미디어용 스토리지

□ 김정림 / 한국 DDN

Ⅰ. 서론

스토리지는 과거 아날로그 작업 프로세스를 백업하는 개념의 2차 플랫폼이었다. 시간이 지남에 따라 프로세스의 기간 단축과 투자대비 효율을 고려하고 자산을 디지타이징하여 보관함에 따라 스토리지는 메인 플랫폼으로 자리잡기 시작하였다. 향후 연구소나 미디어 기업이 자산을 보관하고 작업자들이 무한 신뢰하고 작업할 수 있는 워크스페이스(Work Space)개념으로 스토리지의 역할이 바뀌게 된다. 이러한 용도의 스토리지를 비정형 데이터형 스토리지라고 하는데 기존 정형 데이터용 스토리지들과는 개발의 포인트나 특성, 안정성을 구현하는 방법이 다르다. 본론에서 비정형 데이터용 스토리지는 정형 데이터용 스토리지와 어떻게 다른지에 대해서 알아본다.

Ⅱ. 본론

데이터가 생성되고 가공되는 작업 환경은 기존 아날로그 데이터의 세상에 비해 변화 속도나 확장성 측면에서 예측을 불허한다. 기존 Web 1.0 기반의 웹사이트를 운영하는 서비스 업체에서 데이터를 백업할 경우에는 수 테라바이트의 용량으로 사이트를 운영 가능했지만, 현재 Web 2.0 기반의 인터렉티브(Interactive)한 웹 기반의 데이터들은 페타바이트에 이른다. 단순한 예로 웹기반의 비즈니스에서 볼 수 있듯이 기존의 스토리지 플랫폼으로는 확장성이나 서비스를 위한 성능면에서 상당한 한계가 있다. 과연 무엇이 이러한 차이를 있게 했으며, 앞으로 어떻게 대처해 가야 하는지 해결책을 제시하는 것이 시스템 관리자가 해야 할 숙제이다.

이러한 데이터의 기하급수적 팽창은 기술의 발전으로부터 시작이 되었다. 우주항공연구, 데이터 웨

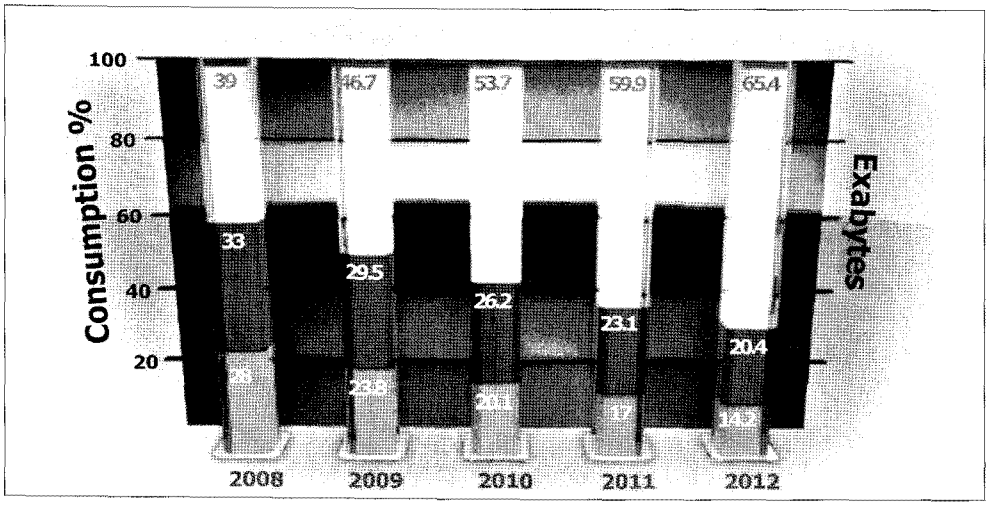
어 하우스, 웹 2.0 비즈니스, 방송 환경(Rich Media), 의료 영상 분야가 바로 폭발적으로 데이터가 증가되는 분야이며, 그 원인에는 데이터 특성의 변화에서부터 시작된다. 현재 데이터는 크게 비정형 데이터와 정형 데이터로 분류가 된다.

