

## 네트워크 스토리지 QoS를 위한 네트워크 대역폭 제어기법\*

류준길<sup>0</sup> 박찬익

포항공과대학교 컴퓨터공학과

{lancer<sup>0</sup>, cipark}@postech.ac.kr

Network Bandwidth Regulation for Network Storage QoS

Junkil Ryu<sup>0</sup>, Chanik Park

Department of Computer Science and Engineering,  
Pohang University of Science and Technology (POSTECH)

### 요약

공유 스토리지 시스템에서 개별 클라이언트의 요구 QoS (Quality of Service) 를 만족시켜주기 위해서 각 클라이언트의 워크로드를 개별화 하고 성능을 차별화해 줄 수 있는 방안이 필요하다. 이것을 위해서 기존 스토리지 시스템에서 사용해온 방법은 YFQ [1], WFQ와 같이 스토리지에 영령 큐를 관리하는 방식을 사용하여 왔다. 그러나 큐 관리 방식은 스토리지 요구영령 크기(Request Size)가 일정하다는 가정 하에서 가능하다. 그런데 클라이언트의 요구영령 크기는 다양하다. 따라서 초당 IO의 수를 규제하여 QoS를 제공하는 큐 관리 방식은 다양한 요구영령 크기를 가지는 네트워크 스토리지 QoS를 위한 최선의 방안이 될 수 없다. 본 논문에서는 클라이언트의 요구영령 크기가 다양하다는 것을 인정하고, 네트워크 스토리지에서 사전에 약정된 QoS를 제공하기 위해서 큐 관리 방식은 다양한 요구영령 크기의 분포는 더욱 넓어지고 있다. 기존 스토리지 시스템에서 QoS를 지원하기 위해서 이용해온 큐 관리 방식은 네트워크 분야의 큐 관리 방식을 응용한 것이다. 네트워크에서 패킷의 사이즈는 일정하기 때문에 큐 관리 방식을 사용해도 아무런 문제가 없었다. 또한 일정 크기의 요구영령 크기를 사용하는 스토리지 시스템 환경이라면 큰 문제가 없을 것이다. 오늘날 네트워크 스토리지의 클라이언트는 데이터베이스나 일반 PC뿐만 아니라 TV, 모바일 기기 등 다양하다. 이러한 환경에서 네트워크 스토리지는 더 크고 다양한 크기를 가지는 요구영령들을 받는다. 따라서 다양한 데이터를 저장하고 있는 (IP) 네트워크 스토리지 환경에서 기존 큐 관리 방식을 고집하는 것은 최선의 방법이 아니다.

### 1. 서 론

1999년에 수행된 PC와 서버 워크로드를 분석한 연구[2]에서 는 클라이언트의 스토리지에 대한 읽기/쓰기 요구영령 크기는 일정한 값이나 평균값 근처에 몰려있는 것이 아닌 넓게 분포되어 있으며 시간에 따른 초당 IO의 수도 다양성을 보여주었다. 현재, 데이터 크기는 더욱 커지고(특히 멀티미디어 데이터의 증가), 데이터 태입은 다양해졌기 때문에 요구영령 크기의 분포는 더욱 넓어지고 있다. 기존 스토리지 시스템에서 QoS를 지원하기 위해서 이용해온 큐 관리 방식은 네트워크 분야의 큐 관리 방식을 응용한 것이다. 네트워크에서 패킷의 사이즈는 일정하기 때문에 큐 관리 방식을 사용해도 아무런 문제가 없었다. 또한 일정 크기의 요구영령 크기를 사용하는 스토리지 시스템 환경이라면 큰 문제가 없을 것이다. 오늘날 네트워크 스토리지의 클라이언트는 데이터베이스나 일반 PC뿐만 아니라 TV, 모바일 기기 등 다양하다. 이러한 환경에서 네트워크 스토리지는 더 크고 다양한 크기를 가지는 요구영령들을 받는다. 따라서 다양한 데이터를 저장하고 있는 (IP) 네트워크 스토리지 환경에서 기존 큐 관리 방식을 고집하는 것은 최선의 방법이 아니다.

본 논문에서 제안하고자 하는 방법은 클라이언트의 QoS를 만족시켜 주기 위해서 초당 IO의 수 (IOPS)를 규제하는 것이 아니라, 네트워크 스토리지와 클라이언트간 연결의 네트워크 대역폭을 규제함으로써 클라이언트의 약정 QoS를 제공할 수 있도록 하는 것이다.

### 2. 문제 정의

네트워크 대역폭 조절을 통해서 네트워크 스토리지 QoS를 제공하기 위해서는 다음과 같은 문제가 해결되어야 한다.

