

X-Internet 기반의 어플리케이션과 웹 어플리케이션의 성능평가

이우기* 이정훈* 윤석호** 차창일** 최영미*

* 성결대학교 공과대학 컴퓨터공학부

** 한양대학교 정보통신 대학원

Performance Analysis on X-Internet Based Applications and Web Applications

Wookey Lee* Jeong-Hoon Lee* Chang-il Cha** Seok-Ho Yoon** Young-Mi Choi*

* Div. of Computer Science, Sungkyul University,

** Div. of Information & Computer Science, Han-Yang Graduate school

요 약

최근 응용 프로그램은 대다수가 웹으로 이루어져 있다. 그것은 빠르게 변화하는 비즈니스 시장에서 살아남기 위해 기업들이 배포 및 유지보수가 쉬운 웹 응용프로그램을 선호했기 때문이다. 하지만 이제는 다양한 고객의 요구조건을 웹 응용 프로그램이 충족시키기에는 한계가 있다. 이런 문제를 해결하고자 많은 기업은 앞 다투어서 새로운 인터넷 구조인 X-Internet 기반의 응용 프로그램을 출시하고 있으면 기업들도 도입을 서두르고 있다. 본 논문은 X-Internet 기반의 응용프로그램과 웹 응용프로그램의 성능을 비교 분석함으로써 X-Internet 기술의 가능성을 알아보고자 한다.

지막으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

1. 서론

응용프로그램 분야는 초기 메인프레임 시대와 C/S 환경을 거쳐 현재의 웹 환경으로 발전되어왔다. 하지만 2000년대에 들어오면서 클라이언트 컴퓨터의 눈부신 발달과 고객 요구사항의 다양성이 대두되면서 웹을 기반으로 하는 인터넷으로는 이러한 변화를 충족시키지 못하고 있다. 이에 따라서 클라이언트 컴퓨터의 성능을 적극적으로 활용하면서 다양한 고객의 요구사항을 충족시키며 웹의 장점을 그대로 유지할 수 있는 X-Internet이라는 새로운 인터넷 구조가 주목을 받기 시작했다 [1, 7, 8]. 많은 소프트웨어 회사들은 X-Internet 솔루션을 공급하기 시작했고, 기업들은 서둘러 X-Internet을 이용한 비즈니스 어플리케이션 도입을 검토하기 시작했다. 그러나 아직은 대다수가 X-Internet에 대한 정확한 개념조차 정립되지 못했고 현재의 웹 응용프로그램과 같이 비즈니스에 바로 적용하기에는 아직 많은 부분에서 검증되지 못한 것도 사실이다 [3, 5, 7]. 이에 따라 본 논문에서는 X-Internet에 대해 소개하고 웹 응용프로그램과 X-Internet 기술을 비교 분석하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 X-Internet에 대한 정의와 관련기술들에 대해서 알아보고 3장에서는 성능 비교를 위한 환경구성에 대해서 기술하며 4장에서는 실험결과와 결과에 대한 분석을 기술한다. 마

2. X-Internet

2.1 X-Internet의 정의

X-Internet이란 executable & extended Internet을 줄인 표현으로써, XML을 기반으로 하며 Web 아키텍처와 Client/Server 아키텍처의 한계성을 극복하고 장점만을 수용하기 위한 새로운 인터넷 아키텍처를 의미한다 [4, 8].

실행 가능한 인터넷(executable Internet)이라는 말은 X-Internet은 현재 웹 환경과 같이 HTML을 이용해서 단순한 웹브라우저에서 실행하는 형태가 아닌 C/S환경에서와 같이 풍부한 사용자 인터페이스(Rich UI) 어플리케이션을 browsing해내는 어플리케이션 환경으로서 능동적인 인터넷 환경을 의미한다.

확장된 인터넷(extended Internet)이라는 말은 Smart Client를 통해 Web 브라우저, 4GL Tool, 독립모듈 등의 다양한 개발 및 운용환경을 지원하고 기존 S/W Component나 각종 외부장비들과의 지능적인 인터페이스를 지원한다. 또한 기존 시스템과 손쉽게 연동할 수 있고 Windows, WinCE, Palm, Embedded Linux 등의 Multi-OS 환경을 지원함으로써 외부 환경과의 다이내믹한 연동이 가능하다는 의미에서 확장형 인터넷 시스템이라 부르는 것이다 [4, 5, 7].