1. 비정형 데이터인 미디어 데이터

데이터의 종류는 정형 데이터(Structured Data) 및 비정형 데이터(Unstructured Data)로 구분이 된다. 정형 데이터(Structured Data)는 데이터의 구조 x,y 행렬의 테이블 구조로 사용자가 요구하는 데이터가 일정한 패턴으로 규정되어 있다. 정형 데이터는 질의 및 응답 (Query and Report)의 형태로 데이터의 추출하며 DMBS와 같은 툴을 사용한다. 비정형 데이터는 (Unstructured Data)는 사용자가 원하는 데이터를 추출하기 위해서는 전체 데이터에 대한 검색을 해야 한다. 데이터베이스와 같이 규정이 되는 데이터에 대해서는 정의(Definition)가 가능하고 텍스트화가 가능한 반면 비정형 데이터는 텍스트화 및 정의(Definition) 자체가 불가능하다. 이러한 비정형 컨텐츠를 신속하게 검색을 하기 위해서 최근 컨텐츠에 태깅(Tagging)을 하거나, 검색어(Key word) 또는 메타 태크(Meta tag)를 삽입한다. 비정형 데이터의 종류로는 대용량 이미지 데이터 (법률, 금융 스캔 문서, 선박 및 자동차 도면, 유전자 지도, 방송 미디어 파일, 영화 시퀀스 파일, DB의 백업 데이터), 슈퍼컴퓨팅 데이터 (기상, 천문 데이터 등)가 있다.

IDC의 자료에 의하면 2012년까지 비정형 데이터의 수요가 폭발적으로 늘어나는 것으로 조사된다.

비정형 데이터의 특성으로는 고속, 대용량이 특징이며 비정형 데이터를 생성, 저장, 가공하기 위한 플랫폼은 기존의 정형 데이터를 위한 플랫폼 (DBMS 서버, 네트웍 스위치, 가상화 장비(스냅샷, 볼륨 복제), DMBS 용 스토리지)을 통해 구현이 불가능하다. 예를 들어, 중대한 천문, 기상연구 또는 영화, 방송제작에 막대한 자본을 투자하여 오차없이 실시간



<그림 1> Worldwide Enterprise Disk Storage, Capacity Consumption Share: IDC

으로 결과물을 저장하여야 함에도 불구하고 저장 플랫폼의 성능저하로 결과값을 저장하지 못했다면 곧바로 프로젝트 실패 및 예산 낭비로 이어지기 때문이다. 따라서, 비정형 데이터를 구현하기 위한 플랫폼을 구현하기 위해서는 인프라 스트럭쳐, 프로토콜, 스토리지 플랫폼 전체가 데이터의 특성에 맞도록 구축이 되어야 한다.

비정형 데이터 구현 플랫폼을 살펴보기 전에 정형 데이터 플랫폼을 살펴보자.

2. 정형 데이터 처리 플랫폼

정형 데이터를 처리하기 위해서는 방대한 데이터베이스를 빠른 속도로 처리할 수 있는 캐쉬 중심(Cache Centric) 의 플랫폼이 요구된다. 데이터베이스 관리 어플리케이션(DBMS Application)은 거대한 메모리를 구성하고 공유된 메모리 영역에 데이터를 상주시켜 빠르게 사용자의 질의 응답에 대응하도록 되어 있다. 데이터의 사이즈는 주로 수 byte 에서 크게 Kbyte 이다. 금융권의 트랜잭션을 위해서는 상당량(256~512GB)의 캐쉬가 탑재된 시스템을 도입하여 동시에 수 만명의 요청에 응답을 한다. 데이터의 상당량이 캐쉬에 상주해 있으므로 디스크에 직접 액세스를 하는 경우는 아주 간헐적으로 일어나며, 디스크 액세스 과정은 작업 프로세스의 후반에 발생한다.

또한, 데이터의 안정성을 위해서 스냅샷 및 볼륨 복제 기능을 갖추고 있다. 스냅샷은 파일시스템의 디렉토리 및 파일 구조에 대한 목록이다. 주로 NAS구성을 위한 파일 시스템에서 운영이 되며 캐쉬에서 스냅샷을 기록한다. 이 스냅샷을 기록하는 이유는 무중단 서비스를 유지하면서 백업을 수행하기 위해서이다. 시스템의 백업을 위해 금융권 시스템을 중단시킨다면 금융권에 심각한 타격을 입힐 수도 있기 때문에 운영 중 시스템을 백업 받도록 스냅샷이라는 고가용성 솔루션을 제공하고 있다. NAS 구성이 아닌 SAN 구성에서 스토리지 볼륨을 백업 받기 위해 스토리지 레벨에서 볼륨 복제 및 스냅샷을 운영하는 경우도 있다. 스토리지 레벨의 스냅샷 및 볼륨 복제는 스토리지 컨트롤러의 자원을 이용하여 수행이 되는데 서비스 대역폭에 지장을 줄 수 있고, 시스템의 장애가 발생하게 되는 경우에 성능 저하를 가중시키기 때문에 최근에는 가상화 장비를 사용하여 볼륨 복제를 수행하고 데이터의 백업은 2차, 3차의 백업 단계로 나누어 계층적으로 데이터를 관리하는 방법을 채택하고 있다.

