ISBN 978-89-5884-983-4 98560

SCSI RDMA Protocol 기술 동향 분석

일 자: 2007년 12월 7일

제출자: 최 민(한국과학기술원)
김한준(한국과학기술원)
차광호(한국과학기술정보연구원)
조혜영(한국과학기술정보연구원)
김성호(한국과학기술정보연구원)

목 차

1. 서론

과거 저장장치 시스템은 네트워크를 고려하여 설계되지 않았다. 하지만, 오늘날의 웹 서버, 데이터베이스, 멀티미디어, 인터넷 데이터 센터 등의 대용량 스토리지 서브시스템의 대중화로 인하여 스토리지 시스템의 대역폭, 성능, 연결성과 같은 여러가지 측면에서 매우 큰 확장가능성(scalability)을 필요로 하고 있다. 결국, 저장장치 서브시스템 역시 오늘날 대부분 경우와 유사하게 인터넷 프로토콜(IP)을 적용하여 원거리 통합을 지원하고 원격 관리를 가능하도록 확장되어야 한다. 이에 따라, 과거 RAID, SCSI, Fibre Channel 등의 과거 저장장치 기술들과 TCP/IP, Ethernet, Infiniband, LAN, WAN, SONET 등의 네트워크 전송기술, 그리고 VPN, IPsec 등의 보안기술이 결합하여 스토리지 네트워크의 통합을 시도하고 있다 (그림 1).

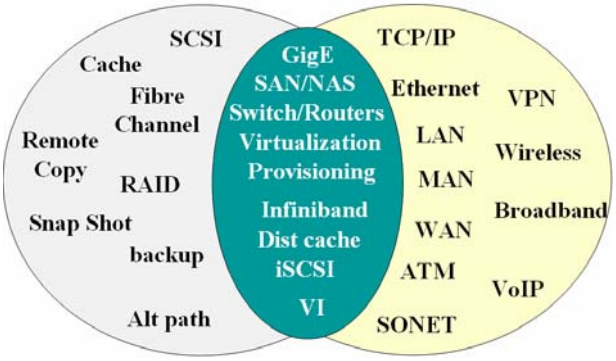


그림 1 네트워크와 저장장치 기술의 통합

오늘날 저장 매체 기술의 발달로 저장장치 가격이 급속히 떨어지고 있다. 또한 네트워크 및 인터넷 기술의 발달에 따라 데이터의 크기가 증가함으로 해서, 대용량 저장장치에 대한 요구가 높아지고 있다. IDC의 분석에 의하면 2003~2005년 사이에 저장장치의 용량은 해마다 75% 이상 증가하고 있고, 이러한 증가추세에는 두가지 중요한 이슈가 있는데 데이터 중요도의 증가와 저장장치 resource 관리의 어려움이 그것이다. 현재의 TCP/IP 기반의 네트워크 저장장치 시스템에서 사용되고 있는 입출력 버스 방식은 디스크 접근 특히 고성능의 서버에 있어서 병목현상의 주요원인으로 나타나고 있다. 이러한 버스 방식은 구조가 단순하다는 큰 잇점을 갖고 있어 지금까지 산업 전반적으로 많이 사용되어 왔지만 버스 기반의 입출력 시스템은 현재의 디바이스 장치들이 요구하는 데이터 전송 대역폭을 처리할 수 있을 만큼의 시스템 입출력 성능을 가지고 있지 않다. 뿐만 아니라 대용량의 데이터를 다수의 사용자에게 서비스하기에는 많은 문제점을 가지고 있다.

인터넷 데이터 센터의 대부분의 경우에 NAS(Network Attached Storage)와 SAN을 이용하여 웹 서비스나 응용프로그램에 저장장치 서비스를 제공하고 있다.

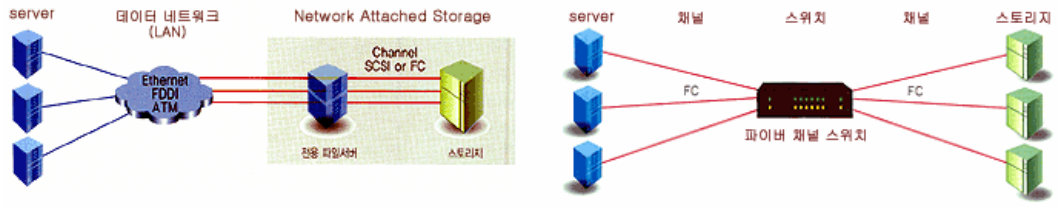


그림 2 NAS와 SAN의 비교

NAS는 네트웍(LAN)에 접속된 저장장치이다. 그림 2의 좌측에서 보듯 NAS는 전용파일서버와 스토리지로 구성되어 있다. 전용파일서버와 NAS의 Data를 이용하는 애플리케이션 서버 사이는 LAN에 접속되어 TCP/IP 프로토콜로 통신하고 전용파일서버와 스토리지는 SCSI 또는 Fibre Channel과 같은 채널로 연결되어 SCSI 프로토콜로 통신한다. NAS의 장점은 파일공유이다. 여러 애플리케이션 서버들이 LAN을 통해 NFS또는 CIFS와 같은 파일 서비스 프로토콜로 전용파일서버에 접속하여 파일에 대한 서비스를 요청하면 단일 파일서버가 그 요청에 따라 파일서비스를 하게 되므로써 즉 NAS에 저장된 파일이 모두 전용파일서버 한 곳에서 관리됨으로써 파일들에 관한 정보들의 Consistency라든가 locking에 문제가 없이 파일을 여러 서버들이 공유할 수 있게 된다. NAS의 단점은 성능과 DB에서 사용할 때의 문제점이다. 성능상의 단점중의 한 요인은 Latency Time이다. NAS는 애플리케이션 서버에서 전용파일서버까지 네트웍으로 접속되고 전용파일서버에서 스토리지사이는 채널로 접속되어 채널로만 접속되는 DAS또는 SAN에 비해 접속단계가 늘어남으로서 Latency Time이 더 걸리게 된다. SAN은 그림 2의 우측에서 볼 수 있듯이 서버와 스토리지 사이의 채널 접속에 파이버 채널 스위치를 넣어 네트웍의 개념을 도입한다. 서버의 접속 포트 하나에서 여러대의 스토리지를 접속할 수 있고 또한 스토리지의 접속 포트 하나에 여러 서버가 접속할 수 있는 유연성이 생기게 된다. SCSI Switch가 아닌 파이버채널 스위치를 사용하는 이유는 SCSI의 경우 Open System의 채널 인터페이스이긴 하지만 접속 거리가 최대 25m로 네트웍으로 구성하기에는 거리제약이 있으며 스위칭을 위한 고려가 전혀 되어있지 않는 인터페이스란 점 때문에 파이버 채널을 SAN의 표준으로 정하게 되었다.

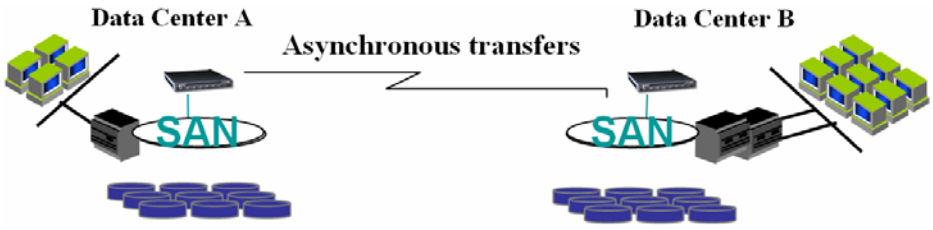


그림 3 원거리 저장장치 네트워크 구조

최근에는 데이터 접근성의 확대, 비즈니스의 성장 및 다양화, 그리고 데이터를 보다 고객

과 가까운 곳에 캐싱(caching)하려는 정책 등에 의하여 저장장치 네트워크의 확장성이 강하게 요구되었다. 따라서, 단일 가상 스토리지(Virtual Storage Point of Presence)의 필요성이 나타나게 되었다. NAS는 SAN을 위한 스토리지 풀(pool)의 형태로 사용되며, 지역적으로 떨어진 SAN 네트워크가 서로 연결되어 가상의 단일 저장장치 서브시스템을 형성한다.

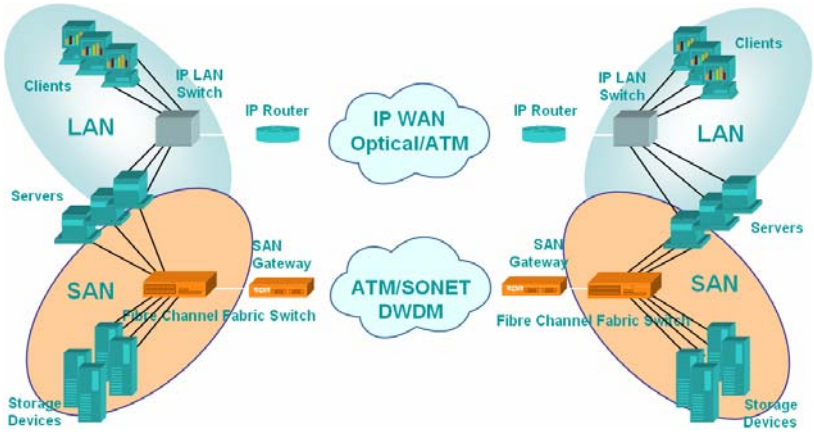


그림 4 과거 스토리지 네트워크 환경[1]

과거에는 이러한 스토리지 서브시스템이 IP 네트워크 상에서 각각의 SAN을 연결하기 위해 별도의 네트워크를 필요로 하는 환경으로 구현되었으며, 점대점(point-to-point) 접근방식으로 그 사용이 제한되었다. 따라서, 기존의 IP 네트워크와의 통합된 서비스가 불가능했다.

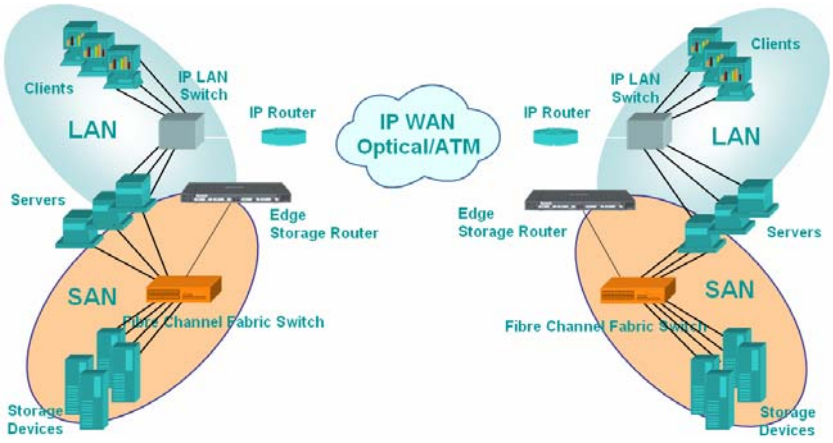


그림 5 이상적인 스토리지 네트워크 환경

하지만, 이상적으로는 그림 5와 같이 보편적인 IP 네트워크를 그대로 활용할 수 있도록 스

[1] DWDM은 조밀 파장 분할 다중화(dense wavelength division multiplexing) 방식으로 각 신호들은 분리된 고유의 광파장 상에서 전송하는 것이다. 하나의 광섬유에 최고 80개의 분리된 파장이나 데이터 채널로 다중화 할 수 있다. 마치, 두개의 라디오 방송국이 상호간에 영향을 주지 않고 서로 다른 파장을 갖는 신호를 방송하는 것과 유사하다.

토리지 네트워크가 이에 통합되어야 한다. 이를 위해 다양한 네트워크 스토리지 기술이 제안되었으며 상당수가 IETF에 의해 표준화 되었거나 현재 draft 형태로 제안되어 있는 상태이다.

이와 같이, 네트워크 기반 저장장치 기술은 사용되는 네트워크의 종류에 따라 두 가지로 분류할 수 있다. 오늘날 가장 많이 사용되는 TCP/IP 기반의 저장장치 기술과 인피니밴드 기반의 저장장치 기술이다. TCP/IP 기반의 네트워크 저장장치도 전통적인 NIC, TOE NIC 등의 하드웨어 발전에 따라 고속의 저장장치 서비스를 제공해 왔고, 소프트웨어에서는 저장장치 기술을 구현하는 프로토콜 기술도 함께 개발되었다. 그러나, 글러스터 시스템에서 요구되는 대용량의 I/O 대역폭을 지원하기에는 역부족이며, 그 결과 인피니밴드 저장장치 기술이 탄생하였다.