2.2 X-Internet 기술의 특징

X-Internet의 기술적 특징은 첫째로 C/S환경에서의 풍부한 인터페이스환경에서 단순한 웹으로의 전환을 이루었던 환경에서 HTML이라는 표현을 표준으로 했던 것과는 달리 XML을 정보전달의 표준으로 삼고 다시 자체 인터페이스를 사용함으로써 풍부한 사용자 인터페이스를 제공할 수 있도록 해준다. 둘째 X-Internet에서는 어플리케이션을 1번만 다운로드하고 어플리케이션에 필요한 데이터만 보내고 받기 때문에 트래픽 양을 획기적으로 줄일 수 있다. 셋째 플랫폼에 독립적인 XML을 사용함으로써 어떠한 시스템에서도 연동이 가능하도록 한다는 것이다.

3. Smart Client(X-Internet)의 특징

우선 X-Internet을 지원하는 스마트 클라이언트와 Java Web Application에 대해 좀 더 자세히 살펴보기로 하자.

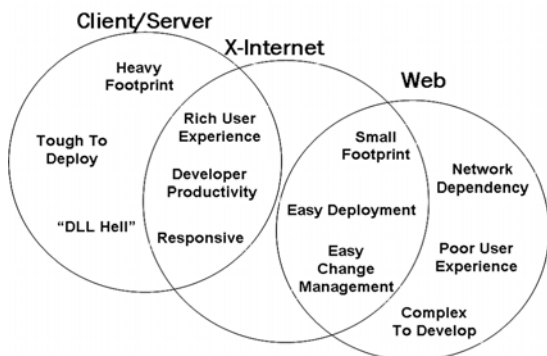


그림 1. X-Internet의 특징

본 연구에서는 X-Internet 솔루션인 .NET 기반의 Smart Client와 Java Web Application에 초점을 맞추었다. 일단 .NET 기반의 Web Application은 Smart Client와 비슷한 구조와 컴포넌트를 가지고 있어서 객관적 비교가 어렵기 때문에 가장 흔하게 쓰이고 있는 JSP를 가지고 Web Application을 구축한다. 비교 기준은 컴퓨터의 성능과 상관이 있는 CPU 사용량, Memory 사용량 그리고 총 작업시간과 통신에 상관이 있는 패킷 양을 측정한다. 실험에 사용할 데이터는 www.tpc.org에서 채택하고 있는 TPC-H 모델과 질의를 채택하였다. 실험방법은 동일한 쿼리에 대해서 어떤 변화가 일어나는지 보고 그것을 분석해서 클라이언트와 서버에 부하 정도를 측정해 어떤 환경구조가 성능이 뛰어난지 알아내는 것에 초점을 맞추었다. 클라이언트 컴퓨터의 물리적 환경과 서버의 물리적 환경은 동일한 조건으로 테스트함으로써 객관성을 유지하고자 한다. 따라서 실험에서는 실제에 사용가능하게 중요한 부분은

웹서비스로 대체하였다.

4. 실험 모형

4.1 Smart Client 와 Java Application

Smart Client는 .NET Framework가 설치되어 있는 클라이언트에서만 실행가능하며 웹브라우저에서 서비스를 제공하는 사이트에 접근 시 프로그램 자체가 다운로드 되는 방식이다. 이런 구조는 프로그램 전체가 클라이언트 쪽으로 다운로드 되어서 실행되는 방식이기 때문에 보안에 문제점이 발생할 수 있다. 그림 2는 스마트 클라이언트를 적용한 사용된 프로그램 환경을 보여 주는 것이다 [1, 3, 6].

Java web Applications은 JSP를 이용해서 제작하였고 특별한 디자인 없이 결과 값을 테이블 형태로 보여주는 방식을 취했다. 실제로 사용한 환경은 Java를 해석할 수 있는 Tomcat 4.1을 사용했으며 데이터베이스 연결 컴포넌트는 Java Beans와 JDBC를 사용했다. 그림 3은 실험에 사용된 Java Application의 구조이다.