데이터베이스의 사이즈는 수Byte 의 텍스트 데이터가 대부분이기 때문에 유통사업의 물류 데이터나 금융권의 데이터베이스일 경우에도 전체 사이즈가 수 십 테라 이상 생성되지는 않는다. 정형 데이터의 백업은 시스템이 운영되지 않는 야간에 주로 백업이 되는데 아주 단시간에 문제없이 처리가 되어야 한다. 즉, 백업 처리 속도가 보장이 되어야 한다. 최근 정형 데이터 처리 플랫폼 환경에서 중요한 구성요소 중 하나로 고속 백업장치가 대두되고 있다.

고속의 백업 장치는 기존의 테이프 저장 장치를 가상화시킨 가상 테이프 라이브러리가 각광을 받고 있다. 다시 말하면, 정형 데이터 플랫폼은 비정형 데이터 플랫폼과 별도로 사용이 되는 것이 아니라 유기적인 관계로 시스템을 구성하고 있다. 그렇다면, 비정형 데이터 플랫폼이 왜 요구되는지 데이터의 특성 및 시스템의 요구사항에 대해서 알아보자.

3. 비정형 데이터 플랫폼의 특성

컴퓨터 사용자라면 날마다 컴퓨터를 켜고 가장 먼저 사용하는 것이 사회 생활의 정보를 포털 사이트에서 확인을 하고 UCC 동영상을 보거나, 주말 여가

시간을 보내며 디지털 카메라로 찍은 사진을 USB 드라이브로 옮겨 저장을 하고, 온라인 프린팅을 요청하는 것이다. 컴퓨터 사용자가 아니더라도 라디오를 듣거나 TV를 보기도 하고, 외출하기 전 기상청에서 발표하는 일기예보에 귀를 기울이고, 특별히 시간을 내서 영화관을 찾아 3D 영화나 장편의 영화를 감상하기도 한다. 특수 목적으로 연구를 하거나, 인터넷 뱅킹을 하지 않는 전부의 시간이 바로 비정형 데이터에 접촉을 하는 셈이다. 최근 공중파에서 방영한 TV 드라마나 스포츠 프로그램에 대해서도 저녁시간에 '다시보기'나 통신사에서 판매하고 있는 IPTV 서비스를 신청하여 다양한 고화질의 콘텐츠를 가정에서 즐길 수 있다. 이러한 점에서 본다면 정형 데이터용 플랫폼에 비해서 비정형 데이터 컨텐츠 서비스 플랫폼은 다양한 범위의 서비스를 지속적으로 해야 한다는 것이 특징이다.

이러한 비정형 데이터를 수용하는 시스템은 우선적으로 QoS를 보장하는 안정적인 시스템으로 구성되어야 한다.

4. QoS (Quality of Service) 보장 시스템

비정형 데이터 스토리지 플랫폼은 특수 목적에 의해 사용되는 시스템이므로 QoS 보장이 절대적인 요건이다. 그 예로 미국의 NASA에서 우주 탐사 및 행성의 특성을 관찰하는 허블 망원경이 보내오는 데이터는 8K x 8K 의 해상도를 가진 초대형 해상도의 이미지 데이터이다. 허블 망원경에서 전송되는 데이터는 지상에 위치한 미국의 항공우주연구소에서 이미지 데이터가 아닌 RAW 데이터의 형태로 수신하며, 렌더링(데이터를 이미지로 변환하는 프로세스) 과정을 통하여 이미지로 추출된다. 이 과정에서 스토리지의 역할이 가장 중요하다. 8K 의 해상도를 가진 데이터를 실시간으로 표출하거나 지정된 시간 내 결과물을 산출하기 위해서는 일관적인 성능(Sustained Throughput)이 보장되어야 하기 때문이다. 만일, 일관적인 성능을 내지 못하는 경우에는 막대한 비용이 투자되는 우주항공 연구 결과가 허사로 돌아가는 경우도 있기 때문이다. 우주 연구에서는 광학 망원경이 아닌 전파 망원경을 사용하는 경우도 있는데 전파 망원경은 데이터의 무결성(Data Integrity)이 100% 보장되어야 한다. 데이터 무결성은 데이터의 수신 당시의 스토리지 저장 속도와 직결되어 있다. 일관적인 데이터 전송속도가 유지 되지 못할 경우에는 재 실험을 해야 하며, 우주 관측 데이터 손실이 불가피 하다.

마찬가지로 디지털 영화제작, 디지털 방송 제작 환경에서도 데이터의 수집단계는 가장 중요한 단계이다. 훌륭한 대본과 감독, 스태프, 유명배우들이 명연기를 펼쳐 하나의 작품을 만들어도 작품을 제대로 저장장치에 담아내지 못한다면 다시 비용을 들여서 제작을 해야 하는 것과도 같은것이다.

위에서 열거한 두 가지의 예를 통하여 보면 QoS의 보장은 최우선적이다. QoS 를 보장하는 미디어 스토리지의 특성은 아래와 같다.

(1) 비정형 데이터를 위한 스토리지 아키텍처
(2) 디스크 성능 기반의 시스템
(3) 읽기/쓰기의 동일한 성능
위 세 가지 특성에 대하여 하나씩 살펴보자.