2. TCP/IP **기반의 저장장치 기술**

TCP/IP 기반 저장장치 기술의 가장 큰 이점은 지금 당장 사용가능할 뿐 아니라 비용 절감 효과가 높다는 것이다. 이와 같은 장점을 발휘하는데 가장 큰 공을 세운 것은 이더넷의 급성장이다. 파이버 채널에 근접하는 수준으로 고속을 구현했기 때문이다. TCP/IP 기반 저장장치의 이점으로는 상호 연동성이 있다. 비용 절감 측면에서 파이버 채널의 HBA(Host Bus Adapter)와 파이버 채널 스위치가 필요 없기 때문에 비용 지출이 줄어든다. IP 스위치 포트는 파이버 채널 스위치 포트보다 2~3배 저렴하며 파이버 채널 회선 대신 저렴한 네트워크 회선을 이용하면 된다. 이론상으로는 TCP/IP 기반 저장장치 기술을 구현해 SCSI 블록 레벨의 프로토콜을 보다 멀리, 그리고 장비를 무제한 연결할 수 있다. 또한 SAN을 관리하기 위한 전문 기술을 필요로 하지 않으며, 기존 IP 네트워크에서 많이 활용되는 NMS(Network Management Software)도 사용 가능한 장점이 있다.

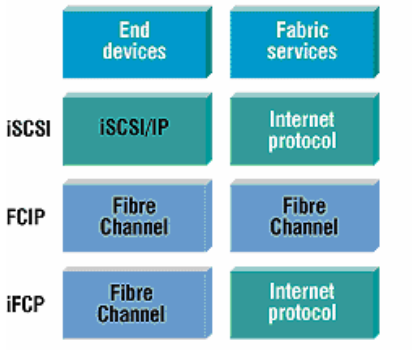


그림 6 iSCSI, FCIP, iFCP의 전송계층

대표적인 IP 저장장치 트랜스포트로는 iSCSI, FCIP, iFCP 가 있다. 이들은 모두 공통적으로 IP 네트워크 상에서 블록 단위(block level) 저장장치 트랜스포트를 제공하며, 사용자로 하여금 기존의 존재하는 저장장치 디바이스(HDD)와 기가비트 네트워크를 활용할 수 있도록 한다. 그 밖에도 더 많은 응용프로그램들로 하여금 활용할 수 있는 최대한의 저장공간을 제공하며 DAS나 SAN 이 갖고 있는 지리적 확장성의 한계를 극복하며, 기존의 응용프로그램들도 수정을 가하지 않고 그대로 사용할 수 있다는 장점이 있다. FCIP와 iFCP는 모두 파이버 채널에 기반한 프로토콜이며 FC는 그림 7과 같은 성능을 보인다.

Product Naming	Throughput (MBps)	Line Rate (Gbaud)	T11 Spec Completed (Year)	Market Avaiability (Year)
1GFC	200	1.065	1996	1997
2GFC	400	2.125	2000	2001
4GFC	800	4.25	2003	2005
8GFC	1,600	8.5	2006	2008
16GFC	3200	17	2009	2011
32GFC	6400	34	2012	Market Demand
64GFC	12800	68	2016	Market Demand
128GFC	25600	136	2020	Market Demand

그림 7 Fibre Channel 성능 차트

2006년 현재 시장에서 구매가능한 4GFC 장비를 기준으로 초당 800MB 수준의 성능을 나타낸다.

2.1 iSCSI(internet Over SCSI)

iSCSI는 SCSI 프로토콜을 사용하여 IP 기반 네트워크에 블록 데이터를 전송하기 위한 IETF(Internet Engineering Task Force)의 표준이다. 즉, TCP/IP 프로토콜 네트워크를 활용해 저장장치 데이터를 전송하는 기술이다. 이 기술은 TCP/IP 네트워크상에서 SCSI 프로토콜이 바로 전송될 수 있도록 한다. iSCSI를 도입한 기업 네트워크는 SCSI의 명령어와 데이터를 원거리 통신망(WAN)에 접속되어 있는 장치(인터넷 경유 방식인 경우는 인터넷에 접속되어 있는 장치)에 전송, 보관할 수 있다. 또한, 공통의 이더넷 기반을 사용해 소규모의 SAN을 복수 구축하는 것도 가능하다. 이에 따라 iSCSI 환경에서는 프로토콜 변환에 따르는 부하가 감소해 저장장치 성능 효과를 얻을 수 있다. 이처럼 iSCSI는 TCP/IP와 SCSI를 결합함으로써 SAN과 NAS의 이점을 갖춘 기술로 각광받고 있다.

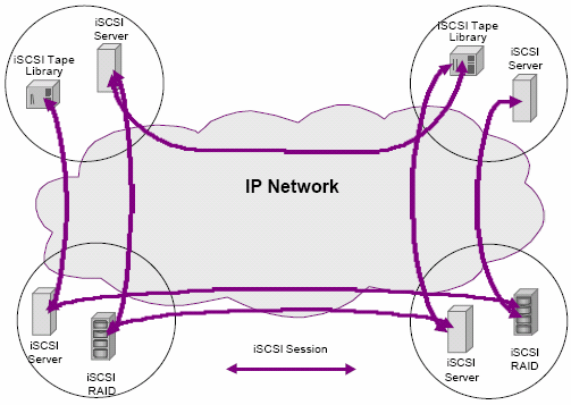


그림 8 iSCSI 네트워크 아키텍처

그림 9에서 볼 수 있듯이 iSCSI SAN은 파일 서버와 같은 iSCSI Initiator와 디스크 어레이나 테이프 서브시스템과 같은 iSCSI Target으로 구성된다. SCSI 인터페이스가 있는 시스템은 SCSI 명령어를 발동시키며, 가장 상위의 SCSI layer가 해당 명령어를 받아들인다. 그러면, iSCSI 계층이 해당 SCSI 명령 및 데이터 등을 캡슐화(encapsulation)하여, TCP/IP와 링크계층을 거쳐 목적지의 iSCSI Target 으로 전달된다.

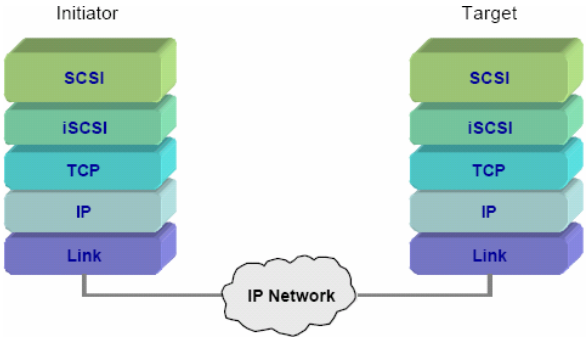


그림 9 iSCSI 프로토콜 계층 모델

수신 시스템은 패킷에서 SCSI 명령어를 분석하여 실행하며, 수신 유닛은 돌아오는 SCSI 명령어와 데이터를 IP 패킷 안으로 캡슐화한 다음 이들을 첫 번째 시스템으로 다시 돌려보낸다. 이 시스템은 데이터나 명령어를 분석하여 이들을 다시 SCSI 서브시스템으로 전달한다. 이러한 모든 작동은 사용자의 개입 없이 이루어지며, 최종사용자에게 완전히 투명하다.
iSCSI의 사양중 상당부분은 호환성 유지를 위해 표준 SCSI 실행들을 따라야 하고, SCSI를 깨뜨리지 않도록 해야 하기 때문에 있는 그대로 구성되었다. 이것은 또한 처음부터 IPv4나 IPv6 용으로 만들어졌다.

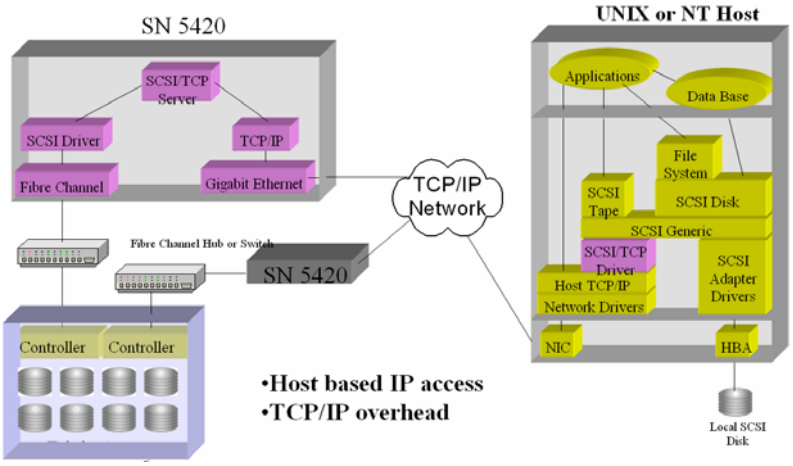


그림 10 CISCO iSCSI 아키텍처

그림 10는 CISCO의 SN 5420 라우터를 활용하여 iSCSI를 구축한 아키텍처이다. 그림 우측의 Unix나 NT호스트는 응용프로그램이나 데이터베이스의 파일 시스템 접근 요청을 SCSI/TCP Driver를 통해 iSCSI 패킷을 생성해 보낸다. 따라서, 마치 로컬영역의 스토리지를 액세스하는 것처럼 IP 네트워크의 임의 위치에 있는 스토리지를 직접 액세스할 수 있다. CISCO SN 5420 Storage Router는 다음과 같은 기능이 포함된다.

- ✓ 1U 높이의 스택형 장치
- ✓ iSCSI 기술
- ✓ Fibre Channel 포트
- ✓ Gigabit Ethernet 포트--1000Base-SX
- ✓ 고가용성
- ✓ GUI / CLI / SNMP 관리
- ✓ Fibre Channel 포인트-투-포인트, 루프 및 패브릭
- ✓ 보안이 확실한 스토리지 액세스를 위한 ACL(Access Control Lists, 액세스 제어 목록)
- ✓ LUN(Logical Unit Number) 맵핑

보안을 유지하기 위해, iSCSI 프로토콜에는 자체의 로그인 절차가 있다. 첫 번째 작동 시에 초기자(Initiator) 노드가 타켓(Target) 노드로 로그인을 한다. 로그인 프로세스를 수행하지 않은 초기자로부터 iSCSI PDU를 받은 타겟 노드는 어떤 것이건 프로토콜 에러를 발생시키고 접속을 종료한다. 단 이 때 타겟노드는 세션을 종료하기 전에 거부 iSCSI PDU를 보낸다. 이것은 기본적인 보안 형태인데, 왜냐하면 통신의 처음만을 보호하며, 모든 패킷 기반에서의 보안을 제공하지 않기 때문이다. 하지만 IPsec의 이용 등, iSCSI를 위한 보안을 제공하는 다른 방법들이 있다. 제어와 데이터 패킷 모두의 측면에서, IPsec은 무결성, 재생 보

호 및 인증을 준비해 줄 것이다. 또한 개별적 패킷들을 위한 암호화도 제공할 것이다.

이에 비해 파이버 채널은 그만큼 안전하지 못하지만, 파이버 채널 패브릭(fabric)으로 접속을 하려면 물리적 액세스와 파이버 채널에 대한 철저한 지식이 필요하다. 물론, 파이버 채널 보안의 핵심은 다른 모든 네트워크 접속을 하지 않는 것이다. 하지만 파이버 채널에는 암호화가 없으며, 프로토콜 레벨의 보안이 거의 없다. 많은 사람들이 알고 있는 바와 마찬가지로, TCP/IP 네트워크는 공격당하기 쉽다. 여기에는 외부 및 잘 알려진 프로토콜로의 접속성이 있다. 시스템 관리자들에게는 IP 네트워크를 안전하게 지킬 수 있는 많은 툴이 있고 다양한 경험을 갖고 있다. 이와 함께, 대다수 iSCSI SAN이 어떠한 공중 액세스도 없이 네트워크를 분리시킬것이라는 가능성은 보안 유지의 확률을 더욱 높여준다.

2.2 FCIP(Fibre Channel Over IP)

TCP/IP 기반 저장장치 네트워크의 또하나의 축인 FCIP(Fibre Channel Over IP)는 IP를 통해 파이버채널 터널링을 제공하며 그림 11과 같이 IP 패킷안에 파이버 채널 프로토콜을 캡슐화(encapsulation)한다.

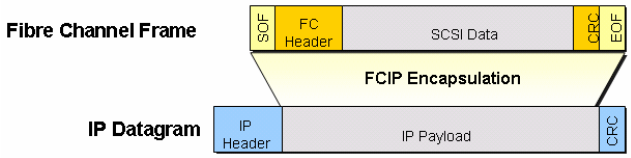


그림 11 FCIP 패킷의 IP터널링

이는 파이버 채널 SAN 네트워크 데이타의 원거리 전송을 위해 IP 네트워크를 하부 트랜스포트 계층으로 사용하는 것이다. 따라서, 적은 부대 비용으로 구축할 수 있고, 거리 제한 없이 FC기반의 SAN과 SAN을 연결할 수 있다. 또, 가상사설망(VPN), IPSec과 같은 기술을 이용하여 손쉽게 보안 문제를 해결할 수도 있다.