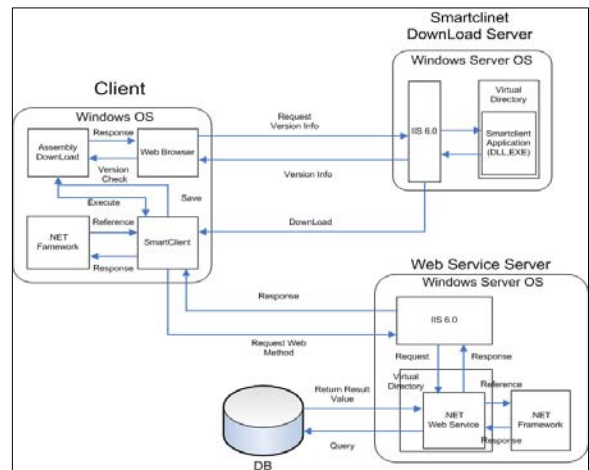


그림 2. 실험에 사용된 X-Internet 구조

실험에 사용된 X-Internet는 크게 3부분으로 이루어져 있다. 일단 첫째로 Smart Client를 제공하는 Smart Client Download Server이며 이는 IIS의 가상 디렉토리를 통해서 제공된다. 둘째로 Smart Client를 사용하는 Client이다. 이는 Smart Client를 관리하는 Assembly Download 폴더 그리고 Smart Client를 실행하기 위한 .NET Framework로 나눌 수 있다. 셋째로 실제에 적용가능하게 핵심적인 루틴을 감추는 Web Service Server가 있다. 이는 데이터베이스와 상호작용하면서 중요한 데이터와 그에 따른 처리를 독립적으로 처리해서 Smart Client에 제공하는 역할을 한다. 시나리오는 사용자가 Smart Client Download Server에 URL로 접근하면 자동적으로 Assembly Download 폴

더에 다운로드 된다. 다시 요청하면 버전을 확인하고 같은 버전이라고 하면 Assembly Download 폴더에서 불러서 실행시키고 버전이 다르면 다시 다운로드한다. 실행 시에는 전반적인 모든 기능은 NET. Framework를 통해서 Local에서 실행되나 중요한 부분 즉, Web Service부분은 Web Service Server에 요청 및 결과를 반환받는다.

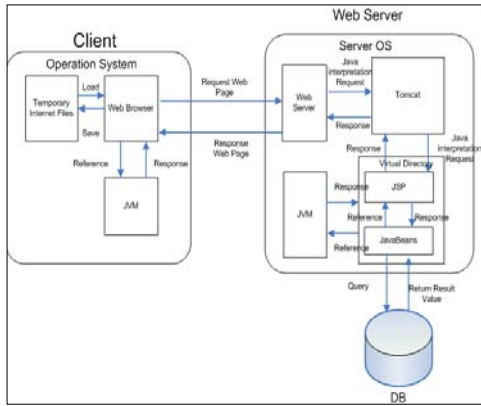


그림 3. 실험에 사용된 Java Application 구조

Java Application의 구조는 현재의 보편적인 웹의 구조와 동일하게 웹 브라우저를 사용하는 클라이언트와 페이지를 생성하고 제공하는 Web Server로 나눌 수 있다. 실험에서는 되도록 단순한 구조를 택했으며 Tomcat Web Server를 통해서 JSP를 해석하고 Java Beans를 통해서 데이터베이스와 연동한다. 보편적인 웹에서 Java Sript 사용하기에 JVM이 Server에서 뿐만 아니라 Client에도 설치했다.

4.2 데이터베이스의 구성

데이터베이스는 MS-SQL 2000 Server로 동일한 물리적, 논리적 환경을 사용했으며 위에서 설명한 것과 같이 TPC에서 제공하는 TPC-H 모델링 구조를 사용했다. 단 Nation과 Region의 테이블은 제외 시켰으며 SF=1로 설정하였다. 그림 4는 실험에 사용한 TPC-H의

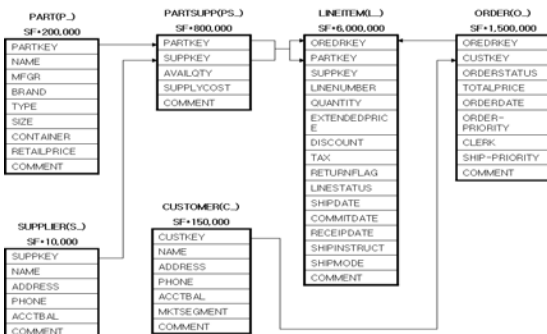


그림 4. TPC-H를 수정한 데이터베이스 모델링

구조이다. 질의문은 TPC-H 모델링 수정에 맞게 수정했으며 실험에서는 그중 5개만 선별하였다[2].