(1) 비정형 데이터를 위한 스토리지 아키텍처
: 스토리지는 컨트롤러와 디스크 엔클로져의 유기적인 연결로 구성되어 있다. 스토리지 컨트롤러에는 호스트와의 인터페이스를 담당하는 부분, 데이터 IO를 담당하는 부분, 시스템을 모니터링하고 관리하는

부분, 디스크를 컨트롤 하는 부분으로 나누어진다. 정형 데이터를 처리하는 스토리지와 비정형 데이터를 처리하는 스토리지는 컨트롤러가 가장 큰 차이점을 가지고 있다.

정형 데이터 스토리지는 데이터를 생성하고 스토리지로 데이터를 전달하는 주체가 DB 서버인 반면, 비정형 데이터용 스토리지로 데이터를 전달하는 주체는 어플리케이션 서버로 미디어를 생성하여 전달하는 미디어 그 자체인 셈이다. 미디어 데이터는 자연 현상 및 인간의 행위를 광학적으로 포착하여 디지타이징 한 후 데이터를 획득하는 서버로 전달되고, 서버는 단순히 그 데이터를 스토리지로 전달하는 역할을 하게 된다. 미디어 관점에서 본다면 스토리지 컨트롤러가 미디어를 게이트웨이 서버를 통해 데이터 전달을 못하게 될 경우 에러를 발생시키게 된다.

미디어 데이터는 디지타이징 과정에서 양자화(Quantization)과정을 거치게 되는데 양자화 과정에서 생성되는 데이터의 사이즈가 256KB 에서 1MB 이상이다. 이러한 데이터를 처리하기 위해서는 스토리지 컨트롤러의 프로세서가 멀티 코어 프로세서 이어야 하고, 멀티 잡을 처리할 수 있도록 Multi thread를 처리할 수 있는 다 수의 프로세서일수록 유리하다. 또, 데이터의 수신부와 데이터 처리 CPU, 캐쉬 컨트롤러, 디스크 컨트롤 엔진의 순으로 처리되는 일반적인 스토리지 아키텍쳐 보다는 데이터를 병렬화시켜 처리하도록 데이터 수신부, 병렬 처리화 엔진, 디스크 컨트롤 엔진, 순으로 구성되어 있어야 빠른 속도로 미디어 데이터가 가능하다.

좀더 구체적으로 설명하기 위해 데이터 액세스 계층에 대해 설명하고자 한다.

① 물리적 액세스 계층

오늘날 모든 컴퓨터 시스템은 상호 연결된 하드웨어/소프트웨어 계층을 이용하여 궁극적으로 디스크에 저장된 데이터에 접근하고 있다. 그 내부에는 다양한 치환이 존재하지만 소프트웨어 중심으로 일단, I/O계층을 어플리케이션에서 파일시스템이 파일 데이터를 읽고 쓰도록 허용하는 내부 시스템 인터페이스에서 출발해 보기로 한다.

파일시스템은 메타데이터를 유지하고 Block-Level 스토리지 수준에서 물리적으로 상주하게 되는 데이터에 대한 정보 통제(데이터 접근경로 설정, 접근권한 부여 등)임무를 수행한다. 물리적 하드웨어 계층은 계층수준에 이용 가능한 다양한 유형의 인터페이스로 작용한다.

