

# iSCSI 기반 네트워크 환경에서의 효율적인 캐싱 기법

조국표, 임효택  
동서대학교 컴퓨터공학과  
{ d7003148, htlim }@dongseo.ac.kr

## Efficient Caching Technique in iSCSI-based Network

Kuk-Pyo Cho, Hyo-Taek Lim  
Dept. of Computer Engineering, Dongseo University

### 요 약

iSCSI 기반의 원격 스토리지 시스템에 대한 관심이 증가하면서 이에 대한 연구가 활발히 진행되고 있다. IP 네트워크를 사용하여 통신하는 iSCSI 는 기존의 IP 네트워크의 많은 문제점을 가지고 있다. IP 네트워크를 사용하는 웹 역시 많은 문제점을 가지고 있으며 이를 해결하기 위해 프락시라는 메커니즘을 사용하였다. 효율적인 네트워크 대역폭 사용을 위해 설계된 프록시는 응답속도를 줄이는데 가장 효과적으로 많이 채택되었다. 그러나, 캐시에 저장된 데이터를 사용자에게 제공하기 때문에 이미 캐시되어진 자료가 웹 서버의 최신 자료와 다를 가능성이 있으며 또한 캐시되어진 데이터에 대한 유효성을 검사하기 때문에 추가의 지연시간이 발생하게 된다. 본 논문에서는 iSCSI 기반의 네트워크에서 캐시되어진 데이터에 대한 유효성 검사 없이 항상 fresh 한 상태의 캐시 데이터를 사용자에게 제공할 수 있는 캐싱 기법에 대한 내용과 이것을 적용한 캐시 스토리지 시스템 설계 및 기존의 원격 스토리지 시스템과의 성능을 비교 분석한 내용을 기술한다.

### 1. 서론

최근 시간과 공간의 제약을 극복하여 정보를 얻고자 하는 사용자의 요구와 더불어 유무선 통신분야의 기술은 급속히 발전하였다. 또한 사용자들은 멀티미디어 데이터와 같은 대용량 데이터의 사용을 원하지만 로컬 디스크에 많은 데이터를 저장할 수 없어 이를 해결하기 위해 많은 연구들이 진행되어 왔다. 저장공간의 제한을 극복하기 위한 기술로서 DAS, NAS, SAN과 같은 원격 스토리지에 대한 기술이 등장하였다. 최근 이러한 기술에 사용되는 프로토콜 중 하나인 iSCSI 가 스토리지 분야에서 가장 큰 관심사로 떠오르고 있다. IP와 SCSI를 결합한 iSCSI는 TCP/IP 네트워크를 통해 뛰어난 전송방식인 SCSI 프로토콜을 사용할 수 있게 해주는 차세대 전송기술이다. iSCSI는 현재의 네트워크를 그대로 활용할 수 있기 때문에 손쉽게 구축할 수 있을 뿐만 아니라 구축비용이 적게 든다는 장점이 있다.

그러나, 인터넷 사용자의 수가 늘어나고 많은 네트워크 자원을 소비하는 음성, 데이터, 동영상과 같은

멀티미디어 데이터가 증가함에 따라 그에 따른 트래픽이 폭발적으로 증가하고 있다. 따라서 IP 네트워크의 속도나 성능에 많은 저하를 가져오게 되었다. iSCSI 또한 IP 네트워크를 사용하기 때문에 요청한 데이터에 대한 응답시간의 지연과 같은 기존의 IP 네트워크의 많은 문제점을 가지고 있다. 본 논문에서는 사용자에게 보다 빠른 응답시간을 제공하기 위한 방법으로 항상 fresh 한 상태의 데이터를 캐시하는 새로운 캐싱 기법을 사용하여 이러한 문제를 해결하고자 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련연구에 대해서 설명하고, 3 장에서는 제안하고자 하는 캐시 스토리지 시스템 구조에 대해 설명한다. 4 장에서는 시스템의 성능을 측정, 분석하며 5 장에서는 결론 및 향후 과제에 대해 설명한다.

