

자바 이동 에이전트를 이용한 웹 서버 부하 모니터링

정 경 진*, 박 홍 진, 김 영 찬
중앙대학교 컴퓨터 공학과

The Load Monitoring of Web Server using Java Mobile Agent

Kyung-Jin Jyung*, Hong-Jin Park, Young-Chan Kim
Dept. of Computer Science and Engineering, Chung-Ang Univ

요약

인터넷의 발전에 따라 웹 서버의 중요성이 크게 부각되고 있으며 짧은 시간안에 웹 서버 사용자의 급증으로 인해 서버의 부하가 크게 증가하는 현상이 발생하기도 한다. 이러한 이유로 웹 서버의 부하 모니터링 기술은 서버의 성능 관리에 중요한 요소가 된다. 웹 서버 부하 모니터링을 위해 기존의 클라이언트-서버 패러다임을 사용하는 것보다 스스로 시스템간에 이동하며 작업을 수행하는 자바 이동 에이전트를 사용한 방법이 보다 효율적이다. 이동 에이전트는 스스로 웹 서버를 이동하며 부하를 측정하고 전체 웹 서버의 부하 정보를 한 노드로 전송한다. 본 논문에서는 네트워크 전송량을 줄이고 웹 서버에 추가적인 부하 부담을 주지 않기 위해 자바 이동 에이전트를 이용한 웹 서버 부하 모니터링 시스템을 설계하고 구현하였다.

1. 서론

인터넷의 급격한 발전에 따라 전세계의 사용자들에게 웹 서비스를 제공하는 웹 서버의 중요성이 부각되고 있다. 최근에는 부하 분산을 위해 미러링 웹 서버들을 사용하기도 하며, 이것은 인터넷상에서 지리적으로 분산되어 있어서 웹 서버의 성능 관리에 어려움이 많다. 이러한 분산된 여러 웹 서버의 부하 모니터링 기술은 웹 서버 성능 관리의 중요한 요소로 인식되고 있다.

웹 서버는 항상 사용자들에게 웹 서비스를 올바르게 제공해야 하며 서비스 전송 속도가 사용자들에게는 가장 중요한 요소이다. 웹 서버가 사용지수와 전송량, 프로세스 수, 메모리 사용량의 급증으로 인해 그 부하가 증가하거나, 시스템 자체에 문제가 발생하게 되면 웹 서비스를 올바르게 제공할 수 없으며 이는 많은 사용자들에게 큰 오류로 인식되어 웹 서버 개발비용을 낭비하는 결과를 초래한다[1]. 웹 서버의 부하 모니터링은 먼저 서버가 항상 올바르게 작동하고 있는지를 지속적으로 검사해야 하며, 웹 서버에 접속하는 사용자와 서버가 전송해야 할 자료의 양이 급증하는 것을 감지해서 서버가 효율적으로 운영될 수 있도록 최적화 작업을 수행해야 한다. 정확한 부하 정보는 왼쪽 서버의 부하를 다른 미러링 웹 서버로 효율적으로 분산시키는데 사용될 수 있으며, 웹 서버 모니터링 기술은 시스템 자체에서 발생한 오류를 빠른 시간내에 검출해서 문제를 해결할 수 있다. 웹 서버의 부하 정보를 DB에 저장해 놓은 Historical Data를 바탕으로 추후 시스템 운영 상황을 예측할 수 있으며, 성능 분석 및 시스템 확장의 근거 자료로 활용할 수 있다.

본 연구에서는 기존의 클라이언트-서버 패러다임을 사용한 웹 서버 모니터링의 단점을 해결하기 위해 자바 이동 에이전트를 사용해서 웹 서버의 부하를 효율적으로 모니터링 할 수 있는 시스템을 개발한다

본 논문의 구성은 다음과 같다. 2장에서는 이동 에이전트 기술에 대해 기술하며, 3장에서 이동 에이전트를 이용한 모니터링 시스템의 설계와 구현에 대해 설명한다. 마지막으로 4장에서 평가와 결론에 대해 기술한다.