- 네트워크 대역폭 제어기법
- 네트워크 및 스토리지 모니터링 기법

\* 본 연구는 한국과학재단 특정기초연구과제 R01-2003-00010739-0의 지원을 받아 수행되었습니다.

### • 병목발생시, 우선순위에 따른 조정

### 2.1 네트워크 대역폭 제어기법

기존 큐 관리 방식의 경우, 읽기 명령과 쓰기 명령을 구별 없이 IOPS를 규제하는 방식을 사용한다. 그러나 네트워크 대역폭을 규제하는 경우 읽기 명령과 쓰기 명령이라는 명령 큐를 관리하는 것이 아닌 각 연결 당 네트워크를 통해서 보내고 받는 것을 각각 규제하여야 한다. (본 논문에서 대상으로 하고 있는 네트워크 스토리지는 IP 네트워크를 사용한다.) IP 네트워크상의 대부분 네트워크 스토리지 서비스들은 TCP를 사용한다. 그러나 표준 TCP는 특정 연결에 네트워크 대역폭을 할당하거나 규제할 수 없다. 오히려 TCP의 경우 프로토콜 상 더 작은 Round-Trip Time (RTT)을 가지는 연결을 선호하기 때문에 더 작은 RTT를 가지는 연결이 더 많은 대역폭을 할당 받게 된다 [3]. 따라서 본 논문에서는 TCP를 이용하는 각 클라이언트에 약정 QoS에 맞는 서비스를 제공해주기 위해, TCP flow control를 이용하여 각 연결의 대역폭을 규제하는 방법을 제시한다. TCP Flow Control를 이용하는 기존 연구로는 "PacketShaper" [4]와 같이 네트워크 라우터에서 특정 방향으로 네트워크 트래픽을 규제하는 경우와 적은 네트워크 대역폭을 가진 PC에서 특정 용용을 위한 연결에게 일정한 수준 받기 대역폭을 유지해주기 위한 연구[5]가 있다. "PacketShaper"의 경우, 데이터 리시버가 보내는 TCP advertised window 크기와 ACKs 를 네트워크상에서 가로채서 원하는 트래픽 수준을 달성할 수 있도록 거짓된 TCP advertised window 크기와 ACKs 를 새 스케줄한다. 그러나 "PacketShaper"는 하드웨어로 구현된 방식이고 이 방식은 네트워크 인프리를 고려하거나 변경해야한다는 단점을 가지고 있다. 연구 [5]의 경우, 당시 네트워크 환경에서 PC는 모뎀수준의 작은 네트워크 대역폭을 가지고 있기 때문에 네트워크를 이용하는 용용들 간에 PC에 연결된 적은 네트워크 대역폭을 나누어 쓰는 것이 목표였고, 이것을 해결하기 위해서 TCP advertised window 크기와 ACKs를 이용하여 PC에서 받는 트래픽에 대해서 네트워크 대역폭을 나누어 쓸 수 있도록 하였다. 그러나 연구 [5]에서 제안한 방식은 PC에서 보내는 트래픽이 있다면 원하는 목표를 달성하지 못하게 된다. 또한 연구 [5]의 방법은 PC에 연결된 네트워크에 병

록 구간이 발생할 경우 사용할 수 있는 방법이다. 본 연구에서는 스토리지 상황에 따라 특정 연결에 대해서 보내기 및 받기 네트워크 트래픽을 규제 할 수 있어야 한다.

## 2.2 네트워크 및 스토리지 모니터링 기법

네트워크 스토리지에서는 네트워크와 스토리지, 두 개의 자원을 모니터링 해야 한다. 스토리지 시스템은 네트워크 시스템과 달리 워크로드에 따라 발휘할 수 있는 성능이 다르다. [그림 1]은 동일 스토리지를 사용함에도 불구하고 워크로드의 특성에 따라서 성능이 다를음을 보여주고 있다. [그림 1]의 순차 읽기의 경우 일정 반응시간이내에서 1600개 이상의 IOPS를 처리할 수 있지만, 임의 읽기/쓰기의 경우 15msec 이내로 서비스할 수 있는 IOPS는 160개 정도이다. 또한 IOPS에 따른 반응시간은 일정 수준의 IOPS에서 갑작스럽게 커지기 때문에 스토리지 시스템의 모니터링은 네트워크에 비해서 세심하게 이루어져야 한다.