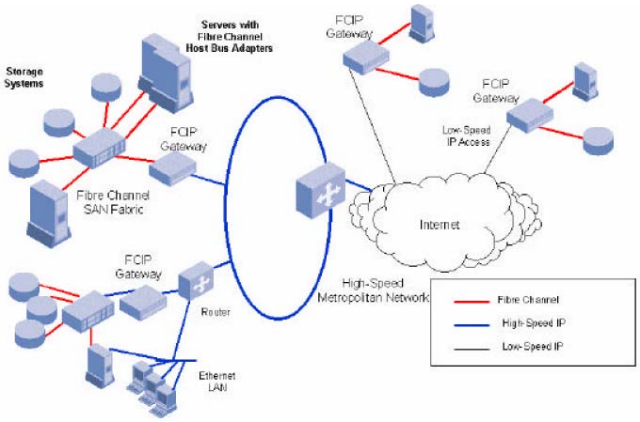


그림 12 FCIP 네트워크 아키텍쳐

FCIP는 둘 이상의 SAN 네트워크들의 상호연결(interconnection) 형태로 구성하여, 파이버 채널 서비스를 IP 네트워크를 이용하여 전송할 수 있도록 한다. 따라서, 상호연결된 둘 이상의 SAN 네트워크가 기존의 SAN 관리 응용프로그램을 이용하여 하나의 단일 SAN 네트워크로 취급되고 관리될 수 있다. 게다가 FCIP는 특성상 원격 백업, 복구 등 재해 복구 시스템에 활용 여지가 많아 저장장치 업체들 사이에 각광받고 있다. 2개 사이트의 완전 미러링을 통해 뜻하지 않은 사고 발생시 미러 사이트에 있는 오프라인 테이프를 통해 신속하게 대체 사이트를 온라인화 할 수 있기 때문이다. 하지만, 파이버 채널을 사용하는 SAN은 비용이 많이 들고, 관리하는데 전문 지식을 필요로 하는 단점이 있다.

2.3 iFCP(internet Fibre Channel Protocol)

iFCP는 IP와 SCSI 혹은 IP와 Fibre Channel 사이에서 게이트웨이 역할을 한다. SCSI나 Fibre Channel 서버와 저장장치가 iFCP 스위치를 통해 LAN이나 WAN으로 접근 가능하다. FCIP 처럼 iFCP는 파이버 채널 프레임을 캡슐화하여 TCP/IP 네트워크를 통해 전송한다. IETF에서는 일반적인 Fibre Channel 포맷을 정의하고 있다. FCIP와 iFCP의 중요한 차이점은 두 프로토콜 간에 강조하고 있는 면에서 차이를 보이고 있다. FCIP 프로토콜의 경우 두 Fibre Channel SAN을 연결하기 위해 점대점(point-to-point) 연결을 설정한다. 반면, iFCP는 그림 13에서 보는 바와 같이 게이트웨이 대 게이트웨이(gateway-to-gateway) 프로토콜이다. 다시 말하면, FCIP의 경우 파이버 채널 패킷을 받아서 그 내용에 관계없이 무조건 터널링 하여 IP네트워크를 통해 전송한다. 하지만, iFCP의 경우 FC 주소를 IP 주소로 변환해서 IP 네트워크로 전송한다. 그러면, 다시 해당 패킷에 대한 응답을 네임서버의 정보를 기반으로 FC 주소로 변환하여 FC 디바이스로 보내는 것이다.

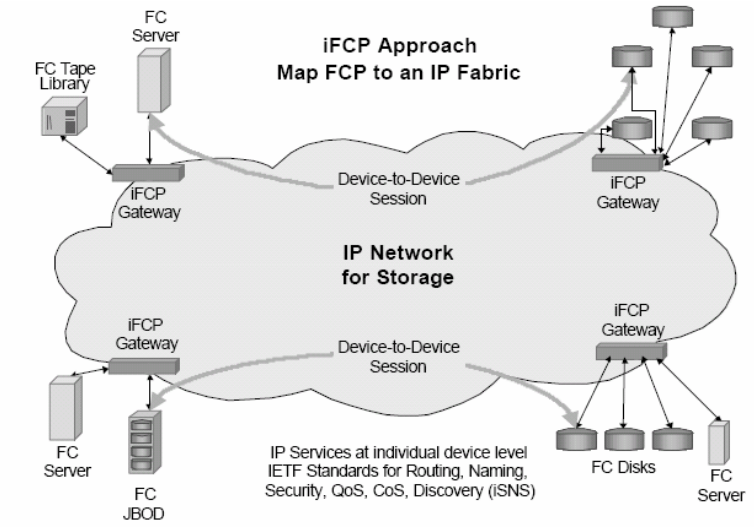


그림 13 iFCP 네트워크 아키텍쳐

따라서, iFCP의 장점은 서로 다른 SAN간 연결(interconnection)에 있어 파이버 채널 패브릭(Fibre Channel fabric)를 사용하지 않고, 트랜스포트 서비스를 TCP상에 매핑(mapping)할 수 있다는 것이다. 기존에 존재하는 FCP기반의 드라이버와 저장장치 컨트롤러들은 이들 제품들에 어떠한 변경도 가하지 않은 채, iFCP를 사용하여 서로 다른 SAN 네트워크들 간에 TCP/IP의 신뢰성 있는 트랜스포트 서비스를 사용할 수 있다.

3. 인피니밴드 기반의 저장장치 기술

결과적으로 TCP/IP 기반의 네트워크 저장장치의 문제점을 해결하기 위해 인피니밴드 기반의 네트워크 저장장치 기술이 등장하게 되었다. 인피니밴드 기술은 대용량 저장장치와 서버 간 입출력 분야에서 소규모의 서버에서부터 수백개의 프로세서와 수천개의 입출력 장치를 가진 대규모의 수퍼 컴퓨터까지 모두 사용 가능하다. 또한, 현재 대부분의 TCP/IP 기반의 네트워크 제품들은 최고의 패킷 처리량과 최소의 전송 지연, 그리고 전송 대역폭에 대한 보장을 요구해 왔다. 이를 인피니밴드에서는 전송 계층(transport layer)을 하드웨어 수준에서 구현하는 TOE(Transport Offload Engine)을 통해 가능하게 하였고, 소프트웨어에서는 커널 바이패싱(kernel bypassing) 등의 zero-copy 메커니즘을 적용하였다. 그리고 네트워킹에서는 신뢰성(reliable) 있는 전송 프로토콜을 사용하여 앞서 언급한 TCP/IP를 사용하지 못함으로서 발생하는 문제점을 해결하였다. 결국 인피니밴드는 전통적인 TCP/IP 기반의 네트워크 저장장치에서 불가능했던 초고속의 네트워크 데이터 서비스를 가능하게 함으로써 다수의 사용자에게 대용량의 데이터 서비스를 제공할 수 있게 되었다. 인피니밴드 기반의 저장장치 기술은 기존의 공유버스 시스템에서의 문제점과 TCP/IP 프로토콜 프로세싱에서의 문제점을 해결하면서 기존의 NAS(Network Attached Storage)와 SAN(Storage Area Network) 사이에 고성능의 데이터 전송을 가능하게 하는 네트워크 저장장치 기술이다. 또한 대규모의 컴퓨팅 파워를 필요로 하는 서버 클러스터 분야에도 적용되고 있다.

Storage Networking World 2004에서 인피니밴드 칩셋 제조 기업인 멜라녹스가 처음으로 초기 인피니밴드 저장장치 플랫폼을 선보였다. 저장장치 관련 산업표준을 기반으로 한 초기 인피니밴드 저장장치 플랫폼은 디스크까지 데이터 처리량이 거의 800MB/sec에 이르렀다. 이와 같은 저장장치 플랫폼은 시장에서 저장장치와 관련된 OEM의 제품 개발주기와 시간을 가속화 시킬 수 있었다. 인피니밴드 저장장치는 서버 클러스터 시장에서 우수한 성능을 인정받고 인피니밴드 저장장치에 대한 많은 요구를 이끌어 내게 되었다. 이와 함께 저장장치 성능에 매우 민감한 응용 서비스에 대해서도 많은 해결책을 제공하였다. 다음은 인피니밴드 저장장치와 관련해서 가격대 성능비가 우수한 응용에 대한 예이다.

- 백업/무디스크 백업
- 미러링/스냅샷/체크포인팅
- 비디오 스트리밍/그래픽
- 재난 복구를 위한 클러스터 스토리지
- 데이터 저장

현재 인피니밴드는 데이터 센터와 같은 고성능의 컴퓨팅 능력과 I/O를 요구하는 분야에서 시장을 주도하고 있다. 또한 10Gb/s의 성능과 전송계층 offload 기술을 갖춘 산업 표준의 기술들 가운데 가장 우수한 가격대 성능비를 나타내고 있다. 결과적으로 고성능, 저비용을 요구하는 관련 업체는 인피니밴드 저장장치 시장을 선점하기 위해 서로 앞 다투어 인피니밴드 저장장치를 내 놓고 있다. 인피니밴드 기반 저장장치 기술을 구현하는 프로토콜은 SCSI 저장장치에 대한 접근 인터페이스 기술을 정의하는 Technical Commitee T10의 SRP(SCSI RDMA Protocol)가 대표적이다.

3.1 SRP(SCSI RDMA Protocol)

인피니밴드망에서 호스트 시스템이 원격지의 저장 장치에 접근을 원할 때 그에 맞는 I/O 프로토콜이 정의 되어야 한다. SRP(SCSI RDMA Protocol)는 원래 ANSI NCITS T10 워킹 그룹에 의해 개발되었다. SRP는 원격의 SCSI 장비를 제어하기 위한 프로토콜로 제안되었고, 인피니밴드 기술의 특성에 맞게 사용되도록 구현되었다. 일반적으로 SCSI 명령어는 저장장치 관련 산업에서는 광범위하게 사용되고, 다양한 타입의 장비에 적용할 수 있다. 현재 블럭 단위 전송 저장장치에 급속도록 적용되고 있는 프로토콜이다.

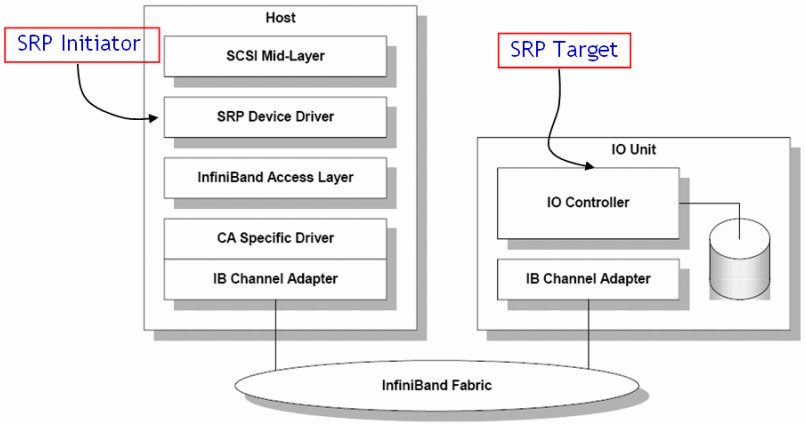


그림 14 SRP 구조

그림 14는 SRP 구조를 도시한 블록 다이어그램이다. SRP는 Initiator가 SCSI 작업을 생

성하고 이를 SRP Target에서 수행하도록 요청하여 서버-클라이언트 모델에 기반한 전송 서비스를 제공하는 프로토콜이다. 또한 SRP와 관련한 모든 통신은 신뢰성을 기반으로 한 연결 서비스를 제공해야 한다. SRP는 메시지 흐름 제어 메커니즘을 제공하는데 이는 기본적으로 인피니밴드 하드웨어에서 제공하는 크레딧 기반(credit based)의 흐름제어 메커니즘에 기반한다. SRP Initiator에 의해 생성되는 작업 요구(work request)에 대한 descriptor를 큐에 넣을 수 있는 개수를 SRP Target이 제한할 수 있도록 한다. 이러한 메커니즘은 Initiator가 여러 개 존재하는 상황에서 필요한 메시지 버퍼를 동적으로 할당할 수 있어 내부 자원을 효과적으로 관리하는데 사용된다. 따라서 제한된 자원에 대한 적절한 이용을 통해 전체 시스템 성능을 향상시킬 수 있다.

SRP 타겟은 모든 데이터 전송을 Initiator 메모리에 직접 읽고 쓰기가 가능하도록 RDMA 기능을 사용한다. Initiator는 자신의 데이터 버퍼를 등록하고, 그 내용을 전송할 SRP 명령어 내에 해당 버서의 주소를 포함시킴으로써 Target이 RDMA를 통한 접근이 가능하다.

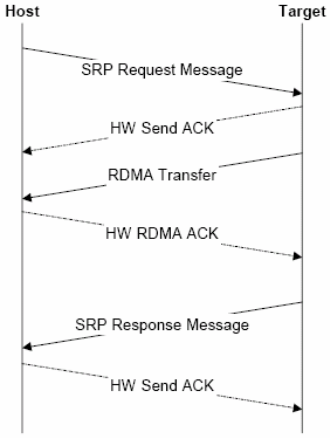


그림 15 SRP I/O 프로토콜

그림 15는 SRP 프로토콜의 I/O 단계를 도시한 것이다.

① Initiator는 SCSI 미들웨어로부터 SCSI 명령어와 LUN(Logical Unit Number) 그리고 데이터 버퍼 디스크립터를 포함한 SRP request 메시지를 생성하고, Target으로 해당 메시지를 전송한다.

② Target은 SRP request 메시지를 받고 메시지에 포함되어 있는 Initiator의 버퍼 공간 주소 정보를 기반으로 RDMA 전송을 수행한다.

③ 타겟은 해당 work request에 대한 완료 내용을 담은 SRP response 메시지를 생성하고 Initiator에게 전송한다.