2. 기반 연구

2.1 기존의 분산 컴퓨팅

지금까지 웹 서버의 부하를 모니터링 하기 위해서 클라이언트-서버 패러다임을 사용하였다. 분산 컴퓨팅 방법인 클라이언트-서버 패러다임은 중앙집중식 방법이며 모든 웹 서버마다 부하를 측정하는 프로그램을 설치해서 웹 서버가 주기적으로 클라이언트의 요청을 받아서 부하 정보를 전송해 주어야 한다[2]. 그러므로 부하를 모니터링 하려는 사용자가 한 노드에서 모니터링 방법에 변경을 가할 수 없으며, 지리적으로 분산된 웹 서버의 관리는 불가능하다 또한 웹 서버에서 계속 실행되고 있는 모니터링 프로그램은 웹 서비스에 치중하는 서버에 추가적인 부담을 줄 수 있으며, 클라이언트와 서버간에 전송되는 메시지는 네트워크 트래픽을 증가시킨다. 클라이언트와 웹 서버는 지속적으로 네트워크 연결이 성립되어 있어야 하며 여러 가지 환경 변화에 자동적으로 대처하기 어렵다[3][4].

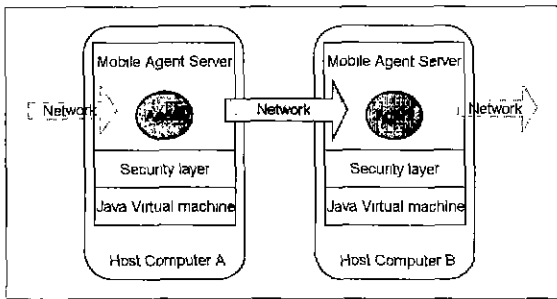
이러한 클라이언트-서버 프로그램의 단점을 해결하기 위해 이동 에이전트가 직접 이동하면서 원하는 정보를 수집해야 한다.

2.2 자바 이동 에이전트

자바 이동 에이전트는 네트워크를 통해서 스스로 자신을 다른 호스트로 이동시켜서 실행할 수 있는 프로그램이다. 처음에 생성된 실행 환경에서 실행되기 원하는 시스템으로 자신의 상태정보(state)와 코드를 이동시킬 수 있다. 상태정보는 수행해야 할 작업이 무엇

인지를 결정하는 속성값(attribute value)을 뜻하며, 코드는 에이전트의 실행에 필수적인 클래스 코드이다. 이동 에이전트는 네트워크로 연결된 여러 시스템들이 낮은 대역폭과 높은 지연시간을 가지는 상황에서도 효율적으로 작업을 수행할 수 있다. 또한 이동 에이전트간에 서로 통신을 수행함으로써 다른 이동 에이전트의 자원을 사용할 수 있으며, 일반적으로 작업을 모두 수행한 후에는 처음에 에이전트가 생성된 시스템으로 귀환하거나 소멸된다. 이러한 이동 에이전트를 사용함으로써 기존의 클라이언트-서버 패러다임보다 분산 애플리케이션을 작성하는데 있어서 보다 편리하고 효율적이고 안전한 프로그래밍 패러다임을 제공한다[5]

여러 시스템에 요청을 해서 응답을 받는 방법 대신에 이동 에이전트가 직접 네트워크에 연결된 시스템으로 이동해서 그 시스템의 필요한 자원을 사용해서 작업을 수행하는 효율적인 방법을 사용한다. 이동 에이전트가 이동할 때만 네트워크 연결이 필요하며 이동 에이전트의 실행 위치, 통신 채널에 대해서는 사용자가 관여할 필요가 없기 때문에 분산 애플리케이션의 개발, 테스트, 배치가 쉽다(그림 1).



[그림 1] 자바 이동 에이전트

2.3 자바 이동 에이전트의 장점

자바 이동 에이전트를 사용함으로써 다음과 같은 장점을 가진다 [6].

