간에도 상호 영향을 주는 등 복잡하지만 , 여기서는 이러한 요소들 중에서 흐름제어와 scalability 문제, 지연감소와 관련된 기술적 사항들을 JMS 서버에서의 선도업체인 FioranoMQ와 SonicMQ의 분석을 통하여 분석을 한다.

2.1 FioranoMQ

FioranoMQ는 최초로 JMS를 구현한 벤더로서 AT&T Wireless, KPMS, FedEx, Motolora와 JP Morgan을 포함하는 선두 기업에 의해 사용되어 지고 있다. 120개 기관이 고유한 탁월한 시스템 구조와 성능 때문에 FioranoMQ를 선택하였다. FioranoMQ는 JMS 1,2 적합성 수준을 시험하기위해 다른 ISV들이 사용하는

JMS 시험도구를 개발하여 메시징 시스템에서의 우위를 더욱 굳히고 있다.

FioranoMQ 시스템은 클러스트링과 로드밸런싱, 중요한 보안 기능(ACL/ACE, SSL)을 포함하여 메시징 서버의 중요한 요구사항을 구현하고 있다. 아울러 응용 서버 뿐만 아니라 LDAP 서버의 통합, NT/Unix 환경의 지원, c/c++ 클라이언트의 XML 메시징 유형과 SOAP 를 지원한다.

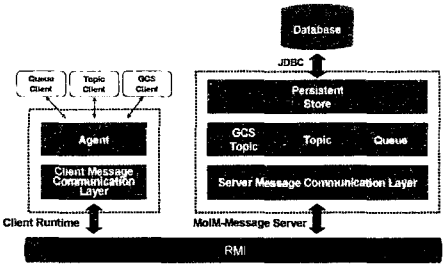
2.2 SonicMQ

Sonic Software에서 개발한 SonicMQ는 JMS의 전 부분을 자바로 작성한 제품으로 pub/sub와 point-to-point 모델을 구현한 제품이다. SonicMQ는 XML메시지 뿐만 아니라 서버 클러스터링과 같은 JMS 확장 기능도 제공한다.

3. MoIM-Message 시스템 구조

MoIM-Message 기본엔진은 p2p(point-to-point) 메시징 모델과 pub/sub(publish/subscribe)의 두 가지 유형의 모델을 지원하며 pub/sub 메시징 모델을 확장하여 그룹 통신 기능을 제공한다. P2P 메시징 모델은 큐(Queue)라는 가상 채널을 사용하여 클라이언트 응용간의 일대일 메시지 전달을 가능하게 하며 pub/sub 메시지 모델은 토픽(Topic)이라는 가상 채널을 이용하여 여러 클라이언트 응용간의 일대다 형태의 메시지 전달 및 분배를 가능하게 한다.

MoIM-Message 기본 엔진은 크게 메시지 전달 및 저장 서비스를 담당하는 서버(MMS Server)와 클라이언트 응용에게 메시지 전달 서비스를 제공하는 클라이언트 런타임(client runtime) 부분으로 이루어진다.



(그림 2) MMS 기본엔진 구성요소

MMS에서 제공하는 기능을 사용할 수 있는 클라이언트 응용은 크게 큐 클라이언트(Queue Client), 토픽 클라이언트(Topic Client) 및 그룹 통신 클라이언트 (GCS Client)로 분류될 수 있으며 각 클라이언트 응용은 각각의 요구에 따라서 클라이언트 런타임이 제공하는 에이전트를 생성하여 에이전트의 메소드를 통하여 메시지를 전달 할 수 있다.

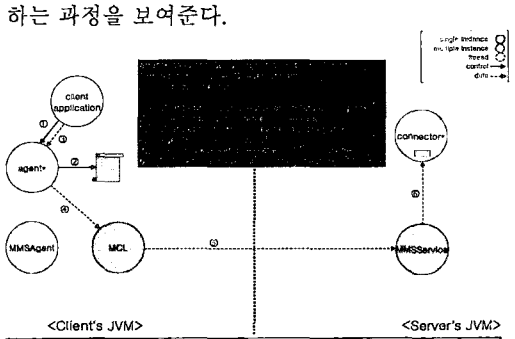
네트워크 통신 프로토콜로서 트랜스 포트 층위에 동작하는 RMI를 사용하였으며 메시지 송수신(RMI 통신) 관점과 논리적인 관점을 분리하여 메시징 모델의 설계와 구현에 효율성을 높이기 위해 클라이언트 런타임과 서버 측에 각각 RMI 통신을 담당하는 메시지 통신계층을 두고 이를 통하여 클라이언트 응용과 서버간에 메시지 전달이 이루어 지도록 설계하였다.