또한, 초기자는 타겟 상에 존재하는 작업(task)을 무시할 수 있는 SRP 작업 관리에 동작을 수행할 수 있다. 게다가, 타겟은 새로운 미디어 추가와 같은 비동기적으로 발생하는 이

벤트에 대한 메세지를 초기자에게 전송할 수 있다. 게다가, 타겟은 새로운 미디어 추가와 같은 비동기적(asynchronously)으로 발생하는 이벤트에 대한 메시지를 초기자에게 전송할 수 있다.

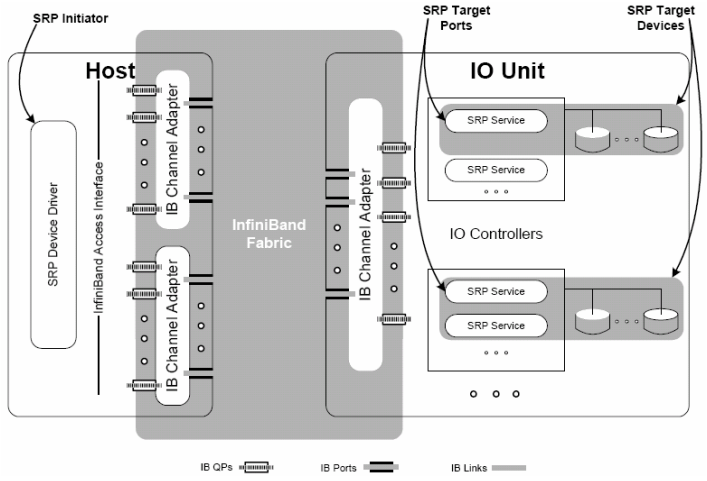


그림 16 SRP 아키텍처와 Infiniband SRP의 연관관계

- 초기자(Initiator)

초기자는 그림 16과 같이 구성되어 있다. 인피니밴드 망과 직접 연결되어 있는 인피니밴드 채널 어댑터(IB Channel Adapter)가 최하단에 위치하며, 그 위로 어댑터 접근을 위한 Verbs 프로바이더 드라이버(InfiniBand Access Interface)가 있다. 이 둘은 상호 긴밀한 관계를 가지며, Verbs 프로바이더는 보통 어댑터 벤더에서 제공한다. 인피니밴드 접근 계층은 다시 두개의 계층으로 나뉜다. 커널 모드와 사용자 모드가 그것이다. 커널 모드는 사용자 모드 하단에 위치하여 Verbs 프로바이더 드라이버와 연결 된다. 사용자 모드는 사용자의 접근을 위해서 커널 모드로 진입하기 전 사용자 수준에서 가용한 인터페이스들을 말한다. SRP 디바이스 드라이버(SRP Device Driver)는 SRP 서비스를 타켓에서 받기 위해서 존재하며, SRP 서비스를 받기 위해서 이루어져야 할 작업들을 처리한다. SRP 디바이스 드라이버와 연결되어 SCSI 기기들에게 명령어를 내리는 SCSI 미들웨어는 SRP 디바이스 드라이버 상단에 위치한다. 실제 처리하는 명령어 집합의 수준은 저수준들이다.

-타켓 (Target)

타겟은 SRP 서비스를 제공하는 서버이다. 하지만 서버임에도 불구하고 구성은 그림 10의우측 과 같이 간단하다. 초기자와 마찬가지로 인피니밴드 망과 직접 닿아 있는 인피니밴드 어댑터가 최하단에 위치한다. SRP 타겟은 인피니밴드 어뎁터로 들어오는 프레임들을 읽어 들인다. 그리고 읽어 들인 프레임을 SRP 프로토콜에 맞게 파싱하여 각각의 명령어를 입출력 제어기로 넘겨준다. 입출력 제어기(I/O Controller)는 저수준의 SCSI 명령어들을 처리하게 된다. SRP 헤더로 인캡슐레이션(encapsulation)되어 도착한 프레임을 SRP 타겟에서 디캡슐

레이션(decapsulation)하여 입출력 제어기로 넘기면 그때 SCSI 저장 장치에 저수준 명령어가 내려진다. SRP 서비스는 기본적으로 서버/클라이언트 모델을 하고 있다. 하지만 인피니밴드 구조를 가졌기 때문에 신뢰성을 보장 받을 수 있다.

SRP 서비스는 기본적으로 명령어를 일정 큐에 저장을 하고 큐에서 순차적으로 프레임을 읽어 들이는 방법을 사용한다. 즉, 비동기적으로 이벤트가 일어나게 된다. 많은 수의 요청이 오더라도 큐에 저장이 되고, 저장된 큐는 순서대로 SRP 서버에서 읽어 들여 처리를 하게 된다.

3.2 Mellanox SRP MTD1000의 성능

Mellanox가 Storage Networking World 2004에서 제시한 인피니밴드 저장장치 플랫폼인 MTD 1000은 SCSI 하드 디스크를 최대 15개까지 확장가능하며, Linux 운영체제에 Dual 3GHz Xeon CPU를 사용한다. PCI Express 버스를 사용하여 10G 인피니밴드 HCA와 연결되며 PCI-X Bridge를 통해 SATA 하드디스크가 연결된다(그림 17).

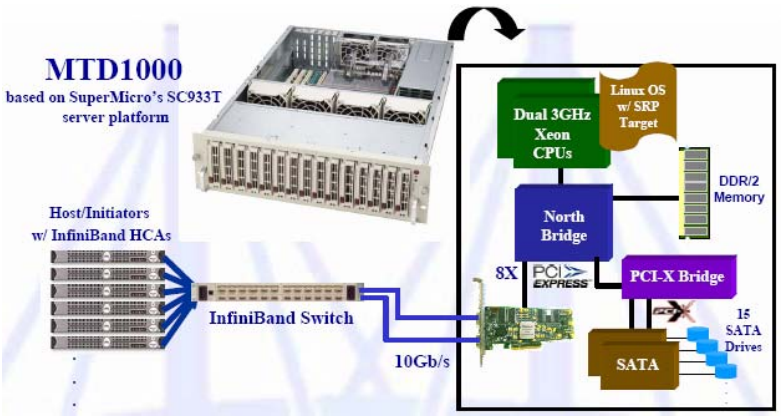


그림 17 Mellanox MTD1000 아키텍처

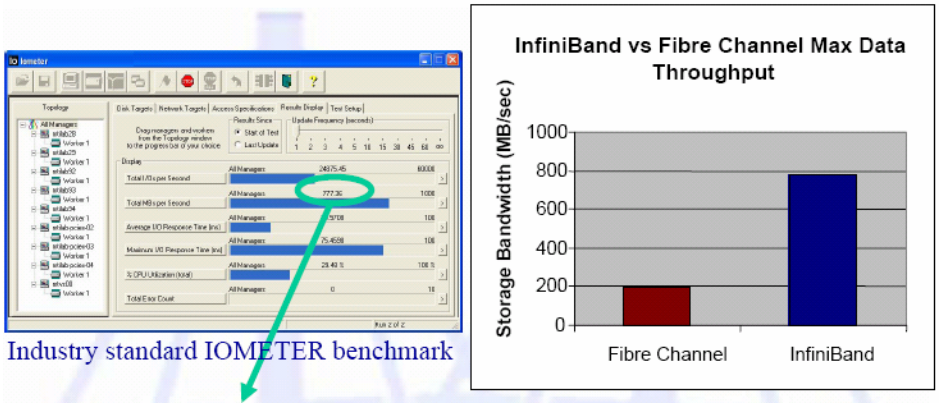
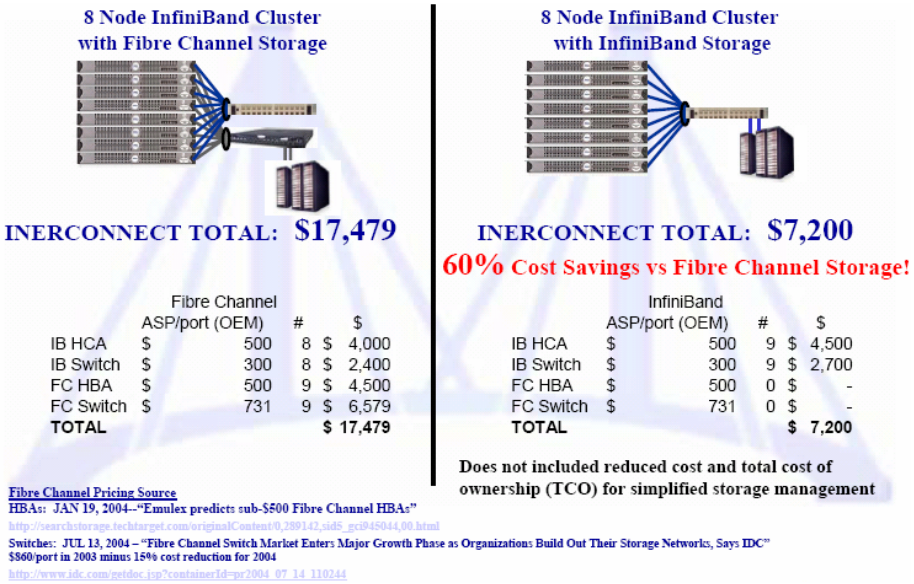


그림 18 SRP와 FC 성능 비교

위 그림 18은 Mellanox 사에서 자사의 Infiniband 기반 SRP의 성능과 Fibre Channel을 이용한 스토리지 네트워크의 성능을 비교해 놓은 것이다. 이에 따르면 Infiniband 저장장치 플랫폼은 디스크까지 데이터 처리량이 거의 800MB/sec에 이른다. 반면, FC의 성능이 200MB/sec에 불과하다고 나타나 있으나, 앞서 그림 7의 FCIA Roadmap 의 자료에서 (http://www.fiberchannel.org) 제시한 바와 같이 2006년 현재 FC기반 저장장치 플랫폼의 경우에도 800MB/sec 의 성능을 나타내는 제품을 판매중이므로 성능상에서는 FC와 SRP의 우열을 가리기 힘들다고 보아야 할 것이다.



반면, 가격대 성능비 측면에서는 Infiniband상에서 SRP를 사용하는 것이 파이버 채널 기반 스토리지 플랫폼보다 훨씬 유용한 것으로 보인다. FC의 경우 기존 네트워크에 추가로 HBA와 FC Switch를 통해 별도의 네트워크를 구축하는데 드는 비용이 상당하다. 하지만, Infiniband의 경우 SRP는 소프트웨어적으로 처리 가능하므로 하드디스크를 포함하는 SRP Target 디바이스만 구매하면 더 이상의 추가 비용 없이 스토리지 네트워크를 구축할 수 있다.

4. SRP **설치 및 성능 테스트 결과**

4.1 Experiment Environment

Intel Xeon 2.80 GHz 서버 4대를 사용하고 있으며, 이 서버 4대는 PCI-X를 사용하는

Topspin HCA 카드를 장착하고 있다. 각 Topspin HCA 카드는 2개의 physical port를 가지고 있으며, 각 노드의 1번 port는 Mellanox Switch를 통해 연결되어 있다. 실험의 편의와 Infiniband와의 성능비교를 위해 4대의 서버는 Gigabit Ethernet으로 연결되어 있다.

처음 실험 시에는 Topspin HCA + Topspin Switch 장비를 사용하였으나, 이후 Topspin HCA + Mellanox Switch 를 사용하게 되었다. Topspin Switch와 Mellanox Switch는 Subnet Manager (SM)를 운영하는 방식이 다르다. Topspin Switch의 경우(Topspin 120 Server Switch), 그 Feature 중 하나로 Intelligent switch with embedded subnet manager를 제공하고 있다. 이것은 Infiniband spec에서는 switch에 연결되어 있는 HCA들을 detect하고 management 역할을 해주는 subnet manager를 한 서버에서 수행해주어야 하는 것과는 달리, 이 subnet manager를 switch에 embedded시켜놓아서 Intelligent gateway가 subnet manager의 역할을 해주게 된다.

따라서 Mellanox Switch를 사용하게 되면서, Subnet Manager를 실행시켜주어야 했다. 처음에는 Topspin device driver에는 Subnet Manager가 포함되어 있지 않을 것이라 생각되어, OpenIB에서 제공하는 device driver를 설치하고자 하였다. 하지만 OpenIB의 Infiniband device driver 중 subnet manager module을 설치할 경우, 다른 module들과의 dependency 문제가 발생하였고, 어쩔 수 없이 verb를 비롯한 몇 개의 dependency가 존재하는 module들만을 선택해서 설치하였는데 결과 정상적으로 작동되지 않았다.

그러던 중에 Topspin device driver에 subnet manager module이 포함되어 있음을 발견하였다. 이미 설치된 OpenIB의 module들을 모두 제거하고 Topspin에서 제공하는 OpenSM을 사용하여 정상적으로 Infiniband가 작동하는 것을 여러 실험을 통해 확인할 수 있었다.