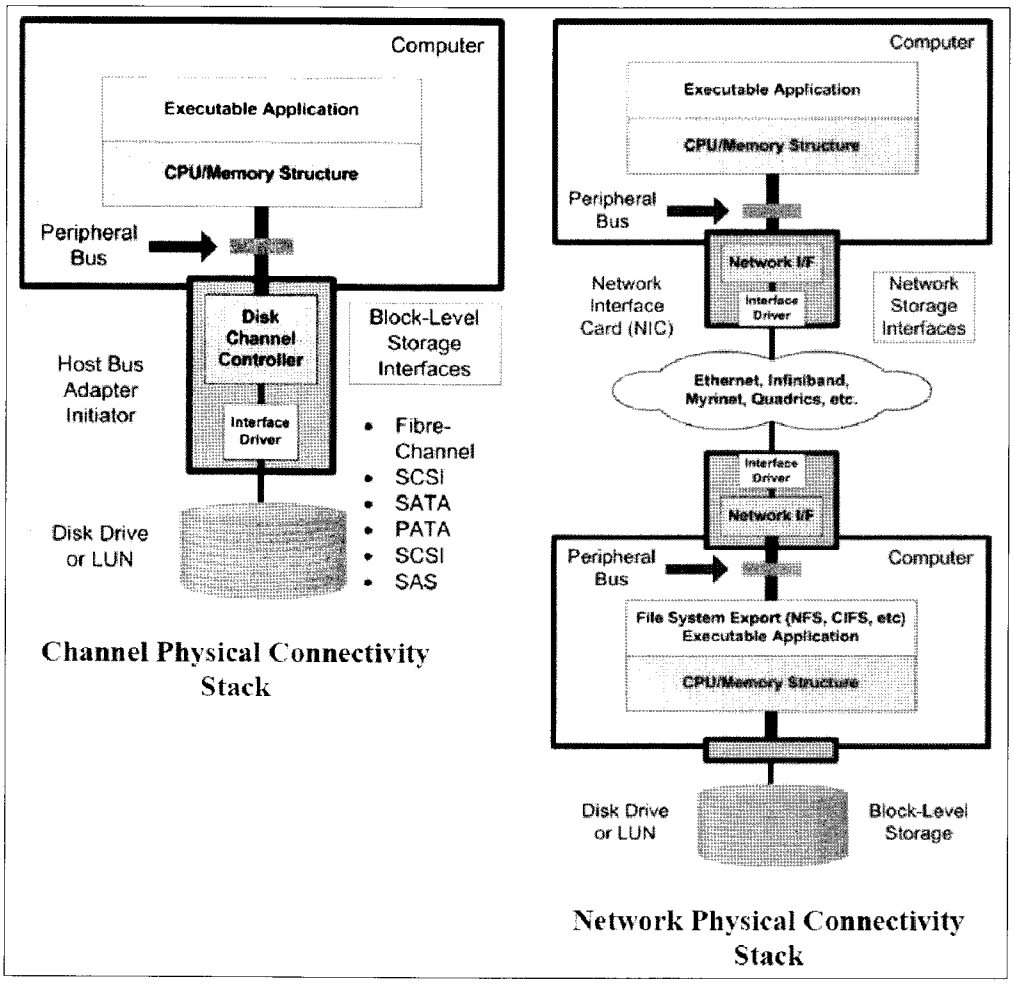
다양한 물리적 하드웨어와 계층 접근의 도구인 소프트웨어를 매우 견고하게 통합해주는 솔루션 개발되는 것이 비정형 데이터 스토리지 시스템 컨트롤러의 핵심이다. 즉, 심플하고 개방된 아키텍처를 사용하여 호스트 CPU사용율의 최소화, 업무효율성 향상, 손쉬운 연결 등을 제공해 주는것이다. Open System 및 개별H/W와 S/W 인터페이스 뿐만 아니라 운영시스템, 그리고 전체 네트워킹과 클러스터 환경에 요구되는 파일시스템 및 Device Driver지원 능력을 높여주는 것이 비정형 데이터 스토리지의 요구 사항이다.

물리적 하드웨어 계층은, 소프트웨어 운용기반에서 데이터와 접속 경로를 제공한다. 하드웨어 차원에서 클라이언트 CPU는 일반적으로 Local Memory와 BUS에 연결되는 데, BUS에는 하나 또는 여러 개의 하드웨어 인터페이스 카드가 다음 둘 중의 하나로 제공된다.

■ Channel : Block-Level디스크 디바이스로의 연결. 또는
■ Network : 파일서버 또는 어플리케이션 디스크 디바이스로의 연결.

채널 직접 연결방식(<그림 2> '좌' 참조)의 경우에는, 클라이언트의 어플리케이션 (소프트웨어 솔루션의 의미로 보면 됨)이 Block-Level 디스크 스토리지 디바이스로 직접적 연결을 지닌 동일한 컴퓨터에 상주한다고 가정한다. 특히 고대역폭 어플리케이션에서, 컴퓨터가 실제 End-User 어플리케이션으로 운영되고 다른 컴퓨터들과 직접 통신하거나 데이터를 보내지 않더라도, 빈번하게 이 어플리케이션에서 수행되는 서비스는 NFS서버 또는 MicroSoft 파일서비스와 같이 네트웍에 연결된 클라이언트에 로컬파일시스템 내보내기를 수행한다.

Block-Level 스토리지의 호환성은 순수한 데이터 저장용 디스크(Raw disk target)와 Initiator (=HBA 카드)로의 연결을 의미한다. Initiator는 스토리지와 직접 연결된 컴퓨터 내부에 있는 데, 전형적으로 Host-Bus Adapter Card로서 마더보드 슬롯상에 탑재되어 에러 교정, 흐름제어, CPU에 부하 발생등의 역할을 한다.



<그림 2> 데이터 액세스의 물리적 연결 계층

Target(즉, 실제 데이터의 저장위치)은 디스크 드라이브 자체이거나 RAID시스템에 의해 생성되는 Logical Unit(LUN)이 된다.