• 메시지 수신

3.1메시지 송수신 시나리오

• 메시지 송신

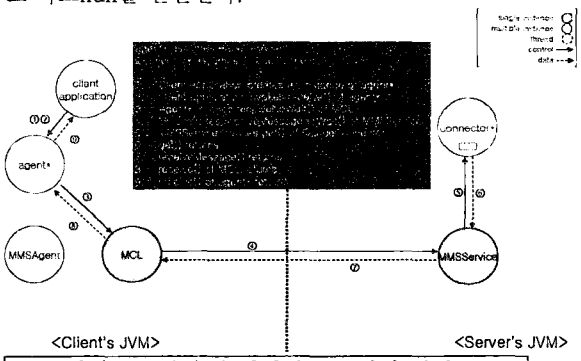
다음 그림은 큐 또는 토픽에 메시지를 전송할 때 발생하는 과정을 보여준다.



1) 클라이언트 응용이 에이전트를 하나 생성
2) 클라이언트 응용이 메시지를 하나 생성
3) 클라이언트 응용이 에이전트의 send() 호출
4) 에이전트가 MCL의 send() 호출
5) MCL은 MMSService의 sendMessage() 호출
6) MMSService가 connector의 put() 호출

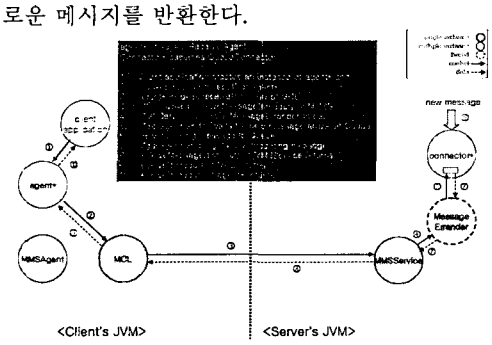
• 메시지 수신

다음 그림은 큐 또는 토픽에서 동기적으로 메시지를 가져오는 과정을 보여준다. 만약 큐 또는 토픽에 메시지가 없을 경우는 새로운 메시지가 도착하기를 기다리지 않고 바로null을 반환한다.



1) 클라이언트 응용이 에이전트를 하나 생성
2) 클라이언트 응용이 에이전트의 receive() 호출
3) 에이전트는 MCL의 receiveNoWait() 호출
4) MCL이 MMSService의 receiveMessageNoWait() 호출
5) MMSservice가 QueueConnector의 get() 호출
6) QueueConnector의 get() 반환
7) MMSService의 receiveMessageNoWait() 반환
8) MCL의 receive() 반환
9) 에이전트의 receive() 반환

다음 그림은 큐 또는 토픽에서 동기적으로 메시지를 가져오는 과정을 보여준다. 만약 큐 또는 토픽에 메시지가 없을 경우는 새로운 메시지가 도착하기를 기다렸다가 새로운 메시지를 반환한다.



1) 클라이언트 응용이 에이젼트를 하나 생성하고 에이젼트의 receive(timeout) 호출
2) 에이젼트는 MCL의 receive(timeout) 호출
3) MCL은 MMSService의 receiveMessage(timeout) 호출
4) MMService는 MessageErrander 쓰레드를 하나 생성하여 실행
5) MessageErrander 쓰레드는 큐에서 새로운 메시지의 도착을 기다림
6) 새로운 메시지가 큐에 도착됨
7) MessageErrnader쓰레드가 새로 도착한 메시지를 가져옴
8) MMSService의 receiveMessage(timeout) 반환
9) MCL의 receiveMessage(timeout) 반환
10) 에이젼트의 receiveMessage(timeout) 반환

4. 구현 및 시험

현재의 MoIM-Message 성능(메모리 사용량, 소요시간)측정을 통한 성능 향상을 도모하기 위한 목적으로 시험을 실행하였다. 아래의 시험은 다음과 같은 가정을 바탕으로 한다. 가정은 Only P2P 메시지 모델, 단일 destination, 고정된 크기 메시지를 기준으로 한다.