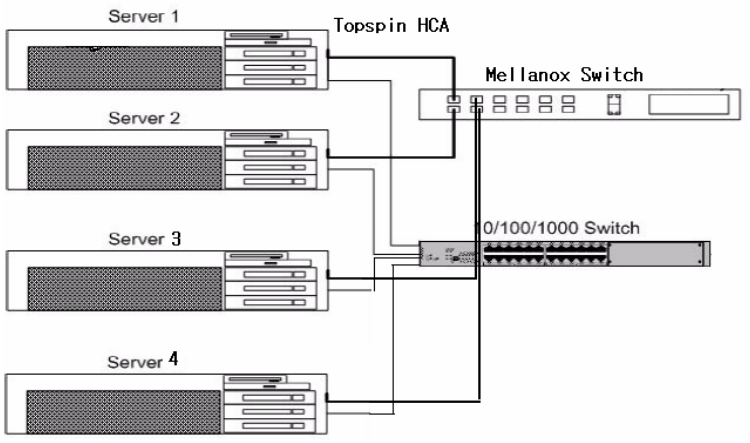


그림 19 Experiment environment

4.2 Topspin & Mellanox Infiniband Device Driver

그림 20 은 Infiniband의 spec에서 정의된 Infiniband module들의 개략적인 그림이다. Topspin device driver와 Mellanox Infiniband Gold Distribution(IBGD)는 모두 이 Infiniband spec을 따르고 있지만, HCA와 Switch 그리고 device driver의 내부적인 동작은 조금씩 다른 것으로 추측된다. (실제로 그 내부 작동방식과 implementation을 확인할 순 없었다. 하드웨어의 내부 작동방식은 알 수 없으며, device driver의 경우 Topspin device driver는 source code가 공개되어 있지 않다. 다만, Topspin device driver와 IBGD의 module을 혼용해서 사용한 경우, 정상적으로 동작하지 않았고, 이를 바탕으로 내부의 작동방식이 다를 것이라 추측할 수 있다.)

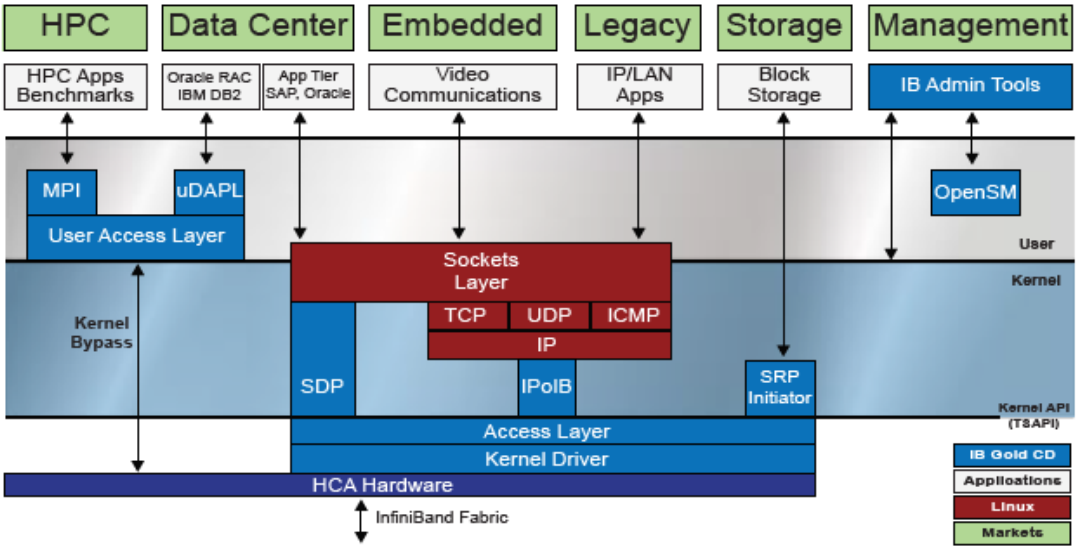


그림 20 Infiniband 전체 구조

Topspin device driver

3개의 rpm file로 구성되어 있다.

topspin-ib-rhel4-3.1.0-113.i686.rpm -

topspin-ib-mod-rhel4-2.6.9-11.EL-3.1.0-113.ia64.rpm -

topspin-ib-mpi-rhel4-3.1.0-113.i686.rpm - MPI

IBGD

기본적인 device driver 외에 여러가지 Administration Tool 등을 같이 제공하고 있다.

ib-verbs-1.8.2-2.6.9_5.ELsmp.i386.rpm – verbs interface

ib-cm-1.8.2-2.6.9_5.ELsmp.i386.rpm – Connection Manager

opensm-1.8.0_1-2.6.9_5.ELsmp.i386.rpm – Subnet Manager

ib-ipoib-1.8.2-2.6.9_5.ELsmp.i386.rpm – IP over IB

ib-dapl-1.8.2-2.6.9_5.ELsmp.i386.rpm – DAPL
ib-sdp-1.8.2-2.6.9_5.ELsmp.i386.rpm – Socket Direct Protocol
ib-srp-1.8.2-2.6.9_5.ELsmp.i386.rpm – SRP

ibadm-1.8.1-2.6.9_5.ELsmp.i386.rpm – IB Admin Tools
mpich_mlx_gcc-0.9.5_mlx1.0.3-2.6.9_5.ELsmp.i386.rpm - MPI
mst-i686-4.3.1-2.6.9_5.ELsmp.i386.rpm – Mellanox Software Tools
pdsh-2.8-1.i386.rpm – PDSH
pdsh-debuginfo-2.8-1.i386.rpm
pdsh-rcmd-rsh-2.8-1.i386.rpm
pdsh-rcmd-ssh-2.8-1.i386.rpm

Topspin device driver에는 SRP Target이 포함되어 있지 않다. 최근 Mellanox에서는 SRP Target을 공개하였으며, Mellanox 홈페이지를 통해 SRP Target이 포함되어 있는 IBGD w/ SRP Target과 포함되어 있지 않은 IBGD w/o SRP Target 두 버전의 IBGD가 제공되고 있다.

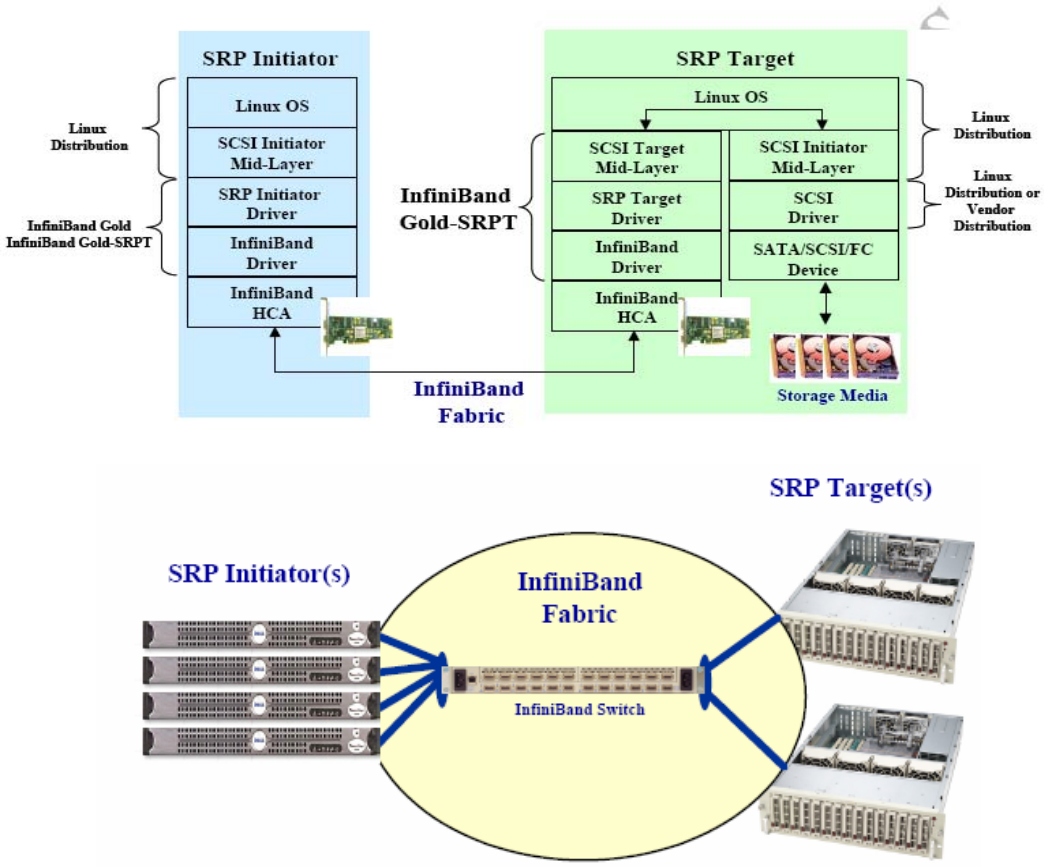


그림 21 SRP Target & SRP Initiator 구조

4.3 IBGD SRP Installation w/o uninstalling topspin device driver

먼저 기존의 Topspin device driver 위에 SRP Target 만을 설치하고자 하였다. package 사이에는 dependency가 존재하므로, 필수적으로 IBGD의 ib-verb를 비롯한 몇 개의 module을 추가적으로 설치하였다.

rpm –ivh ib-verbs-1.8.2-2.6.9_5.ELsmp.i386.rpm
rpm –ivh ib-cm-1.8.2-2.6.9_5.ELsmp.i386.rpm
rpm –ivh opensm-1.8.0_1-2.6.9_5.ELsmp.i386.rpm
rpm –ivh ib-dapl-1.8.2-2.6.9_5.ELsmp.i386.rpm
rpm –ivh ib-srp-1.8.2-2.6.9_5.ELsmp.i386.rpm

설치한 후, 리부팅해 본 결과 정상적으로 부팅되지 않았다. 기존에 설치되어 있는 Topspin device driver와 IBGD의 device driver가 충돌되는 것으로 추측된다. (원격에서 실행해 보았기 때문에 정확한 원인은 확인할 수 없었다.

이번에는 dependency를 무시하고 Topspin device driver 위에 바로 SRP만을 설치하고 module을 load해 보았다. 그 결과는 그림 22, 그림 23과 같다. dmesg 결과, Topspin 과 IBGD의 module들이 사용하는 Symbol이 다르기 때문에, SRP는 Topspin device driver 위에서 load될 수 없었다.

```
[root@xeon5 RPMS]# rpm -ivh ib-srp_target-1.8.2-2.6.9_5.ELsmp.i386.rpm --nodeps
Preparing...                ########################################### [100%]
   1:ib-srp_target          ########################################### [100%]
[root@xeon5 RPMS]#
[root@xeon5 RPMS]#
[root@xeon5 RPMS]# modprobe ib_srp_target
WARNING: Error inserting scsi_target (/lib/modules/2.6.9-5.ELsmp/kernel/drivers/
FATAL: Error inserting ib_srp_target (/lib/modules/2.6.9-5.ELsmp/kernel/drivers/
[root@xeon5 RPMS]# dmesg
```

그림 22 SRP Target install 결과

```
[root@xeon5 RPMS]# rpm -ivh ib-srp-1.8.2-2.6.9_5.ELsmp.i386.rpm --nodeps
Preparing...                ########################################### [100%]
   1:ib-srp                 ########################################### [100%]
[root@xeon5 RPMS]# modprobe ib_srp
FATAL: Error inserting ib_srp (/lib/modules/2.6.9-5.ELsmp/kernel/drivers/infinib
[root@xeon5 RPMS]# dmesg
```

그림 23 SRP Initiator install 결과

4.4 IBGD Full Installation w/ uninstalling topspin device driver

결국 Topspin의 device driver를 모두 제거하고, IBGD로 설치하였으며 SRP Initiator와 SRP Target 모두 정상적으로 작동함을 확인할 수 있었다.

Topspin device driver 제거

rpm –e topspin-ib-rhel4-3.1.0-113.i686.rpm

rpm –e topspin-ib-mod-rhel4-2.6.9-11.EL-3.1.0-113.ia64.rpm

rpm –e topspin-ib-mpi-rhel4-3.1.0-113.i686.rpm

IBGD 설치

InfiniBand Gold Distribution (IBGD) Software Installation Menu

1) View IBGD Installation Guide
2) Install IBGD Software
3) Show Installed Software
4) Configure IPoIB Network Interface, IBADM Server, and OpenSM Server
5) Uninstall IBGD Software
6) Build IBGD Software RPMs

Q) Exit

Select Option [1-6]:2

Select IBGD Software

1) Typical (ib-verbs, ib-ipoib, opensm, ibadm and mpi)
2) Minimal (ib-verbs only)
3) All packages (ib-verbs, ib-ipoib, ib-cm, ib-sdp, ib-dapl, ib-srp, opensm, ibadm, mpi, pdsh)
4) Customize

Q) Exit

Select Option [1-4]:3

The following compiler(s) on your system can be used to build/install MPI: gcc

Do you wish to create/install an MPI RPM with gcc? [Y/n]:

Following is the list of IBGD packages that you have chosen
(some may have been added by the installation program due to package dependencies):
ib-cm
ib-dapl
ib-ipoib
ib-sdp
ib-srp
ib-verbs
pdsh
opensm
mpi_osu
ibadm

WARNING: This installation program will remove any previously installed IB packages on your machine.

Do you want to continue? [Y/n]:

The default installation directory for IBGD Software is /usr/local/ibgd
Do you want to continue? [Y/n]:

Removing previous IBGD Software installations
...
...