네트웍 연결(<그림 2> '우' 참조)방식의 경우에는, 클라이언트 아래에 분리된 파일서버는 클라이언트의 요구에 서비스를 하기 위해 네트웍 인터페이스 카드(NIC)와 로컬Block-Level스토리지로의 직접적인 연결을 위한 채널 인터페이스를 둘 다 지니고 있는 고유의 하드웨어 계층을 가지고 있다. 이것은 파일서비스 아키텍처를 위한 표준 Network Attached Storage(NAS)에서 찾아볼 수 있는 전통적인 클라이언트/서버 모델을 형성한다. 일반적인 채널 인터페이스 타입에는 Fibre-Channel, Serial and Parallel ATA(SATA and PATA), parallel SCSI, 그리고 Serial SAS가 있으며 최근에 급부상하고 있는 인터페이스로는 이미 잘 알려진 Infiniband와 iSCSI (SCSI-over-TCP/IP)가 있다. Fibre-Channel은 외장 스토리지 연결의 지배적인 수단이 되어왔고, 일부 신기술 이용자를 중심으로 스토리지 및 외부와의 연결에 Infiniband를 사용하기 시작했다. Fibre-Channel과 Infiniband는 원거리 고성능-네트웍 구축시와 대부분의 경우에 스토리지 하드웨어 내부의 성능보다도 고성능 수준으로 성능을 공유한다.

FC-4,8(=4,8Gb/s FC Channel)와 IB(=20,40Gb/s Infiniband)인터페이스의 고성능 인터페이스 수단들이 성능, 신뢰성, 레이턴시, 거리, 비용 또는 확장성 면에서 유리하며 어떠한 시스템들과도 호환이 용이하게 된다.

네트웍 연결 계층인 TCP/IP나 범용 프로토콜들은 파일 접근이 가능하도록 하기 위해 이용된다. 고성능 컴퓨팅에서는 Infiniband가 보편적이고 개방적인 전송 수단으로 확산됨에 따라 Myrinet, Quadrics, 또는 서버 개발사 전용의 인터페이스들이 클라이언트 노드와 서버노드간 연결시 자주 이용된다. 소규모 시스템 환경이나 대다수 비즈니스 어플리케이션의 경우에는, 범용 NIC카드 또는 합리적인 비용 수준에서 전체 시스템의 성능을 향상시키기 위해 CPU-offloading (CPU의 부하 제거)기능을 보유한 카드를 이용하여 Ethernet이 TCP/IP 전송 수단으로 이용된다.

② 소프트웨어 액세스 계층

하드웨어 계층위로 데이터 접근을 제공하기 위해서 반드시 디스크로의 접근을 관리하는 몇 단계의 소프트웨어 계층이 존재한다. 데이터의 이용자인 사용자 응용프로그램은 파일시스템에 접속한다. 여기서, 파일시스템은 반드시 내부 CPU, 내장 메모리, 인터페이스 카드 장치 드라이버, 그리고 End-User응용프로그램과 프로세스에 의해 접근 가능한 볼륨을 구성하는 LUN들을 파일시스템으로 내보내는 일반적인 RAID시스템내의 내부 데이터 경로를 거쳐야 한다.

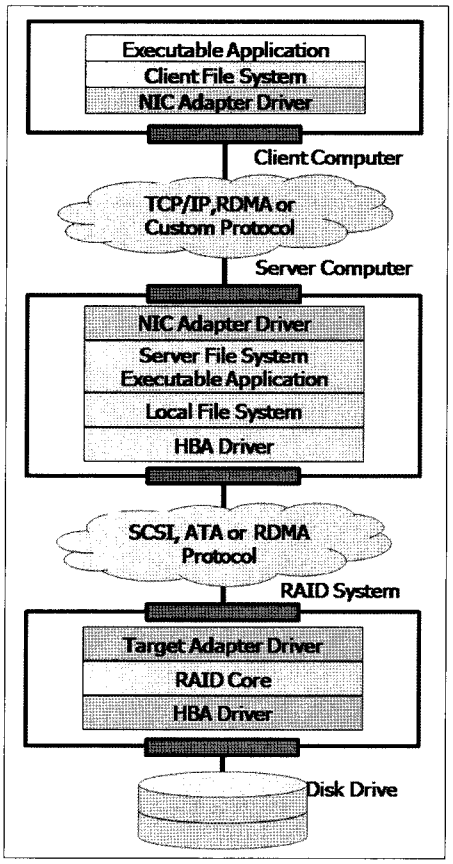
어플리케이션은 싱글 컴퓨터 또는 네트웍내 다수의 컴퓨터가 동시에 데이터 프로세싱을 하는 병렬처리 환경에서 운용되도록 컴퓨터 메모리상에 기록될 수 있는데, 파일 시스템들은 또한 싱글 또는 멀티 컴퓨터에서 컨텐츠 접속을 관리할 수 있도록 개발되어 왔다. 역사적으로, Block-Level 디스크 스토리지는 싱글 컴퓨터에만 접속 가능하였고, 파일시스템은 그 컴퓨터에 상주하게 되며 오로지 해당 컴퓨터로만 운용이 가능하였다. 이 같은 내부 파일시스템은 단순히 자신이 보유하고 있는 메타데이터를 관리하며 Locking 메커니즘을 통하여 싱글 컴퓨터 환경에서의 보안과 신뢰성을 확보해왔다. 시간이 흐르면서, 멀티 컴퓨터 환경에서 동시에 같은 데이터를 공유하려는 요구로 다수의 장비들이, 많게는 수 만대의 컴퓨터가 고속으로 대량의 데이터에 접근을 가능하게

하는 공유 파일시스템의 개발이 이루어졌다. 이러한 대용량 공유 파일 시스템도 비정형 데이터를 처리하는 플랫폼의 중요한 요소로 고려되고 있다.