5. 성능 평가

5.1 서버 및 클라이언트 메모리 사용량

Smart Client는 서버 Data Set이라는 객체에 데이터를 저장하였다가 클라이언트에 보내기 때문에 Java 웹 어플리케이션보다 높은 수치가 나왔지만 커다란 차이를 보이지 않는다. 그에 비해서 클라이언트 쪽에서는 데이터가 증가할수록 큰 차이를 보이고 있다. 이는 아래에 나온 패킷 량과도 관련이 있어서 XML과 HTML의 구조적인 차이 때문이다 [5].

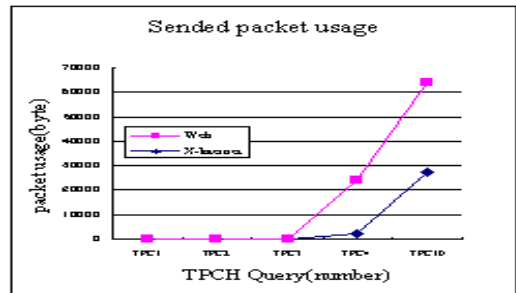


그림 5. Client의 메모리 사용량

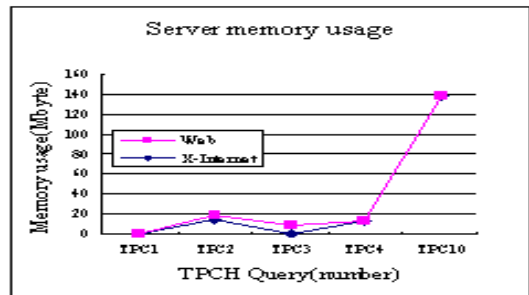


그림 6. 서버의 메모리 사용량

5.2 보낸 패킷 량과 받은 패킷 량(통신량)

보낸 패킷 량과 받은 패킷 량은 실험결과는 데이터양이 증가함에 따라 선형으로 증가하며 웹 쪽이 더 큰 기울기로 증가하고 있다. 결국 같은 데이터를 가지고도 보낸 패킷 량과 받은 패킷 량의 차이가 증가하는 것은 역시 XML과 HTML의 구조적 차이라고 볼 수 있다.

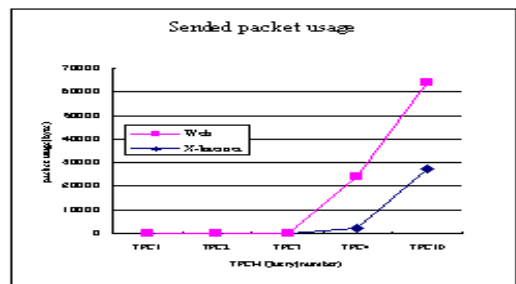


그림 7. 보낸 패킷 량

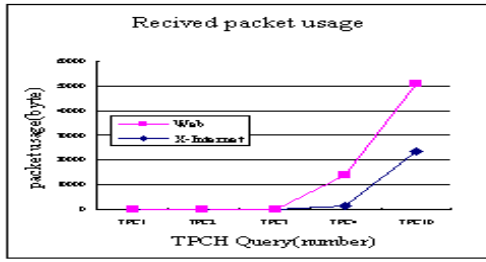


그림 8. 받은 패킷 량

5.3 작업시간

작업시간은 질의에 대한 데이터베이스의 작업 시간과 처리된 데이터를 클라이언트에 전송하는 시간으로 나눌 수 있다. 데이터베이스가 동일하므로 처리시간의 차이는 전송시간의 차이라고 할 수 있으며, 이 역시 위와 동일한 결과이다.