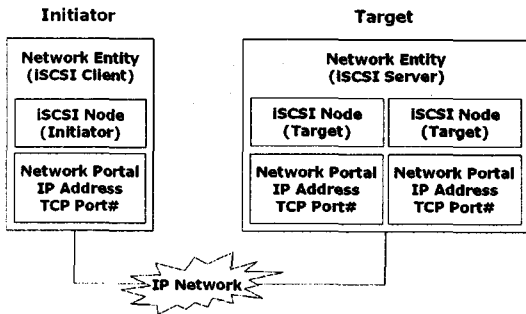
### 2. 관련연구

#### 2.1 iSCSI

iSCSI(Internet Small Computer Systems

Interface)는 IETF의 작업 그룹인 IPS(IP Storage)에서 개발된 프로토콜 중 하나로서 최근 표준으로 지정되었다. IP 네트워크에서 블록 단위로 스토리지 트래픽을 전송해 주는 iSCSI는 스토리지 I/O 블록 데이터를 전송하기 위한 단대단(End-to-End) 프로토콜이다. 보편적으로 사용되는 두 가지 기술인 스토리지의 SCSI 명령과 네트워크의 IP 프로토콜을 기반으로 IP 네트워크로 SCSI 명령을 캡슐화한 후 전송하여 원하는 명령을 수행할 수 있다.

iSCSI는 저장한계를 극복하기 위한 방법으로서 원격지에 있는 스토리지 디스크를 마치 자신의 로컬 디스크처럼 사용할 수 있도록 해준다. 하나 이상의 TCP 연결을 제공하며 address and naming convention 과 session management, error handling security로 구성되어 있고 구성요소들은 각각의 특징을 가지고 있다. 또한, 기존의 IP 네트워크에 구축하기 때문에 많은 비용을 절약할 수 있다. 그리고 하나의 디바이스에서 iSCSI initiator 모듈과 iSCSI target 모듈을 동시에 사용할 수 있기 때문에 사용자와 스토리지 중간에 위치하여 이 둘간의 통신 중계 역할을 할 수 있는 캐시 서버를 구축하는 것이 가능하다. 따라서 사용자에게는 target mode 로서 동작하고 스토리지에게는 initiator mode로 동작할 수 있다. 이러한 특징을 가진 캐시 서버를 구축하여 사용자들에게 요청한 데이터에 대한 보다 빠른 응답시간을 제공함과 동시에 네트워크 트래픽을 줄이고자 한다.



[그림 1] iSCSI의 네트워크 엔티티 모델

2.2 캐시

캐시란 사용했던 데이터에 대한 재사용시 데이터를 빠르게 사용할 수 있도록 데이터들을 임시 저장하는 장소를 의미한다. 많은 분야에서 캐시에 대한 연구가 진행되고 있는데 그 중 현재 웹 캐시에 대한 연구가 활발히 진행되고 있다.

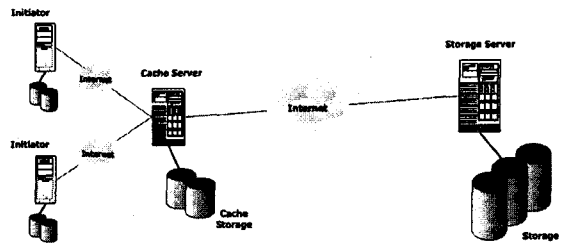
웹 캐싱은 웹의 사용으로 인한 네트워크의 트래픽을 줄이고, 사용자들에게 보다 빠른 응답시간을 제공하여 전체적인 WWW의 성능을 향상시키기 위한 하나의 기술이다. 사용자가 특정한 사이트를 반복적으로 접근하고, 많은 사용자가 같은 사이트에 접근하려는 경향으로부터 착안되었다. 자주 접근되는 정보인 경우 매번 웹 서버에서 제공 받을 필요 없이 한번의 방문

을 통해 가까운 저장소에 데이터를 저장한 후 재사용함으로써 보다 빠른 응답시간을 제공함과 동시에 네트워크 트래픽을 줄여 네트워크의 성능을 향상시킬 수 있다. 본 논문에서 이러한 캐싱 기술을 스토리지 시스템에 적용하여 네트워크 성능을 향상시키고자 한다.