- 네트워크 부하의 감소 : 다른 시스템들과 많은 상호작용이 필요한 분산 시스템에서는 네트워크 트래픽이 증가하지만, 이동 에이전트는 목적지로 이동해서 그 시스템 내부에서 작업을 수행하기 때문에 전송해야 할 자료의 양이 줄어든다.
- 네트워크 지연의 해결 : 환경의 변화에 실시간으로 반응해야만 하는 중요한 실시간 시스템에서는 네트워크를 통한 작업은 중대한 지연시간이 발생하지만, 이동 에이전트가 이동해서 중앙 관리자의 지시를 직접 수행할 수 있다.
- 비동기적이고 자율적인 실행 : 이동 에이전트는 이동시에만 네트워크 연결이 필요하고 지속적으로 네트워크 연결을 유지할 필요가 없으며, 이동한 시스템 내부에서 비동기적이고 자율적으로 작업을 수행한다.
- 프로토콜 캡슐화 : 분산 시스템간에 자료를 교환할 때 자료의 입출력에 관계된 프로토콜을 구성해야 하지만, 이동 에이전트는 독자적인 프로토콜을 사용하여 원격 호스트로 이동한다.
- 동적 적응 : 실행 환경을 감지하고 변경사항에 대해 자율적으로 반응할 수 있다.
- 이질성 : 인터넷을 통한 네트워크 컴퓨팅은 근본적으로 이질성을 갖는다.
- 결함허용 : 시스템에 문제가 발생한 부직절한 상황에서도 결함허용 분산 시스템을 구축할 수 있다

3. 이동 에이전트를 이용한 웹 서버 부하 모니터링 시스템

본 장에서는 이동 에이전트를 사용한 모니터링의 장점과 자바 이동 에이전트를 사용한 웹 서버 부하 모니터링 시스템의 설계, 구현에 대해 설명한다.

3.1 이동 에이전트 기반의 모니터링

웹 서버의 부하를 모니터링 하기 위해 기존의 클라이언트-서버 패러다임을 사용하는 것보다 위와 같은 특성과 장점을 가진 이동 에이전트를 사용하는 것이 훨씬 편리하고 효율적이다

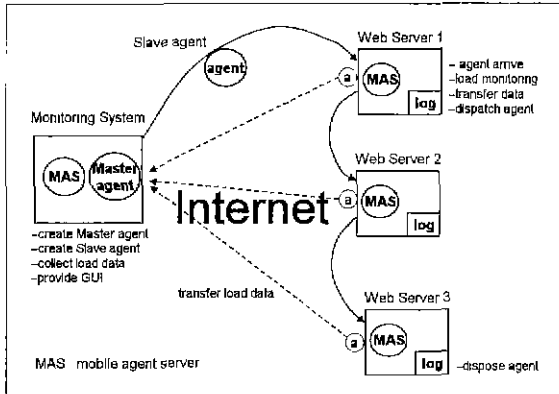
이동 에이전트를 사용한다면 모니터링의 대상이 되는 모든 웹 서버에 별도의 프로그램을 설치하지 않고도 모든 웹 서버의 부하 정보를 한 시스템에 수집할 수 있다. 이동 에이전트의 실행 코드가 네트워크 상에서 웹 서버로 이동해서 내부적으로 부하를 측정하고 그 결과값을 모니터링 서버에 전송한 후에 다음 웹 서버로 이동한다. 이동 에이전트는 부하 측정에 필요한 핵심적인 코드만 소유하고 있기 때문에 아주 작은 크기로 프로그래밍 될 수 있으며, 이동시에 필요한 전체 네트워크 전송량이 적다. 또한 웹 서비스 제공에 치중하는 웹 서버에서 계속 실행되지 않고 에이전트가 이동했을 때 에민 잠시 실행되기 때문에 웹 서버에 추가적인 부하 부담을 주지 않는다. 웹 서버의 부하를 측정하기 위해서는 큰 용량의 로그파일을 필요로 하지만 로그 파일을 네트워크를 통해 전송하지 않고 에이전트가 직접 이동해서 시스템 내부적으로 로그 파일을 빠른 시간 안에 분석해서 사용자수와 전송량을 계산해 낼 수 있다.

이동 에이전트는 독자적인 프로토콜로서 ATP(Agent Transfer Protocol)를 사용해서 이동하기 때문에 모니터링 하려는 웹 서버의 위치에 상관없이 작업을 수행할 수 있으며, 여러 서버를 순차적으로 이동해서 작업을 수행하기 때문에 웹 서버의 수에 크게 영향을 받지 않는다. 특히 인터넷망에서 지리적으로 분산되어있는 있는 미러링 웹 서버의 부하를 한 노드에서 모니터링 할 수 있으며, 측정 한 부하 정보는 미러링 웹 서버의 부하 분산 기술에 사용될 수 있다. 이와 같은 이유로 인해 기존의 클라이언트-서버 프로그램, 보다 이동 에이전트를 사용한 모니터링이 더 효율적이다.