5.4 CPU의 사용량

CPU의 사용량은 크게 차이가 나는 것을 볼 수 있다. X축이 시간이고 Y축이 CPU 사용으로 TPC4와 TPC10의 경우이다. 이것은 첫 번째 데이터의 양이 이런 결과를 가져오고 있으며 두 번째로 웹 브라우저를 사용하는 Java Application에 비해서 자체 윈도우 폼을 사용하는 경우가 더 안정적인 CPU사용량을 보인다. 이것은 HTML이 XML에 비해서 Well-Formed 되지 못해서 브라우저의 파싱에 많은 리소스가 소비되는 것을 나타낸다. Java Application에서 CPU 사용량이 갑자기 급격하게 올라가는 부분이 브라우저에 로딩 되는 것이라는 부분에서 이점을 잘 설명해 주고 있다.

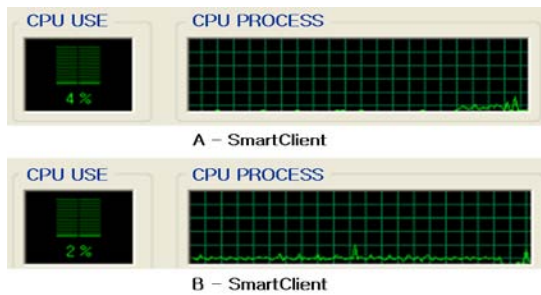


그림 9. SmartClient CPU 사용량

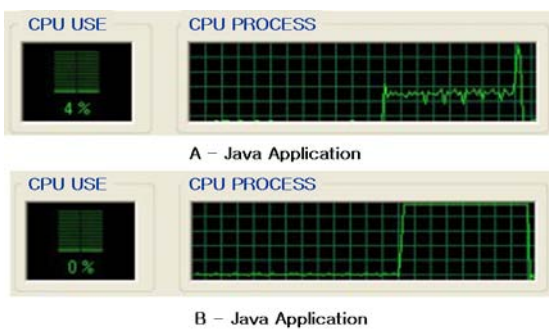


그림 10. Java Application CPU 사용량

6. 결론

성능에 가장 큰 영향을 미친 것은 통신량의 차이인 것으로 나타났는데, 그것은 XML과 HTML의 차이라고도 설명할 수 있다. 송수신 데이터 패킷 량의 증가가 핵심적인 역할을 한다는 것을 밝혔다. X-Internet 기반의 Application은 Web Application보다는 패킷 량이 적다는 것을 의미하며 대용량 통신에 좀 더 나은 구조를 지닌다고 볼 수 있다. 또한 자체 인터페이스를 가지고 있는 X-Internet기반의 Application은 인터넷 브라우저보다 더 나은 인터페이스와 성능 그리고 보다 적은 리소스 사용한다는 것을 확인할 수 있었다. 현재 Web Application이 가지고 있는 장점과 단점 중에서 장점은 그대로 유지하면서 단점인 유저인터페이스의 미비 그리고 패킷 량이 많다는 것을 해결할 수 있는 차세대 아키텍처로서 X-Internet은 그 가능성을 입증하였다.

참고문헌

- [1] Banerjee, Ashish, et all. C# Web Service. 2002.
- [2] <http://www.tpc.org>.
- [3] J. Lapalme, E. Aboulhamid, G. Nicolescu, L. Charest, F. Boyer, J. David, G. Bois, ".NET Framework - A Solution for the Next Generation Tools for System-Level Modeling and Simulation," DATE (2004) pp. 732-733
- [4] M. Löwis, P. Tröger, Microsoft's .NET. The Industrial Information Technology Handbook 2005: 1-12
- [5] M. Torrens, B. Faltings, P. Pu, "SmartClients: Constraint Satisfaction as a Paradigm for Scalable Intelligent Information Systems," An International Journal of Constraints (2002) 7(1), pp. 49-69
- [6] <http://www.snc.sapmi.net/References/>
- [7] S. Yang, J. Khan, "Open Standard Based Visualization of Complex Internet Computing Systems," in Proc. IC Computational Science, 2004. pp. 1008-1012.
- [8] M. Jordan, G. Czajkowski, K. Kouklinski, G. Skinner, "Extending a J2EETM Server with Dynamic and Flexible Resource Management," in Proc. Middleware (2004) pp. 439-458.