3. iSCSI 기반의 캐시 스토리지 시스템 구조

3.1 시스템 구조

iSCSI 기반의 네트워크 환경에서 네트워크의 성능 향상을 위한 캐시 스토리지 시스템 구조는 [그림 2]와 같다. 그림에서 보는 것과 같이 캐시 스토리지 시스템 구조는 데이터를 저장하고 있는 storage server와 storage server에 저장된 데이터를 사용하는 initiator, 그리고 storage server에 저장된 데이터를 캐시하는 cache server로 구성된다. Initiator와 cache server, cache server와 storage server는 모두 iSCSI 프로토콜을 사용하여 통신하며 initiator와 storage server는 cache server를 통해서만 통신할 수 있다.



[그림 2] iSCSI 기반의 캐시 스토리지 시스템 구조

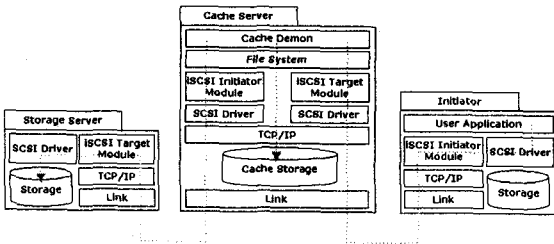
기존의 캐시 시스템은 스토리지 자체에서 데이터를 수정하여 이미 캐시되어진 데이터가 스토리지에서 수정된 데이터와 동기화되어 질 수 없었다. 따라서 캐시되어진 데이터는 항상 fresh한 상태를 유지할 수 없었으며, 또한 사용자에게 의해서 요청된 데이터가 캐시되어져 있다고 하더라도 캐시되어진 데이터에 대한 유효성을 검사하는 추가적인 작업이 필요하였다. 이로 인해 네트워크의 불필요한 트래픽이 발생하여 IP 네트워크의 속도 저하를 가중시켰다.

만일 스토리지의 데이터가 스토리지 자체에서 수정되지 않고 오직 사용자에게 의해서만 수정 될 경우 사용자와 스토리지 중간에서 통신 중계 역할을 하는 cache server는 어떤 데이터들이 사용자에게 의해서 수정되는지 알 수 있다. 사용자에게 의해서 데이터가 수정되면 cache server에서 수정된 최신의 데이터를 스토리지로 복사하여 캐시 스토리지의 데이터를 스토리지 데이터와 동기화할 수 있으므로 캐시 스토리지는 항상 fresh한 상태의 데이터를 캐시할 수 있다. 따라서 본 논문에서는 캐시되어진 데이터가 항상 fresh한 상태를 유지할 수 있게 하기 위해 스토리지 자체에서 데이터를 수정할 수 없고 오직 사용자에게 의해서만 데

이터를 수정할 수 있다고 가정한다.

### 3.2 시스템 계층적 구조

캐시 스토리지 시스템은 iSCSI initiator module 이 동작하는 initiator 와 iSCSI initiator module 과 iSCSI target module 이 함께 동작하는 cache server, iSCSI target module 이 동작하는 storage server 로 구성되며 그 구조는 [그림 3]과 같다.



[그림 3] 시스템 계층적 구조

Cache server 는 initiator 에게는 target mode 로 storage server 에게는 initiator mode 로 동작하며 모두 iSCSI 프로토콜을 사용하여 통신한다. [그림 3] 과 같은 계층적인 구조에 따라 initiator 에서 storage server 로의 명령은 cache server 를 통해서 전달되는데 이때 cache server 의 caching demon 에 의해서 initiator 에서 사용하려고 하는 데이터가 캐시 스토리지에 캐시되어 있는지 검사되어 진다. 만일 캐시 스토리지에 캐시되어져 있다면 initiator 가 캐시되어진 데이터를 사용할 수 있도록 한다. 그러나 만일 데이터가 캐시 스토리지에 캐시되어져 있지 않다면 initiator 가 사용하려는 데이터를 storage server 에서 캐시 스토리지로 복사한 다음 복사한 데이터를 initiator 가 사용할 수 있도록 하는 구조이다.

