1. 캐시 서버의 이용

여러 대의 같은 기능을 가진 서버를 설치하는 것이 아닌 다른 방법으로 부하 분산을 하는 방법도 있다.

이것은 데이터 베이스 서버, 웹 서버 등 역할에 따라 나누는 방법으로 그 중 하나가 **캐시 서버**를 사용하는 방법이다.

- **캐시 서버**

  ➔ 캐시 서버는 프록시라는 구조를 사용하여 데이터를 캐시에 저장하는 서버이다.

  ➔ 프록시는 웹 서버와 클라이언트 사이에 들어가서 웹 서버에 대한 액세스 동작을 중개하는 역할을 하는데, 웹 서버에서 받은 데이터를 디스크에 저장해두고, 웹 서버를 대신하여 데이터를 클라이언트에 반송하는 기능을 가지고 있다.

  ■ 이것을 캐시라고 부르며, 캐시 서버는 이 기능을 이용한다.

  ➔ 캐시 서버는 보존해 둥 데이터를 읽어서 클라이언트에게 송신만 하므로 웹 서버보다 빨리 데이터를 송신할 수 있다.

캐시에 데이터를 저장한 후 웹 서버 측에서 데이터가 변경되면 캐시의 데이터를 사용할 수 없게 된다. 따라서 언제든지 캐시의 데이터를 이용할 수 있는 것은 아니다.

그러나 액세스 동작의 일정 부분은 캐시 서버로 처리할 수 있기 때문에 얼마라도 고속화할 수 있으면 전체 성능이 향상된다고 생각하는 것이다.

2. 캐시 서버의 콘텐츠 관리

- **캐시 서버의 동작**

  ➔ 캐시 서버를 사용할 때는 부하 분산 장치와 마찬가지로 캐시 서버를 웹 서버 대신 DNS서버에 등록한다.

  ➔ 그러면 사용자는 캐시 서버에 HTTP의 리퀘스트 메시지를 보낼 것이고, 캐시 서버가 메시지를 받아 수신 동작을 한다. 이때 수신 동작은 웹 서버의 수신 동작과 같다.

  ■ 접속을 기다리는 패킷을 만들고, 여기에 사용자가 접속하면 사용자가 보낸 리퀘스트 메시지를 받는다.

  ■ 이후 메시지의 내용을 조사하고, 데이터가 자신의 캐시에 저장되었는지 조사한다.

- **캐시에 데이터가 저장된 경우**

  ➔ 먼저 사용자로부터 도착한 리퀘스트 메시지가 캐시에 저장 되었는지 확인한다.

  ➔ 그리고 웹 서버 측에서 데이터가 변경되었는지 조사하기 위한 'If-Modified-Since'라는 헤더 필드를 추가하여 웹 서버에 전송한다.

  ➔ 웹 서버는 'If-Modified-Since'헤더 필드의 값과 페이지 데이터의 최종 갱신 일시를 비교하여 변경이 없으면 변경이 없는 것을 나타내는 응답 메시지를 반송한다.

  ■ 이때는 최종 갱신 일시를 조사하는 것으로 끝나므로 페이지 데이터를 반송하는 것보다 부담이 적어진다.

  ➔ 캐시에 저장된 데이터가 최신 데이터와 같은 것이 확인되면 캐시 서버는 데이터를 추출하여 사용자에게 보낸다.

  ➔ 데이터가 변경된 경우

  ➔ "캐시에 데이터가 없는 경우"와 같은데 웹 서버로부터 받은 최신 데이터에 'Via'헤더를 부가하여 사용자에게 전송하고 데이터를 캐시에 저장한다.

- **캐시에 데이터가 없는 경우**

  ➔ 이 경우 캐시 서버는 리퀘스트 메시지에 캐시 서버를 경유한 것을 나타내는 'Via'라는 헤더 필드를 추가하여 웹 서버에 리퀘스트를 전송한다.

그림 1. 캐시에 데이터가 저장되어 있지 않은 경우

  ➔ 우선 어느 웹 서버에 리퀘스트 메시지를 전송해야 할지 판단해야 한다.

  ➔ 웹 서버가 한 대 밖에 없으면 웹 서버의 도메인명이나 IP주소를 캐시 서버에 설정해 두고 무조건 거기로 반송하도록 한다.

➔ 그러나 여러 대의 서버일 경우 에는 리퀘스트 메시지의 내용에 따라 전송 대상의 웹 서버를 판단해야 하는 방법이 필요하다.

- 여러 방법 중 대표적인 것은 리퀘스트 매시지의 URI에 쓰여 있는 디렉토리를 보고 판단하는 방법이다.
- URI가 /dir1/이라는 디렉토리였다면 www1.lab.cyber.co.kr에 전송한다.
- URI가 /dir2/이라는 디렉토리였다면 www2.lab.cyber.co.kr에 전송한다
- 이때는 캐시 서버가 클라이언트가 되어 웹 서버와 통신하고 응답 메시지를 받게 되며, 이후 다시 자신이 웹 서버가 되어 클라이언트에게 응답하게 된다.
- 그리고 이때 캐시 서버를 경유한 것을 나타내는 'Via' 헤더 필드를 부가한다.

이렇게 클라이언트와 웹 서버 사이를 중개하는 것이 프록시 서버의 구조이다.

3. 포워드 프록시(클라이언트측)

지금까지 설명한 내용은 프록시 구조를 웹 서버 측에 두고 캐시 기능을 이용하는 것이지만, 클라이언트 측에 캐시 서버를 두는 방법도 있다.

➔ 사실 프록시 구조는 원래 클라이언트 측에 두는 방법에서 시작되었다.

➔ 이 유형이 프록시 원형으로 **포워드 프록시**라고 한다.