또한 소프트웨어 계층에서 파일시스템은 인터페이스를 통해 내장 소프트웨어 모듈과 버퍼링 시스템을 지닌 디스크 스토리지와 통신을 한다.

마지막으로 스토리지 디바이스 자체에는, 즉 RAID 디바이스에는 사용자 어플리케이션으로 데이터를 제공하기 위해서는 반드시 경유해야 하는 고유의 하드웨어/소프트웨어 계층을 가지고 있다. 따라서, 비정형 데이터에 유리한 아키텍처는 데이터 계층의 문제점들을 최소화하는 기능을 제공하여야 하고, 내재된 패러럴리즘(병렬처리 기술)의 강력한 통합 기능을 제공하여 과도한 CPU사용 또는 인터페이스 카드 드라이버나 파일시스템과 같은 소프트웨어에 의해 유발되는 레이턴시가 원천적으로 차단시켜야 한다. 이를 통해 미디어 환경에서 요구하는 일관적인 성능 제공을 위한 Multi-Pathing과 Parallel-Access를 제공하여야 하는 것이다. 또한, RDMA방식의 아키텍처를 지원하여 메모리 카피와 드라이버에서 발생하는 레이턴시를 줄여줌으로써 성능과 확장성이 개선되는 효과를 가져와야 한다.

비정형 데이터 프로세싱에 유리한 시스템은 LUN 가상화를 통한 스토리지 가상화 기능으로 WWN매스킹/필터링과 포트 조닝이 간단하고 시스템에 적용이용이하여야 한다. 데이터 병렬 처리 아키텍처를 지닌 시스템은 스토리지 레벨에서 전체 자원을 (100TB이상) 단일 볼륨으로 스트라이핑 할 수 있는 능력을 제공하며, 파일시스템 기반 소프트웨어-Stripe (과도한 스위칭 레이턴시의 유발 및 FC 패브릭 스위치에서 스토리지 포트 병목 현상 유발 원인)없애줄 수 있도록 하드웨어 Stripe 기능을 제공하여야 한다. 이러한 점이 정형 데이터 스토리지 시스템과 비정형 데이터 스토리지의 대표적인 차이점 중 하나이다.



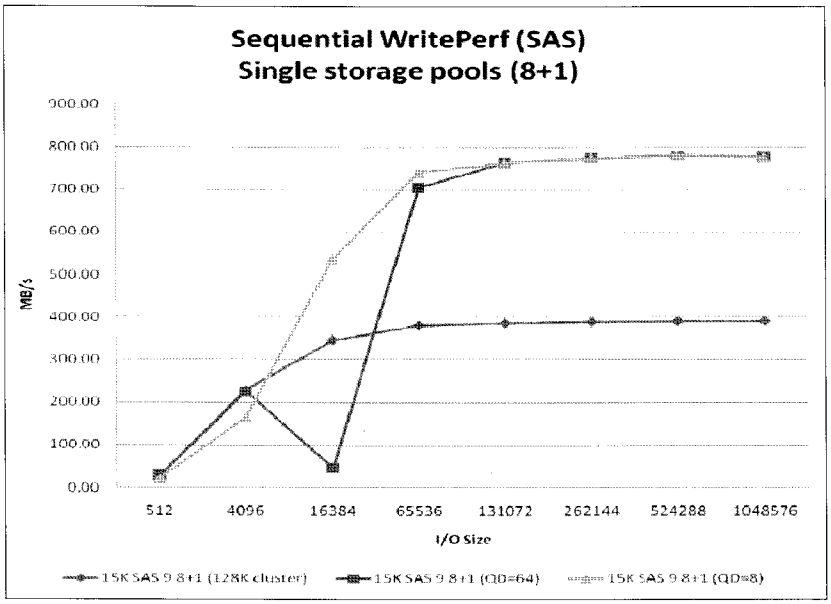
<그림 3> Software Protocol Stack

하드웨어 병렬 처리 구조(Parallel Access)는 일반적으로 공유메모리 구조 및 진보된 캐시 동기화 능력의 제공으로 어떠한 일반 RAID시스템(거의 대부분 데이터베이스와 Back-dffice 환경용의 범용 목적으로 설계된RAID) 보다 월등한 확장성을 제공하며 SAN Fabric 의 레이턴시 없이 호스트와 디스크 컨트롤러간 최소의 경로를 제공함으로써 사용자 환경에 최적 IO를 제공하는 개념이다.