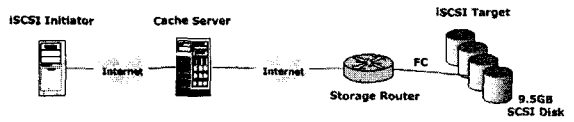
### 3.3 시스템 동작과정

iSCSI 기반의 네트워크 환경에서 네트워크의 성능 향상을 위한 캐싱 기법의 동작과정은 다음과 같다. Storage server 에 대한 initiator 의 명령을 cache server 에서 먼저 받아들여 initiator 에서 수행하고자 하는 명령의 데이터가 캐시 스토리지에 캐시되어져 있는지 검사한다. 이때 캐시되어진 경우와 그렇지 않은 경우에 대해 각각 동작하는 방법에 차이가 있다. 먼저, 데이터가 캐시 스토리지에 캐시되어져 있다면 본 논문에서 제시한 캐싱 기법에 따라 그 데이터는 항상 fresh 한 상태이므로 initiator 가 바로 사용할 수 있도록 한다. 이때 만일 initiator 에 의해 데이터가 수정된다면 수정된 데이터에 대한 명령의 실행이 종료된 후 수정된 데이터를 storage server 에 복사하여 캐시 스토리지의 데이터와 storage server 의 데이터를 동기화한다. 두 번째 경우인 캐시되어져 있지 않았을 경우에는 initiator 가 사용하려는 데이터를 storage server 에서 캐시 스토리지로 복사하고 캐

시 스토리지에 복사된 데이터를 initiator 가 사용하도록 한다. 그리고 initiator 에 의해 데이터가 수정되면 캐시되어진 경우처럼 캐시 스토리지는 storage server 와 데이터를 동기화하기 위해 수정된 데이터에 대한 명령의 실행이 종료되어지면 수정된 데이터를 storage server 에 복사한다.

### 4. 성능측정 및 분석

본 논문에서는 성능평가를 위해 리눅스 기반의 네트워크 시뮬레이터(NS-2)를 사용하였다. Cache server 가 존재하지 않는 기존의 원격 스토리지 시스템과 cache server 가 존재하는 캐시 스토리지 시스템(hit, miss)과의 성능을 서로 비교하였다. 실험환경은 하나의 iSCSI initiator, cache server, 스토리지 라우터, 4 개의 SCSI 하드디스크로 구성하며 [그림 4] 와 같다. Initiator 와 cache server 는 Linux Kernel 2.4.18 의 OS 를 사용하였다. Initiator 는 cache server 와 연결되며 cache server 는 스토리지 라우터와 연결된다. 스토리지 라우터는 9.5GB 크기를 가지는 4 개의 SCSI 하드디스크를 FC 인터페이스로 연결하고 있다.



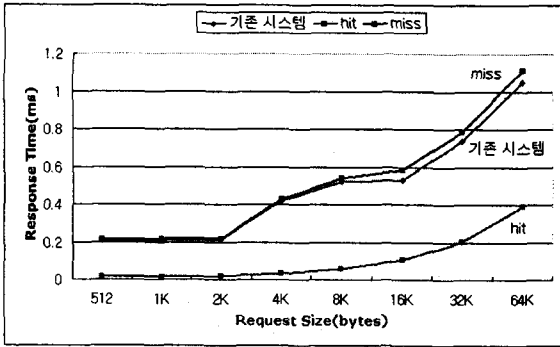
[그림 4] 실험 시스템 구성

[그림 5]는 cache server 가 존재하지 않는 기존의 원격 스토리지 시스템과 cache server 가 존재하는 캐시 스토리지 시스템의 hit 일 때와 miss 일 때의 성능을 서로 비교한 결과이다. Initiator 의 request size 를 512Byte 에서 64KByte 까지 늘려가며 각각의 크기에 따른 응답시간을 측정해 보았다. [그림 5]에서 볼 수 있는 것과 같이 hit 에 대한 응답시간은 기존의 원격 스토리지 시스템과 miss 일 때의 응답시간에 비해 매우 짧은 응답시간을 제공한다. 이는 initiator 가 먼 곳의 스토리지까지 가지 않고 가까운 곳에 있는 cache server 의 데이터를 사용하기 때문에 응답시간이 줄어드는 것으로 분석되며 request size 가 2K 이상의 크기를 가질수록 응답시간 차이의 폭이 더 커진다는 것을 알 수 있다.