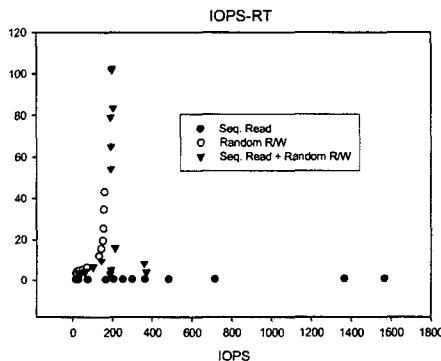


그림 1 초당 IO의 수와 반응시간의 관계  
(타겟스토리지: Seagate ST336607LW)

## 2.3 병목발생시 우선순위에 따른 조정

네트워크 스토리지 QoS를 위한 제어대상으로 각 연결의 네트워크 대역폭으로 하였다. 네트워크 대역폭을 제어대상으로 한 것은 다양한 스토리지 요구명령 크기를 대응하기 위해서 큐 관리 방식을 사용하는 것은 최선의 방법이 아니라는 인식과 스토리지뿐만 아니라 네트워크에서 병목현상이 발생했을 때 각 클라이언트의 QoS에 맞게 네트워크 대역폭을 할당할 수 있어야 하기 때문이다. 병목현상 발생시 우선순위 결정은 미리 결정된 우선순위가 있다면 그에 따르면 되지만 우선순위의 우열을 가릴 수 없을 경우, 그에 따른 해결책이 있어야 한다. 또한 스토리지 트래픽과 네트워크 트래픽은 bursty 하기 때문에 네트워크 스토리지 성능을 효율적으로 이용하기 위해서는 병목 해소되는 시점을 적절히 모니터링해서 각 클라이언트의 네트워크 대역폭을 동적으로 조정할 수 있어야 한다.

본 논문에서 사용하게 될 표현에 대해서 간단히 정리하면 다음과 같다.  $C_i$ 와  $Q_i$ 는 네트워크 스토리지 클라이언트  $i$ 와 그것의 네트워크 스토리지 QoS를 나타낸다. 일반적으로 스토리지 QoS는  $Q_i = \{f_i, iops_i, sz_i, s_i, rt_i\}$ 로 표현 된다[7]. 여기서  $f_i$ 는 클라이언트  $i$ 의 읽기/쓰기 요구명령의 비율,  $iops_i$ 는 초당 IO의 수,  $sz_i$ 는 요구명령의 크기,  $s_i$ 는 요구명령의 순차정도,  $rt_i$ 는 반응시간을 나타낸다. 그러나 클라이언트의 요구 QoS를 상세 할 때,  $f_i$ 와  $s_i$ 는 예측하기 어려운 값이다. 따라서 본 논문에서는 네트워크 스토리지를 위한 QoS를 다음과 같이 표현하였다.  $Q_i = \{r_{MBps}, w_{MBps}, avg(reqSize_i)\}$ ,  $r_{MBps}$ 와  $w_{MBps}$ 는 클라이언트

$i$ 의 읽기 및 쓰기를 위한 네트워크 대역폭,  $avg(reqSize_i)$ 는 평균 요구명령 크기이다.

## 3. 네트워크 스토리지 QoS를 위한 시스템

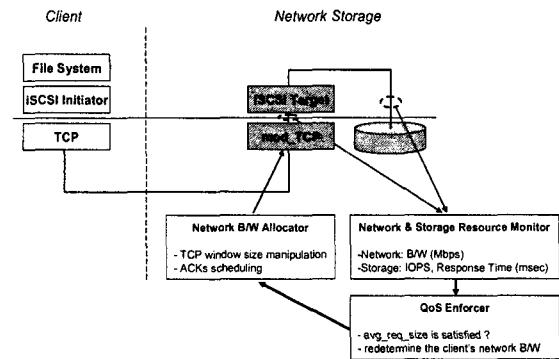


그림 2 네트워크 스토리지 QoS 제한된 방법

네트워크 스토리지 QoS를 위해서 제안된 시스템은 네트워크 대역폭 할당자 (Network Bandwidth Allocator, NBA), 네트워크 및 스토리지 시스템 모니터 (Network and Storage Resource Monitor, NSRM), QoS Enforcer로 구성되어진다.