SRP Initiator 설정

1. boot시에 SRP initiator를 load하도록 설정
   vi /etc/infiniband/openib.conf

그림 24 /etc/infiniband/openib.conf

SRP_LOAD를 yes로 설정해 놓는다.

2. boot시에 load하지 않은 상태에서, load하는 방법

modprobe ib_srp

3. 제거시

modprobe –r ib_srp

SRP Target 설정

1. boot시에 SRP initiator를 load하도록 설정

SRP Initiator 설정과 비슷함

2. boot시에 load하지 않은 상태에서, load하는 방법

modprobe ib_srp_target

설정 전 & 후

[root@xeon3 ~]# fdisk -l

Disk /dev/sda: 73.4 GB, 73407900160 bytes

255 heads, 63 sectors/track, 8924 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot Start End Blocks Id System

```
/dev/sda1   *           1         3824    30716248+  83  Linux
/dev/sda2            3825         4079     2048287+  82  Linux swap
/dev/sda3            4080         8924    38917462+   5  Extended
/dev/sda5            4080         8924    38917431   83  Linux
```

```
[root@xeon3 IBGD-1.8.2]# modprobe ib_srp
[root@xeon3 IBGD-1.8.2]# fdisk -l

Disk /dev/sda: 73.4 GB, 73407900160 bytes
255 heads, 63 sectors/track, 8924 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         3824    30716248+  83  Linux
/dev/sda2            3825         4079     2048287+  82  Linux swap
/dev/sda3            4080         8924    38917462+   5  Extended
/dev/sda5            4080         8924    38917431   83  Linux

Disk /dev/sdb: 73.4 GB, 73407900160 bytes
255 heads, 63 sectors/track, 8924 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1         8793    70629741   83  Linux
/dev/sdb2            8794         8924     1052257+  82  Linux swap
```

4.5 Experiment Result

4.5.1. Bonnie

Bonnie는 각각 putc(), getc()와 같은 함수를 통해 I/O의 성능을 알아보는 Benchmark 프로그램이다. Size는 default인 100Mb로 실험하였다.

실험 결과 Write의 경우에는 SRP가 NFS에 비해 압도적으로 좋은 성능을 보였으며, Local에 거의 근접한 성능을 보임을 확인할 수 있었다. 반면 Read의 경우에는 Local, SRP, NFS가 거의 비슷한 성능을 보였으며, 오히려 NFS가 높은 성능을 보이기도 했다. 매 실험

마다 어느 정도의 오차가 존재하므로, 거의 비슷한 성능을 보인다고 볼 수 있다. Random Seek의 경우에는 Write와 마찬가지로 Local = SRP >> NFS 의 성능을 보이고 있다.

Bonnie의 경우에는 매우 간단한 test여서, 여러 size의 read, write에 대해 test해보기 위해 Iozone으로 Benchmark을 해 보았다.

	Sequential Output			Sequential Input		Random
	Per Char	Block	Rewrite	Per Char	Block	Seeks
Local	40105	280825	381400	43231	592123	37433.6
SRP	39970	280781	383391	43088	588861	37420.2
NFS	17502	29740	28443	44140	614015	3303.8

표 1 Bonnie 결과 (K/sec)

4.5.2. Iozone

각각의 실험 결과 Graph는 SRP, NFS, Local의 순서대로 나열하였다. (각 그래프마다 scale이 조금씩 다르기 때문에 scale을 주의해서 보아야 한다.)

또 NFS의 경우에는, 시간이 지나치게 오래 걸려서, 131072까지만 test하였다.