- **프록시 구조 도입 이유**

➔ 당시 포워드 프록시의 기능은 캐시를 이용하는 목적과 방화벽을 실현한다는 목적이 있었다.

➔ 방화벽의 목적을 달성하는 현실적인 방법은 인터넷과 사내의 패킷 왕래를 전부 정지시키는 것이다.

➔ 그러나 이럴 경우 인터넷에 대한 액세스도 정지되므로 필요한 것을 통과 시키는 방법을 생각하기 위해 "프록시"구조를 고안하였다.

➔ 프록시 구조를 사용하면 이전에 액세스한 페이지의 경우 사내 LAN에서 프록시에 액세스 하기만 하면 데이터를 받을 수 있으므로 저속 회선에서 매우 빨라질 수 있다.

➔ 또한 프록시는 리퀘스트의 내용을 조사한 후 전송하므로 리퀘스트의 내용에 따라 액세스가 가능한지 판단할 수 있다.

- 위험한 사이트나, 작업과 관계없는 사이트에 대한 액세스는 금지할 수 있다.

- **프록시 서버**

➔ 포워드 프록시를 사용할 경우 브라우저의 설정 화면에 준비되어 있는 **프록시 서버**라는 항목에 포워드 프록시의 IP주소를 설정한다.

➔ 이렇게 되면 URL의 내용에 상관없이 리퀘스트를 전부 포워드 프록시에 보낸다.

- 결과 프록시를 설정하지 않으면 URL에서 파일이나 프로그램의 경로명의 일부를 추출하여 URI부분에 기록하지만 프록시를 설정하면 http://...라는 URL을 그대로 리퀘스트의 URL에 기록한다.

4. 리버스 프록시(서버측)

- **포워드 프록시 단점 :**

➔ 브라우저에 대한 설정이 꼭 필요하다.

➔ 브라우저에 설정하는 수고나 장애의 원인 뿐만 아니라 인터넷에 공개하는 웹 서버는 누가 액세스 하는지 알 수 없다.

이에 따라 브라우저에 프록시를 설정하지 않아도 사용할 수 있도록 **리버스 프록시**를 사용한다.

➔ 리퀘스트 메시지의 URI에 쓰여 있는 디렉토리 명과 전송 대상의 웹 서버를 대응시켜서 URI부분에 http://... 라고 쓰여 있지 않은 보통의 리퀘스트 메시지를 전송할 수 있도록 한다.

➔ 이것이 서버 측에 설치하는 캐시 서버에 채택하고 있는 방식으로 **리버스 프록시**라고 부른다.

5. 트랜스페어런트 프록시(클라이언트 – 트랜스페어런트 – 서버)

- **캐시 서버에서 전송 대상을 판단하는 방법**

  ➔ 패킷의 맨 앞에 있는 IP헤더에는 수신처 IP주소가 기록되어 있으므로 이것을 조사하면 액세스 대상 웹 서버가 어디 있는지 알 수 있는데, 이 방법을 **트랜스페어런트 프록시**라고 한다.

  ➔ 이 방법은 포워드 프록시처럼 설정할 필요가 없고, 전송 대상을 캐시 서버에 설정할 필요도 없으므로 어느 웹 서버에나 전송할 수 있다.

트랜스페어런트 프록시는 포워드 프록시와 리버스 프록시의 좋은 점만 모은 형태의 구조이지만, 여기에 리퀘스트 메시지를 건네 주는 방법에 주의해야 한다.

- **트랜스페어런트 프록시 사용시 주의할 점**

  ➔ 브라우저에 설정하지 않기 때문에 브라우저는 웹 서버에 리 퀘스트 메시지를 보낸다.

  ➔ 리버스 프록시처럼 DNS서버에 등록하는 방법이라면 리퀘스트 메시지가 프록시에 도착하겠지만 트랜스페어런트 프록시는 DNS에 등록하는 것도 없다.

  ➔ 이런 경우라면 리퀘스트 메시지는 브라우저에서 웹 서버로 흘러갈 뿐 트랜스페어런트 프록시에 도착하지 않게 된다.

- **트랜스페어런트 프록시에 메시지 전달 방법**

  ➔ 위의 문제를 해결하기 위해 브라우저에서 웹 서버로 리퀘스트 메시지가 흘러가는 길에 트랜스페어런트 프록시를 설치한다.

  ➔ 이후 메시지가 트랜스페어런트 프록시를 통과할 때 그것을 가로챈다.

  ➔ 편법 같지만 이런 방식으로 리퀘스트 메시지가 트랜스페어런트 프록시에 도착하고 웹 서버에 전송할 수 있다.

  ➔ 리퀘스트 메시지가 흐르는 길이 많을 경우 길이 한 개로 수렴하는 형태로 네트워크를 만들고, 그곳에 트랜스페어런트 프록시를 설치하는 것이 보통이다.

■ 인터넷에 연결하는 액세스 회선 부분이 이런 형태이므로 여기에 설치해도 된다.

트랜스페어런트 프록시를 사용하면 사용자도 프록시의 존재를 알아차릴 필요가 없고, 캐시를 이용한다는 측면에서 비중이 높아지고 있다.

- **트랜스페어런트 프록시 활용 목적**

  ➔ **검열 및 모니터:** 특정 사이트 접근을 막을 수 있고, 제한된 콘텐츠 액세스 시도를 기록확인 할 수 있다.

  ➔ **대역폭 사용량 절약:** 100대로 받기보다 1대로 다운받고 그 저장된 데이터를 나머지가 받으므로 속도가 높아진다.

  ➔ **사용자 인증:** 요청 트래픽 저장이 가능 및 추적 가능하다.