### 3.1 네트워크 및 스토리지 시스템 모니터 (NSRM)

NSRM은 스토리지 시스템에서 각 클라이언트의 IOPS와 반응시간을, 네트워크 시스템에서는 네트워크 대역폭과 RTT를 모니터링 한다. NSRM은 일정 시간 구간마다 각 클라이언트가 약정 QoS를 만족하는지 [수식 1]을 이용하여 확인한다. [수식 1]은 요구 QoS의 네트워크 대역폭과 평균 요구명령의 크기를 가지고 스토리지 시스템이 제공해야 할 개별 요구명령에 대한 반응시간을 계산하고 일정 시간 구간에서 측정된 반응시간이 계산된 반응시간을 넘는지 확인한다. 본 네트워크 스토리지 시스템은 다양한 요구명령 크기에 대해서 QoS를 제공하는 것이기 때문에 개별 요구 명령에 대한 반응시간은 중요하지 않다. 단 평균화된 반응시간 또는 계산된 반응시간을 초과하는 횟수 등을 이용하여 QoS가 불만족인 상황인지를 파악한다.

$$W\_RT_i = \frac{avg(reqSize_i)_{QoS}}{w\_MBps_{QoS}}, R\_RT_i = \frac{avg(reqSize_i)_{QoS}}{r\_MBps_{QoS}}$$

$$w\_rt_{ij} + rtt_i \geq W\_RT_i + \alpha, r\_rt_{ij} + rtt_i \geq R\_RT_i + \alpha$$

[수식 1]  $w\_rt_{ij}$ 와  $r\_rt_{ij}$ 에서  $i$ 는 연결을  $j$ 는 단위구간에서  $j$ 번째 명령을 의미한다.

본 논문에서는 측정된 반응시간이 계산된 반응시간을 초과한 횟수가 일정치를 넘게 되면 QoS Enforcer에게 QoS 불만족 상황을 해결하도록 알려주게 된다.

### 3.2 QoS enforcer

각 클라이언트의 요구 QoS가 만족하지 못하면 QoS enforcer는 우선 네트워크 자원이 병목현상을 보이고 있는 확인하다. 이것은 간단하게 개별 연결들의 네트워크 대역폭의 합이 네트워크 자원을 90% 이상 사용하고 있는지와 RTT를 확인한다. 네트워크 자원에 문제가 발생하지 않은 경우 [수식 2]와 같은 상황이 발생했는지 확인하다.

$$\text{if } \text{avg}(\text{reqSize})_{M_m} \leq \text{avg}(\text{reqSize})_{QoS}$$

$$IOPS_{M_m} \geq \frac{(r_w)_{MBps} QoS}{\text{avg}(\text{reqSize})_{QoS}}$$

[수식 2]  $X_{M_m}$ 은 측정된 값,  $X_{QoS}$ 는 QoS 값

[수식 2]는 스토리지로의 요구명령의 크기가 약정된 평균 요구명령의 크기보다 작다면 네트워크 자원에는 영향을 미치지 않으면서 스토리지 성능에 영향을 미칠 수 있기 때문에 약정된 평균 요구명령 크기보다 작은 요구명령을 내려 보낼 때는 일단 약정 QoS 설정을 클라이언트가 위반하게 된 것이기 때문에 QoS를 지켜줄 우선순위에서 밀리게 된다. 따라서 측정된 IOPS가 계산된 IOPS보다 많은지를 확인하고 많으면 네트워크 대역폭을 줄여서 해당 워크로드를 규제한다.

### 3.3 네트워크 대역폭 할당자 (NBA)

NBA는 클라이언트의 약정 QoS가 만족하도록 각 클라이언트의 네트워크 대역폭을 할당한다. NBA는 클라이언트의 도움 없이 네트워크 스토리지 서버의 TCP Flow Control 조작을 통해서 네트워크 대역폭을 할당하기 때문에 매우 중요하다. NBA는 [수식 3]이 보여주는 TCP Flow Control을 이용하여 구현되었다.

$$\text{flow rate} = \frac{w \times \text{Packet size}}{\text{RTT} + d}$$

w: TCP (receiving/sending) window size,  
d: ACKs 지연시간

[수식 3] TCP Flow Control

예를 들어, 네트워크 스토리지 시스템으로 들어오는 트래픽을 줄이기 위해서 네트워크 대역폭을 줄이기 위해서는 receiving window 크기를 줄이고 ACKs를 지연시킨다. 연구 [5]의 경우 RTT가 수십 msec 단위였기 때문에 RTT를 조정하더라도 충분하지만 본 연구의 실험환경은 기가비트 네트워크 환경으로 1 msec 이하의 RTT를 갖고 있기 때문에 RTT를 임의로 조작하지는 않았다. 또한 연구 [4,5]는 네트워크에서 트래픽을 조절하면 되지만 본 연구에서는 네트워크 트래픽의 결과가 스토리지 단에 나타나야 한다. 본 연구에서 사용하는 환경인 iSCSI 프로토콜은 요구명령(request)에 대한 반응명령(response)이 와야 한 트랜잭션이 완료된다. 따라서 보내고 받기 트래픽을 분리해서 제어할 때, ACK를 piggybacking하는 것을 주의하여 분리해야 한다.