4.5.2.1. Write

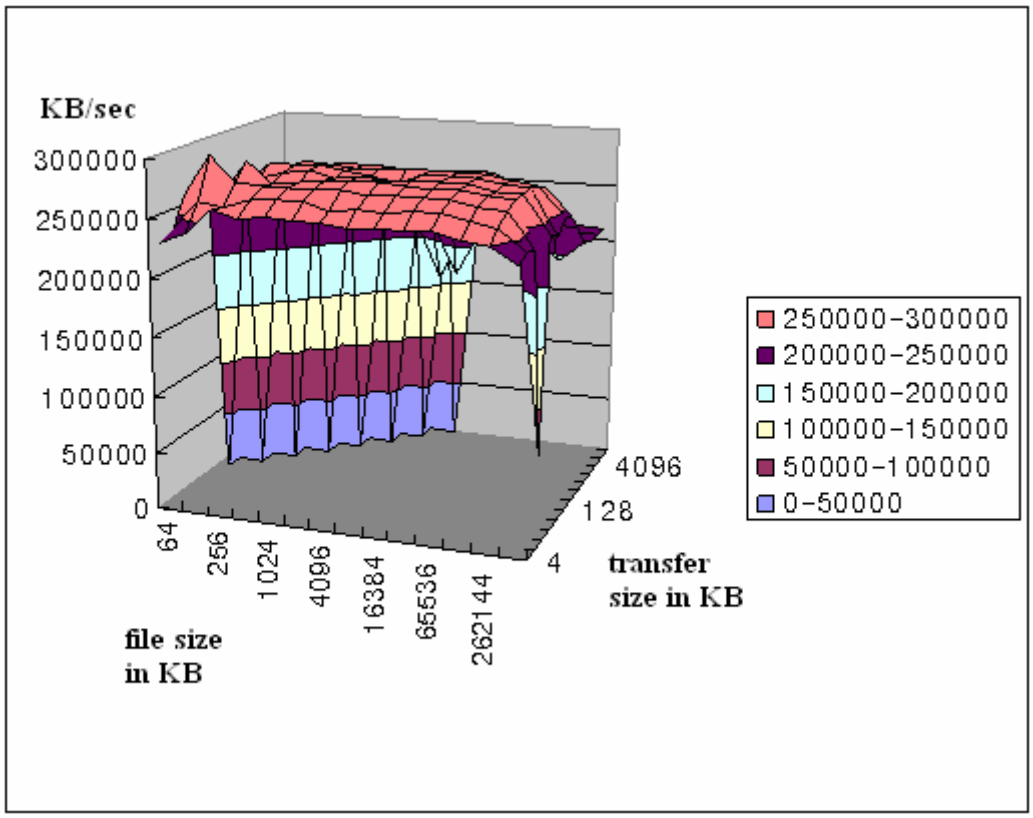


그림 25 SRP Write

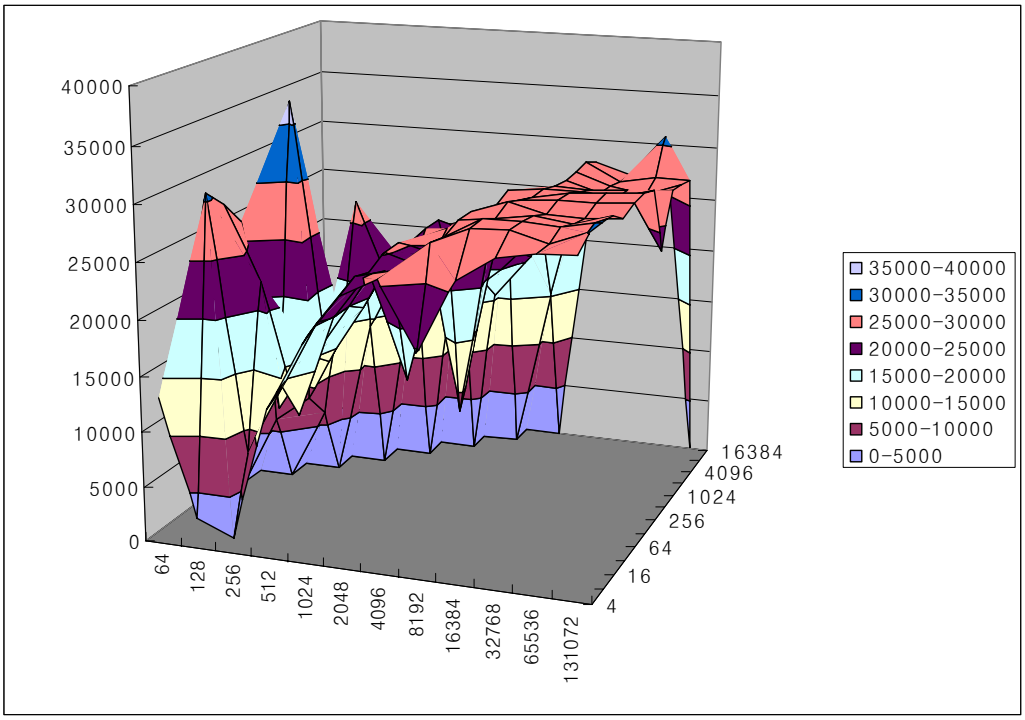


그림 26 NFS Write

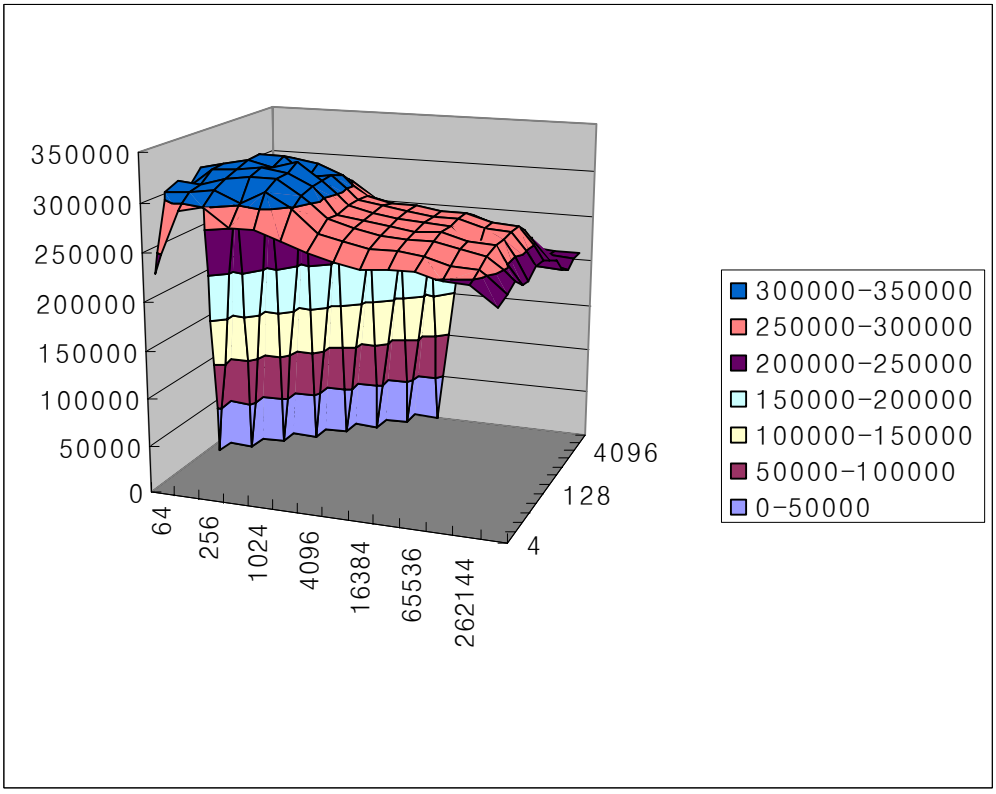


그림 27 Local Write

5.2.2. Read

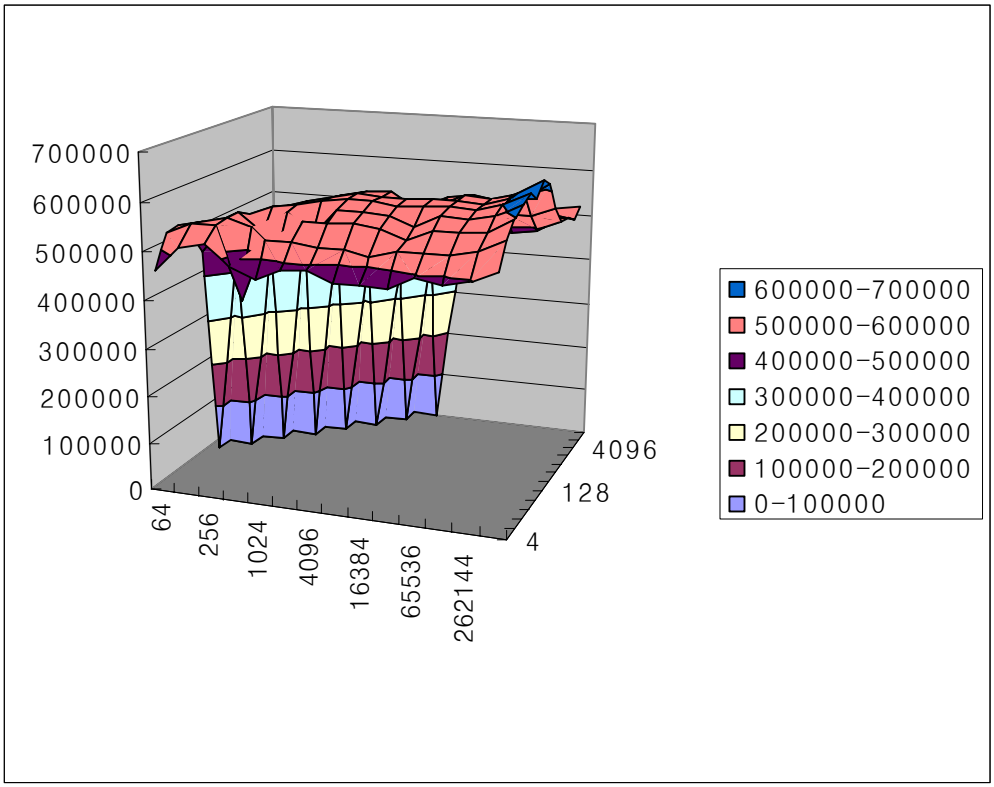


그림 28 SRP Read

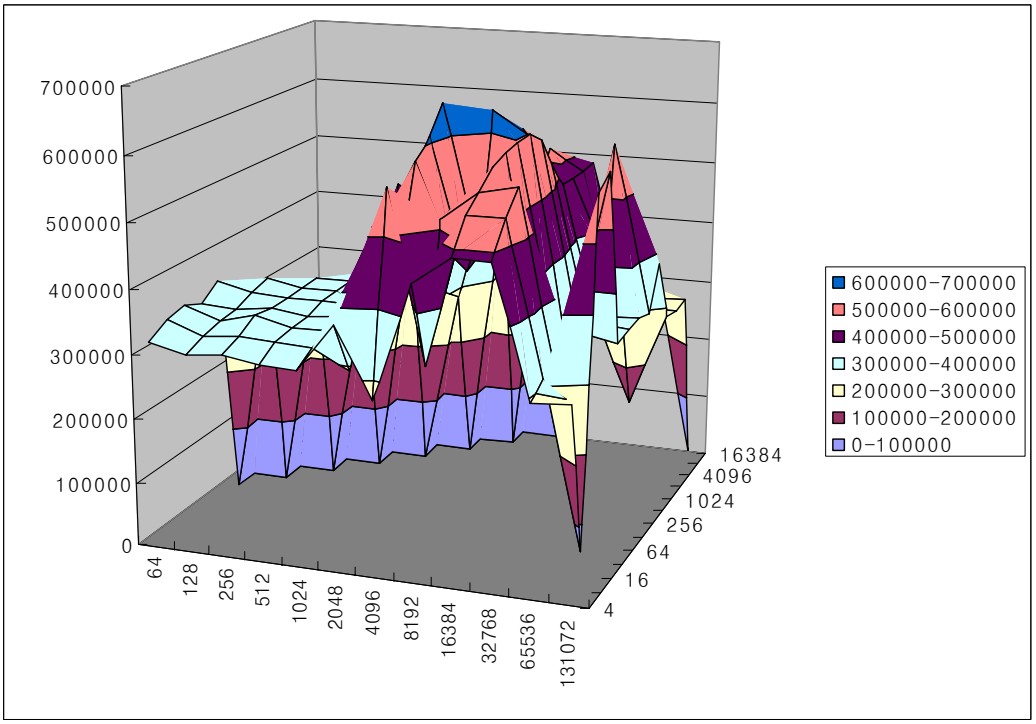


그림 29 NFS Read

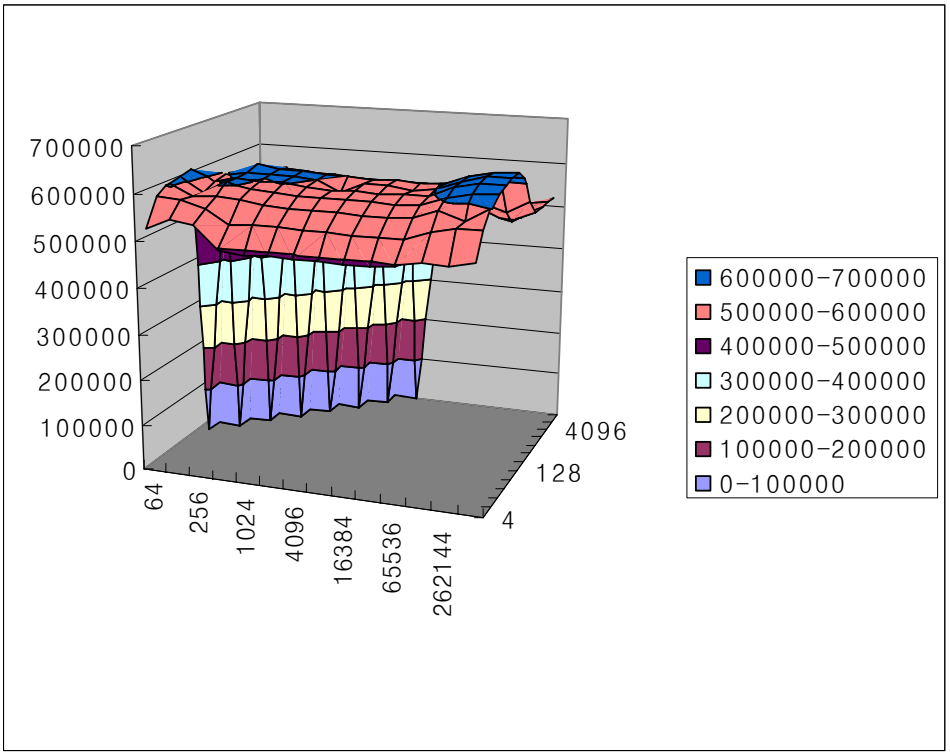


그림 30 Local Read

5.2.3. Random Read

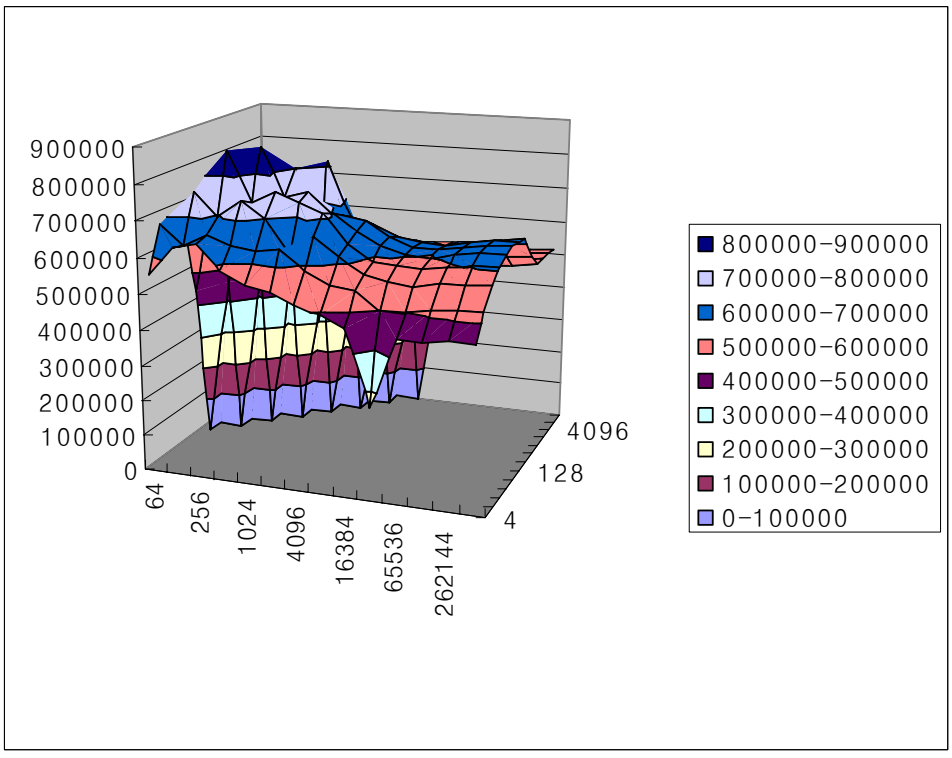


그림 31 SRP Random Read

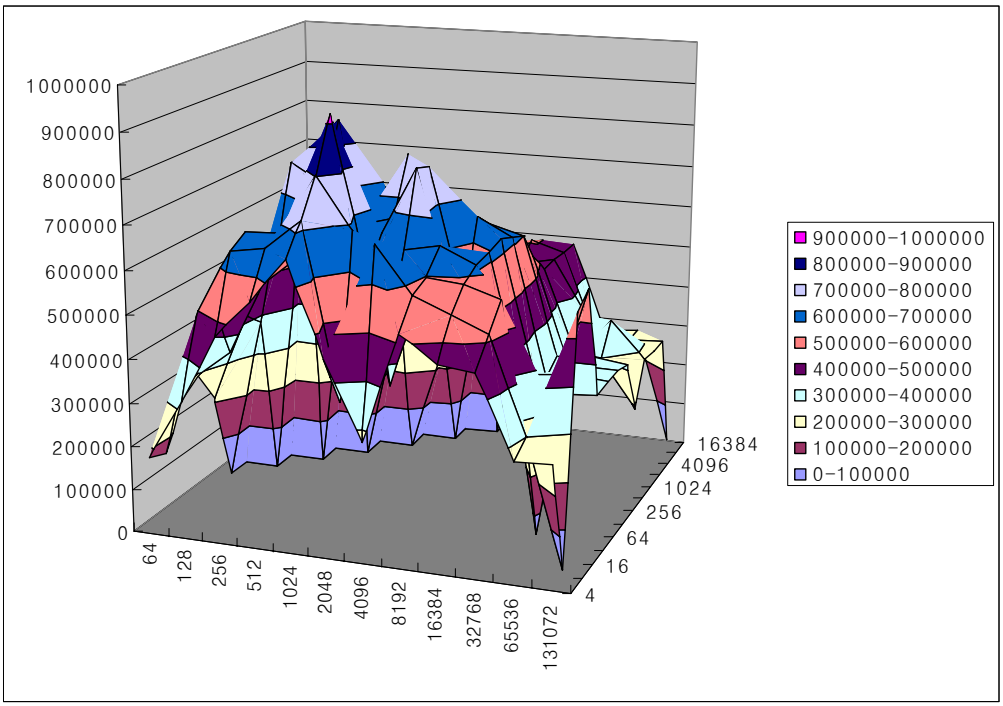


그림 32 NFS Random Read

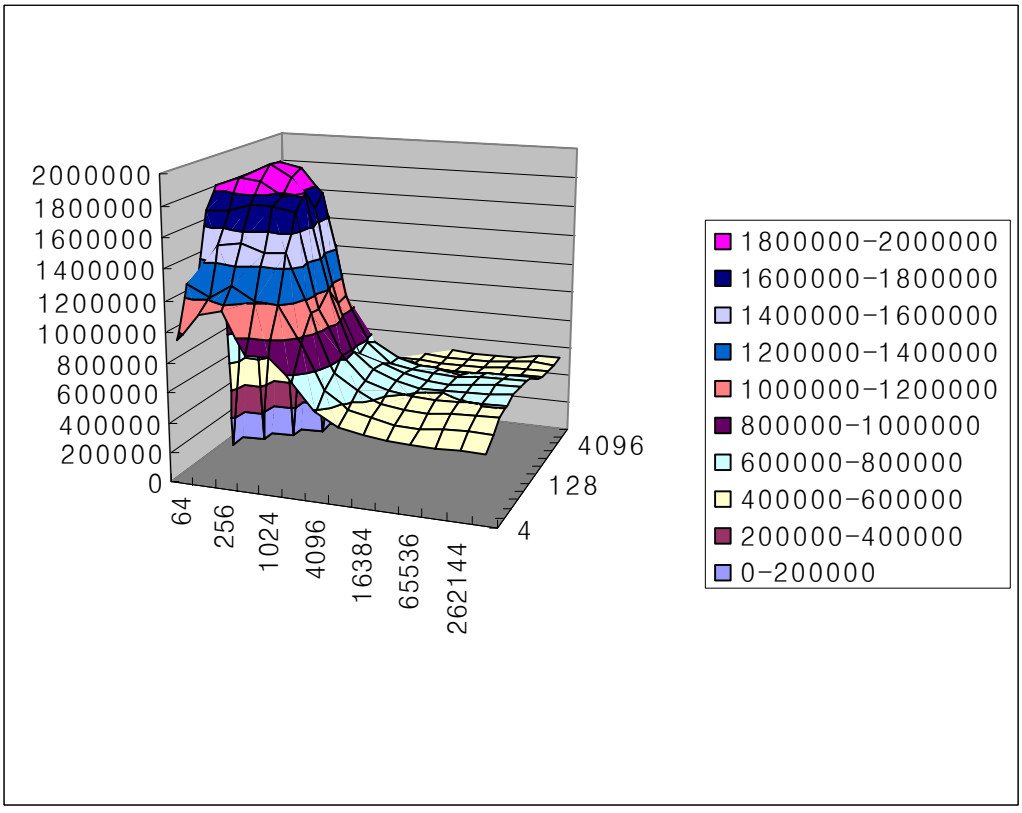


그림 33 Local Random Read

5.2.4. Random Write

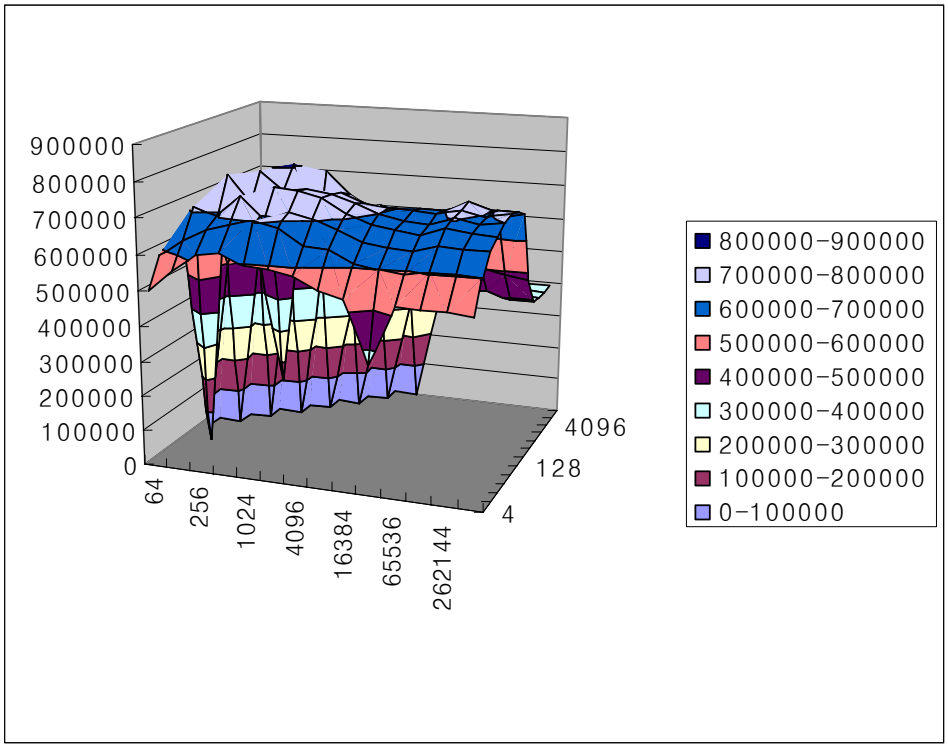


그림 34 SRP Random Write

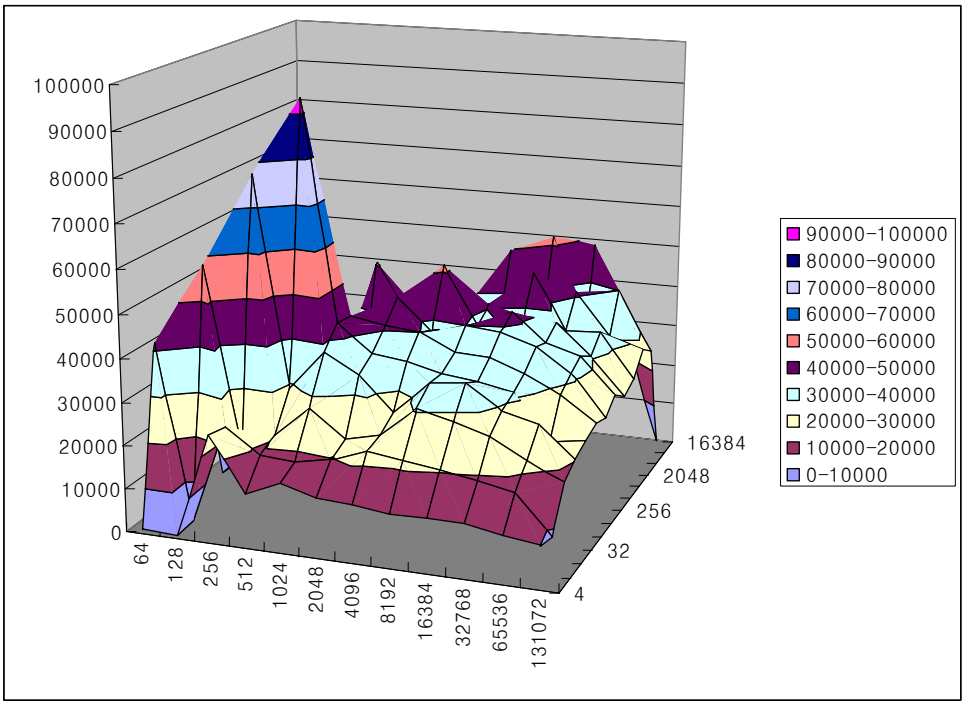


그림 35 NFS Random Write

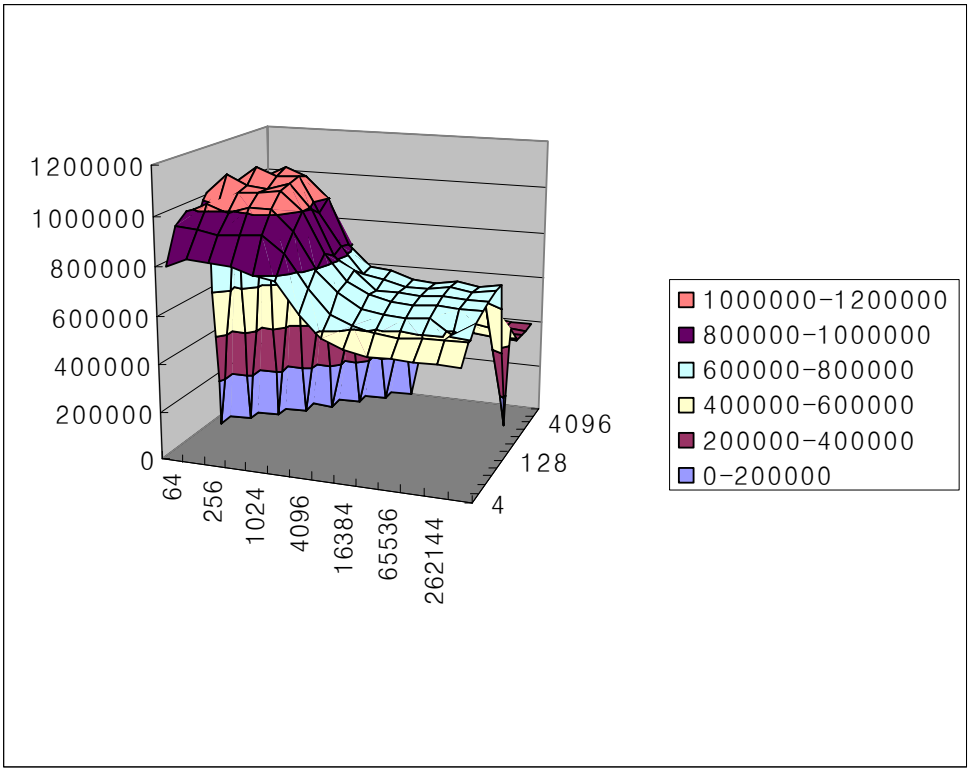


그림 36 Local Random Write

위의 결과들을 살펴보면, 먼저 Write의 경우에는 예상한 바와 같이 SRP는 Local access에 거의 근접한 성능을 보이며, NFS는 그에 훨씬 미치지 못하는 성능을 보인다.

Read의 경우에는 SRP와 Local access는 역시 비슷한 성능을 보이며, NFS는 Variation이 크지만 그래도 SRP에 근접한 성능을 보여주는 것을 볼 수 있다.

5. 결론

지금까지 살펴본 네트워크 스토리지 시스템(networked storage system)은 모두 다양한 고유의 접근 방법을 취하고 있다. 하지만, 가상화(virtualization)의 관점에서 보면 공통적이다. 즉, 별도의 네트워크에서 사용되는 SCSI 명령어 집합을 IP 네트워크 또는 Infiniband 네트워크에서 전송할 수 있도록 추가적인 프로토콜 레이어를 도입하는 것 자체가 바로 가상화이다. 지금까지 살펴본 SRP, iSCSI, FCIP, iFCP 등의 프로토콜은 모두 이런 가상화 방법을 취하는 것이라 할 수 있다. 가상화는 구현수준에 따른 분류에 따라 그림 19와 같이 3가지로 나뉘어 진다.

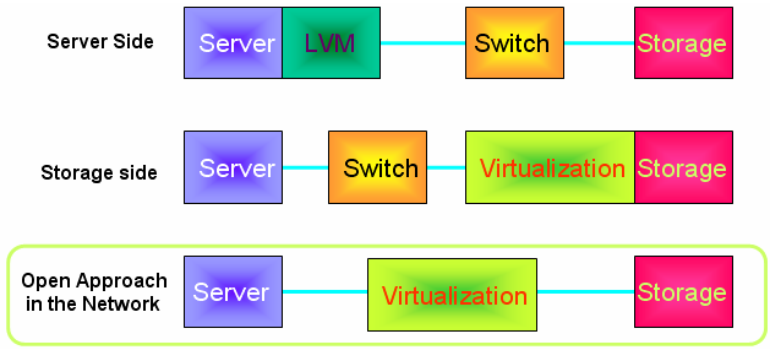


그림 37 가상화 레이어의 위치