3.2 부하 모니터링 시스템의 설계

인터넷상에 지리적으로 분산되어 있는 여러 웹 서버의 부하를 효율적으로 모니터링 하기 위해서 이동 에이전트를 사용한 분산 컴퓨팅 방법을 사용해야 한다. 전체 구성은 모니터링의 대상이 되는 하나 이상의 웹 서버들과, 사용자가 웹 서버들의 부하 정보를 볼 수 있는 모니터링 시스템으로 이루어진다. 각각의 시스템에는 MAS(Mobile Agent Server)가 있다. MAS(Mobile Agent Server)는 각각의 호스트에서 이동 에이전트가 실행될 수 있는 환경을 제공하며 이동 에이전트의 생성, 이동, 소멸과 같은 기능이 수행될 수 있다. 에이전트는 이동 방법에 관계된 순회(Itinerary) 패턴과 작업 수행과 관계된 주-종속(Master-Slave) 패턴을 결합하여 설계한다. 모니터링 시스템의 MAS에서 Master 에이전트가 실행되며, Master 에이전트는 실제 웹 서버로 이동할 Slave 에이전트를 생성한다. Slave 에이전트는 스스로 자신을 첫 번째 웹 서버로 이동시켜서 웹 서버의 로그 파일을 분석하고, 시스템 자체의 부하를 측정 한 후에, 부하 정보를 Master 에이전트에게 전송한다. 부하 정보를 전송한 후에 다음 순서의 웹 서버로 이동해서 같은 작업을 계속 수행한다.

Master 에이전트는 모니터링 시스템에서 계속 실행하면서 GUI 기능을 제공하며 Slave 에이전트로부터 부하 정보를 받아서 사용자에게 전체 웹 서버들의 부하 정보를 보여주고, 주기적으로 새로운 Slave 에이전트를 생성시킨다. 이와 같은 이동 에이전트의 기능은 [그림 2]에 나타나 있다.

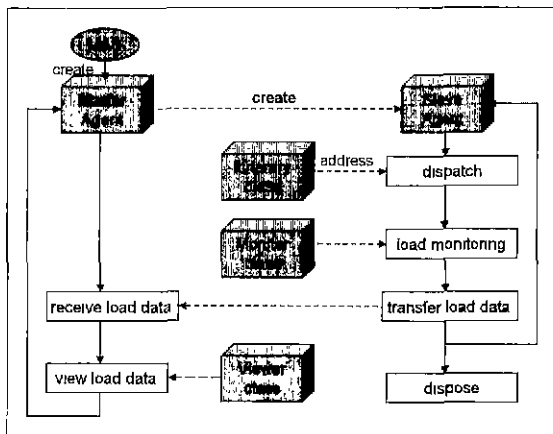


[그림 2] 순회 패턴을 이용한 Master, Slave 에이전트의 기능

3.3 웹 서버 부하 모니터링 시스템의 구현

자바 이동 에이전트를 이용한 웹 서버의 부하 모니터링 시스템은 Master 에이전트와 Slave 에이전트로 구성된다. Master 에이전트는 Slave 에이전트로부터 웹 서버의 부하 정보를 받아서 전체 웹 서버의 부하 정보를 Viewer 클래스를 사용해서 화면상에 보여주는 GUI 기능을 수행하며, 일정한 시간간격으로 Slave 에이전트를 새로 생성시켜서 모니터링 작업을 계속 수행한다. Master 에이전트에 의해 생성된 Slave 에이전트는 스스로 그 자신을 웹 서버로 이동시킨다. Slave 에이전트가 이동해야 할 웹 서버의 주소를 Itinerary 클래스에서 제공받는다. Itinerary 클래스는 모니터링의 대상이 되는 웹 서버들의 주소를 가지고 있으며, Master 에이전트가 Slave 에이전트를 생성시킬 때 매개변수로 전달한다. 또한 Itinerary 클래스는 에이전트를 다음 주소로 이동시키고, 이동할 웹 서버에 문제가 있을 때 그 다음 웹 서버로 이동시키는 기능을 제공한다([그림 3])