성능 평가 방법은 응용측과 서버측을 구분하여 평가한다. 응용측은 초기화 연결 후에 서버연결, 메시지 생성 및 송신 순으로 평가하며 서버측은 초기화 이후에 응용을 등록하고 메시지를 도착한 후에 메시지 전달 순으로 평가한다. 환경 변수로는 송신측 클라이언트가 10개, 100개, 100개 메시지를 송신하며, 수신측은 4가지 메시지 수신 방법에 따라서 receiveNoWait(), receive(#), receive(0), 비동기 수신방법을 이용한다. 성능 평가 Tool은 Numega를 이용한다.

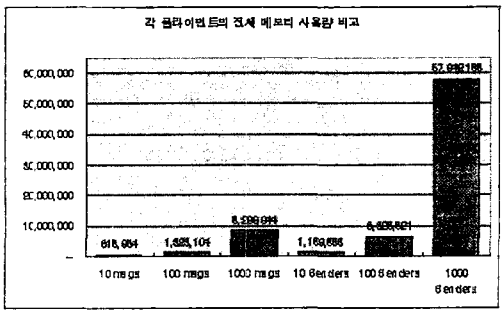
Numega Tool은 응용의 신뢰성 및 성능 향상을 위한 성능 분석툴로서 응용성능의 절대치가 아닌 상대적인 값을 나타낸다. 이 시험에서는 MemoryProfiler라고 하는 메모리 성능 툴을 이용하였다.

성능시험의 순서는 송신응용이 동기수신 응용에게 메시지를 전달하는 첫번째 방법과 비동기 수신응용을 등록하고 송신응용을 수행하는 순서로 진행된다.

1) 송신응용의 성능 시험은 초기화 후에 서버에 연결하여 송신하는 방법을 취한다. 송신응용1의 경우에 한 개의 클라이언트가 10, 100,1000개의 메시지를 송신한다. 송신응용 2의 경우에는 10, 100, 100개의 송신 쓰레드를 가진 클라이언트를 동작시킨다.

2) 수신응용의 성능 시험은 초기화 이후에 서버에 연결하며 수신한다. 송신응용1인 경우에 4가지 수신방법을 시험한다.

3) 수신응용의 성능 시험은 초기화 이후에 메시지가 도착된 후에 메시지를 전달한다. 송신응용 1의 경우에 메시지를 수신한다. 4가지 방법으로 접근하는 수신응용에게 메시지를 전달한다.



5. 결론

이 논문은 MoIM-Message 의 통신방법으로 RMI를 사용하여 설계 및 구현한 방법을 기술하였다. Nemega를 이용한 메모리 시험을 통하여 초기화, 연결, 메시지 송신 과정 가운데 1000개 이상의 메시지를 전달했을 때 성능이 급격하게 저하되는 현상을 발생하였다. 따라서 이에 대한 해결방법으로 Send, Receive 시 전용 쓰레드를 통하여 성능을 높일 수 있도록 구현할 예정이다. 향후 MoIM-Message 시스템은 모바일 클라이언트 환경이나 Reliable 메시지 보장 방법을 허용하기 위해서 소켓 통신 방법과 쓰레드 풀을 이용하여 개선할 예정이다.

참고 문헌

[1] A Fiorano White Paper , "A Guide to undestanding the Pluggable , Scalable connection Management(SCM) Architecture in FioranoMQ5", Feb 2001

[2] P.Giotta, S.Grant, M.Kovacs, S.Maffeis, K.Morrison, G.Raj, M.Kunnumpurath, "Professional JMS Programming", 2001

[3] Richard Monson-Hoefel & David A. Chappel, "Java Message Service", O'Reilly, 2001