가상화(virtualization)는 SAN의 여러 다른 위치에서 구현될 수 있는데 server side, storage side, 그리고 network side에서의 3가지 수준에서 실현 가능하다. 서버 벤더인 경우 논리적 볼륨 관리자(Logical Volume Manager: LVM)를 서버측에 두어 서버 안에 가상화를 구현하려 한다. 스토리지 업체의 경우 스토리지 측에 가상화 레이어를 두어 구현하려 한다. 하지만, 특정 스토리지나 서버 업체에 구애받지 않으며 스토리지 서비스의 가상화를 얻으려는 접근방법은 아무래도 네트워크 상에 구현하는 방법이 유리하다. 이 방법은 엔드유저에게 자신들의 비즈니스 요구사항에 적합한 서버나 스토리지 업체를 임의로 선정할 수 있는 유연성을 제공한다.

이와 같은 가상화는 비용측면에서 상당한 장점을 제공한다. 실제로, 저장장치에 대한 관리 비용은 저장장치 구매 비용의 6~8배에 이른다. 그러나 기존 NAS/SAN을 활용한 스토리지 네트워크 보다 가상화 방법을 도입함으로서 상당한 TOC의 절감을 가져온다(그림 20).

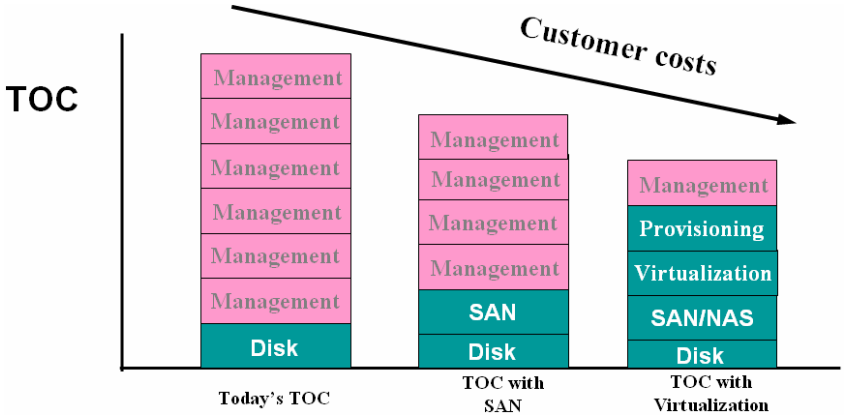


그림 38 가상화와 비용절감

이러한 상당한 비용절감의 원인은 스토리지 네트워크를 구축하는데 있어 별도의 네트워크를 구성할 필요가 없다는 장점으로 인해 우선 구매비용이 줄어드는데다, 기존의 네트워크 유지 관리 소프트웨어, 그리고 네트워크 보안 소프트웨어 및 프로토콜을 그대로 활용할 수 있기 때문이다. 그 밖에도 관리 편의성을 비롯한 여러가지 측면에서 가상화는 상당한 장점을 가져다 준다.

향후 이러한 가상화 개념은 스토리지의 가상화, 관리 인터페이스에 대한 가상화 측면에서 더욱 발전되어야 한다. 우선, 스토리지의 가상화 측면에서 물리적으로 서로 다른 저장장치들 사이에도 공통의 스토리지 풀(storage pool)로 통합관리 할 수 있어야만 명실상부한 가상화 접근방법이라 할 수 있다. 관리 인터페이스 측면에서는 고객으로 하여금 스토리지 자원의 관리, 검색(discover), 모니터링 등에 있어 공통의 단일 인터페이스를 통해 서비스되어야 하는 측면이 있다.

참고문헌

[1] InfiniBand Trade Association. InfiniBand Architecture Spec., Release 1.1, October 24 2004.

[2] Mellanox Technologies. Mellanox InfiniBand Storage, July 2004.

[3] RDMA Consortium. iSCSI Extensions for RDMA (iSER) and Datamover Architecture for iSCSI (DA) Specifications, 2004.

[4] SCSI RDMA Protocol, Software Architecture Specification Revision- Draft 2, Intel, 2002

[5] Technical Committee T10. SCSI RDMA Protocol, 2002.

[6] K. Z. Meth and J. Satran. Design of the iSCSI Protocol. In 20th IEEE Smposium on Mass Storage Systems, 2003

[7] J. C. Mogul. TCP Offload Is a Dumb Idea whose Time Has Come. In 9th Workshop on Hot Topics in Operating Systems (HotOS IX), May 2003.