### 4. 성능 평가

네트워크 스토리지 QoS를 위한 시스템은 Linux-2.6.11 커널과 Intel iSCSI 프로토콜[6]을 변경하여 구현하였다. 실험환경은 두 대의 클라이언트(P4 2GHz), 한 대의 스토리지 서버(P4 2GHz, SCSI 디스크: Seagate ST336607LW)와 3Com 기가비트 스위치로 구성하였다.

[그림 3]은 크기가 128KB이고 임의의 쓰기인 IO 요구명령을 초당 1개부터 250여 개까지 발생하여 측정한 것이다. 약 22 MBPS 정도에서 스토리지의 반응시간이 급격히 증가하는 것을 알 수 있다 (RTT는 본 실험환경에서는 0.1 msec로 무시할 수 있음). [그림 4]는 두 클라이언트가 약정 QoS에 관계없이 시간에 따라 많은 임의 쓰기 요구명령을 발생 시킬 때, 각 QoS를 만족하지 못하는 경우가 발생하게 되고 이에 따라 QoS를

규제하는 것을 보여준다. 9초 시간에 클라이언트 A의 QoS (10MBPS, 128KB: 12.5 msec이하의 반응시간 (수식1))을 만족시키지 못하게 되고 이에 따라 약정 QoS 이상을 사용하고 있는 클라이언트 B의 네트워크 대역폭을 규제하게 된다. 마찬가지로 13초에 가면 QoS 불만족 상황이 발생하게 되고 이전 클라이언트 A의 네트워크 대역폭을 규제하게 된다.

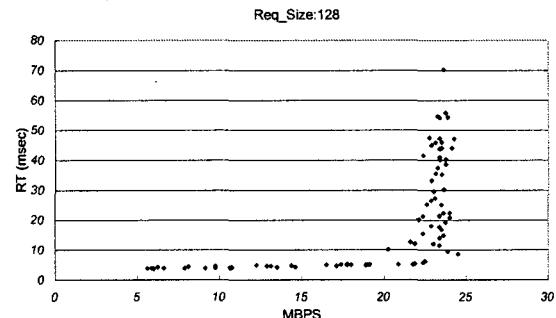


그림 3 MBPS(IOPS) 증가에 따른 반응시간

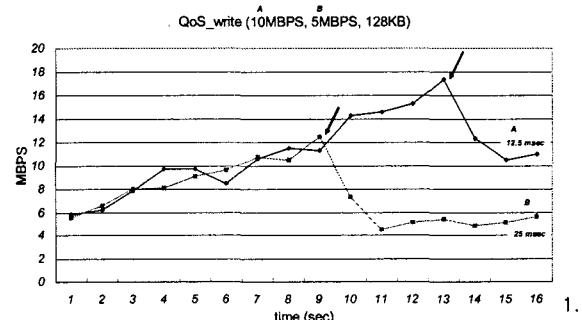


그림 4 두 클라이언트 간 QoS에 따른 네트워크 대역폭 조정과 결과

본 논문에서는 네트워크 대역폭 조정을 통해서 네트워크 스토리지 QoS를 제공할 수 있다는 것을 보여주었다. 이것과 관련하여 더 많은 실험과 여기에서는 미처 언급이 되지 못한 과부 하된 상태에서 우선순위를 결정하여 QoS를 제공하는 방법에 대한 연구가 진행될 것이다.

### 참고 문헌

- [1] Silberschatz, "Disk Scheduling with Quality of Service Guarantees," IEEE ICMCS 1999
- [2] W. Hsu and A. Smith, "Characteristics of I/O traffic in personal computer and server workloads," IBM System Journal, vol. 42, no.2, 2003
- [3] Thomas R. Henderson, Emile Sahouria, Steven McCanne and Randy H. Katz, "On Improving the Fairness of TCP Congestion Avoidance," in proceedings of Globecom 1998
- [4] Packeteer, <http://www.packeteer.com>
- [5] Puneet Mehra, Avideh Zakhor and Christophe De Vleeschouwer, "Receiver-driven Bandwidth Sharing for TCP," in proceeding of INFOCOMM 2003
- [6] Intel iSCSI reference implementation <http://sourceforge.net/projects/intel-iscsi>
- [7] J. Wilkes, "Traveling to Rome: QoS specifications for automated storage system management," Proceedings of International Workshop on Quality of Service, June 2001.