기존의 원격 시스템과 캐시 스토리지 시스템의 miss 일 때의 응답시간을 살펴보면 응답시간 폭의 크기는 크지 않지만 기존의 원격 시스템의 응답시간이 miss 일 때의 응답시간보다 조금 더 빠르다는 것을 알 수 있다. 이것은 miss 일 때는 initiator 가 요청한 데이터가 캐시 스토리지에 캐시되어져 있는지를 검사하는 시간이 더해지기 때문에 그만큼의 지연시간이 추가된다는 것으로 분석되어진다.

결과적으로 initiator 가 요청한 데이터가 캐시 스토리지에 캐시되어져 있다면 사용자에게 보다 빠른 응답시간을 제공할 수 있다는 것을 알 수 있다. 따라

서 제안되어진 캐시 스토리지 시스템에 initiator 의 요청에 대한 hit rate 를 높인다면 많은 데이터에 대한 응답시간을 줄여 주므로 더욱 좋은 성능을 가질 수 있다는 사실을 추측할 수 있다.



[그림 5] Request size 에 대한 응답시간

5. 결론 및 향후 과제

본 논문에서는 iSCSI 기반의 네트워크 환경에서 네트워크의 성능을 향상시키기 위한 새로운 캐싱 기법을 제안하였다. 캐시되어진 데이터가 모두 fresh 한 상태이기 때문에 캐시되어진 데이터에 대한 유효성을 검사 하지 않는다는 것이 제안된 캐싱 기법의 가장 큰 특징이다. 이 캐싱 기법을 사용하여 성능을 측정 한 결과 기존의 원격 스토리지 시스템보다 본 논문에서 제안한 캐시 스토리지 시스템이 보다 나은 성능을 가진다는 것을 알 수 있었다. 따라서 사용자들에게 캐시 스토리지에 캐시되어진 데이터를 사용할 수 있도록 한다면 보다 빠른 응답시간을 제공할 뿐만 아니라 cache server 와 storage server 사이의 네트워크 대역폭을 효율적으로 사용할 수 있어 네트워크의 성능을 향상시킬 수 있다.

항상 fresh 한 상태의 데이터에 대한 initiator 의 hit rate 을 최대화 할 수 있다면 네트워크의 성능은 더욱 향상 되어질 것이다. Hit rate 를 최대화 할 수 있는 replacement policy 을 제안하여 본 시스템에 적용하는 것이 향후 과제이다.

참고문헌

- [1] Julian Satran, iSCSI Draft 20, "http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt"
- [2] Nishan iSCSI Technical White Paper, "http://www.NishanSystem.com"
- [3] SNIA IP Storage Forum White Paper, "http://www.SNIA.org"
- [4] Xubin He, Qing Yang, "Architecture and Performance Potential of STICS-SCSI-TO-IP Cache Storage"
- [5] Configuring Hierarchical Squid Caches, "http://www.squid-cache.org/Doc/Hierarchy-Tutorial/tutorial-1.html"
- [6] 박수라, 문보석, 김대근, 김주호, 김경훈, 박명순, Chutimet Srinilta, 신해선, 박성순, "모바일 기기를

위한 iSCSI 기반의 원격 스토리지 시스템 설계 및 구현", 2003 춘계 스토리지 시스템 학술대회 논문집, pp. 236-240

- [7] NS Manual, "http://www.isi.edu/nsnam/ns/ns-documentation.html"
- [8] NS Tutorial, "http://www.isi.edu/nanam/ns/tutorial/"