웹 서버로 이동한 Slave 에이전트는 Monitor 클래스를 사용해서 웹 서버의 로그를 분석하고, 시스템 자체의 부하를 측정한다. 모든 부하 정보를 Master 에이전트로 전송한 후에 다음 웹 서버로 이동하며, 더 이상 이동할 웹 서버가 없으면 모든 작업을 수행한 후에 스스로 소멸된다. Monitor 클래스는 웹 서버의 최근의 사용자 수, 최근의 전송량, 최근의 에러 발생 횟수, 1.5·15분간의 평균 직업



[그림 3] 부하 모니터링 시스템의 클래스 구성

수, Swap 메모리 양, Free 메모리 양, 디스크 사용률, 사용한 디스크 양, 사용가능한 디스크 양을 측정한다. 모니터링 시스템에서 전체 웹 서버의 부하 정보를 [그림 4]와 같이 디스플레이 한다

server	pid	user	transfer volume	err or	trnsErr or	15m load	15m load	swap memory	free memory	disk I/O	used disk	avail disk	time
hube	1	1	58731	0	0.05	0.47	0.78	119646	1378	0.0	1175911	2561329	11:23:07
schwaika	2	1	78500	1	0.12	0.15	0.14	315936	46272	0.0	2438898	2520952	11:19:59
op4a	1	3	437149	2	0.04	0.04	0.05	886704	181527	0.0	18172548	15618517	11:18:51
ms			bytes		jobs	jobs	jobs	Kbytes	Kbytes	%	bytes	bytes	

[그림 4] 전체 웹 서버의 부하 정보 디스플레이

4. 평가 및 결론

각 웹 서버에서 실행되는 프로그램의 크기는 Slave 에이전트가 3.2KB, Monitor 클래스가 4.3KB, Itinerary 클래스가 2KB로서 전체 크기는 9.5KB이다. 9.5KB의 클래스가 웹 서버에서 부하 측정이 필요한 때만 실행되고 그 이후에는 소멸되므로 항상 메모리와 CPU를 점유하고 있는 클라이언트-서버 프로그램보다 웹 서버에 추가적인 부하 부담을 주지 않는다. 그리고 모니터링 시스템을 실행시키고 첫 번째 에이전트가 이동할 때만 전체 클래스가 이동하며 그 이후로는 Slave 에이전트만 이동하기 때문에 처음에만 9.5KB가 이동하며 그 이후로는 3.2KB만 이동하게 된다. 그러므로 네트워크 사용량이 아주 적다

본 연구는 인터넷에서 지리적으로 분산되어 있는 여러 웹 서버의 부하를 자바 이동 에이전트를 사용하여 효율적으로 모니터링 하는데 그 목적이 있다. 자바 이동 에이전트를 사용함으로써 관련 프로그램을 설치하지 않고서도 한 노드에서 여러 웹 서버의 부하를 모니터링 할 수 있으며, 모니터링 방법을 변경 할 수 있다. 기존의 클라이언트-서버 패러다임에서는 구현할 수 없었던 새로운 메커니즘을 제공한다.

앞으로 자바 이동 에이전트를 이용한 웹 서버 부하 모니터링 기술의 기반으로 웹 서버에 집중되는 부하를 다른 웹 서버로 효율적으로 분산시키는 미러링 웹 서버의 부하 분산에 대한 연구에 주력할 것이다

참고문헌

[1] Web Server Monitoring, <http://www.freshtech.com>
 [2] M Caprini, R. Nacach, Z Qian, "Java Mobile Agent for monitoring task . evaluation report", <http://attdoc.cern.ch/Atlas/Notes/078/Note078-1.html>, 1998.
 [3] MARS(Monitoring Application for Resources and Servers), <http://www.altara.org/mars.html>
 [4] E. Anderson, D. Patterson, "Extensible, Scalable Monitoring for Clusters of Computers", Proceedings of 1997 LISA Conference, 1997.
 [5] R. Gray, D. Kotz, S. Nog, D. Rus, G Cybenko, "Mobile agents for mobile computing", Technical Report PCS-TR96-285, 1996.
 [6] D. B. Lange, Mitsuru Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison Wesley press, 1998.