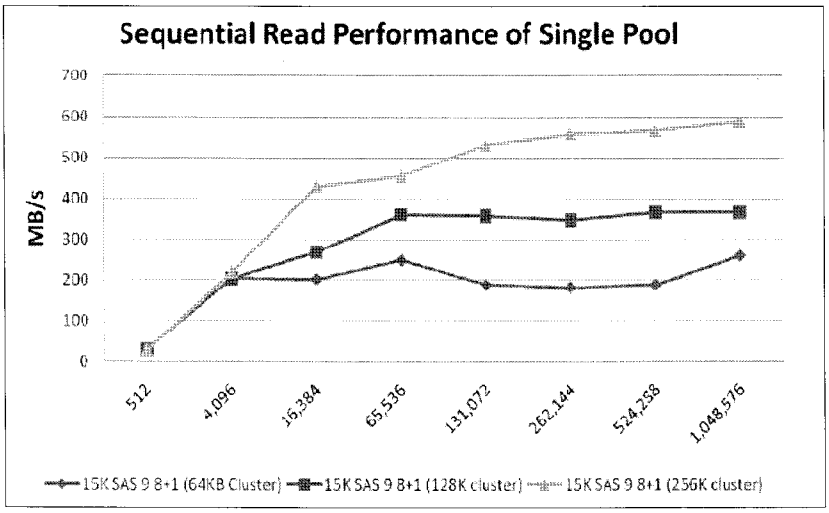
(2) 디스크 성능 기반의 시스템

대용량 캐쉬 기반의 성능을 제공하는 스토리지 시

스템은 데이터 IO 시 항상 데이터의 일부분을 캐쉬에 상주시킨다. 어플리케이션에서 데이터를 메모리에 상주시키는 것이 아니라. 스토리지 시스템의 아키텍쳐 자체가 그렇다. 일정량의 데이터를 읽을 시에는 일시적으로 효과가 좋을 수 있지만 쓰기 시에 대용량의 미디어 파일이 캐쉬에 상주하게 되었다가 디스크로 내려 쓸 경우에는 캐쉬 성능만큼 디스크의 성능이 받쳐주질 못하기 때문에 병목 현상이 발생하게 된다.



<그림 4> SAS 드라이브 쓰기 성능 8+1 RAID 그룹 1 개



<그림 5> SAS 드라이브 읽기 성능 8+1 RAID 그룹 1 개

비정형 데이터를 효과적으로 처리하기 위해서는 호스트와 스토리지 시스템 사이에 최소량의 캐쉬가 존재해야 한다. 일관적인 성능 구현 중 캐쉬에서 디스크로 내려쓰기를 한다면 매 번 디스크에 쓰기를 하는 대기열에 의해 레이턴시가 발생을 하게 되고 0.001%의 오차를 허용하지 않는 비정형 데이터 환경(우주 항공, 미디어 분야)에서 사용할 수 없게 된다.

캐쉬 시스템에서는 데이터의 IO 사이즈의 미리 읽기(Prefetch) 만큼 보유한 후, 바로 디스크로 내려쓰기를 하는 것이 미디어를 위한 최적 성능을 구현하는 방법이다. 또한, 디스크에 데이터를 내려 쓸 경우에는 디스크 그룹의 수만큼 비례하는 성능 수치를 구현할 수 있어야 한다.

각 디스크 그룹(RAID그룹)의 성능이 스토리지 레벨에서 스트라이핑 될 경우에 Block Level 에서의 성능은 최대 6 ~ 10GB/S 에 달한다. 이러한 디스크의 성능을 호스트로 손실 없이 전달하기 위해서는 다음과 같은 방식을 채용하는 것이 가장 효과적이다.

디스크 백엔드 채널 수 만큼 LUN 을 생성하여 모든 LUN 을 스트라이핑 할 경우 최적 IO 가 구현이 되는 것으로 측정된다. IO 사이즈와 성능의 그래프를 통하여 알 수 있는 상황으로 디스크의 수가 일정 수량 이상이고 데이터의 IO 사이즈가 64KB 이상이면 최적 성능을 발휘하는 것이다.

(3) 읽기/쓰기의 동일한 성능

대부분의 외장 스토리지 디바이스(External Storage Device)는 캐쉬 성능으로 인해 읽기 성능이 쓰기성능에 비해 상대적으로 뛰어나다. 슈퍼컴퓨터나 미디어 파일들을 기록하기 위해서는 읽기/쓰기 성능이 동일하여야 한다. CPU 의 자원과 Fibre Channel의 레이턴시(Latency)를 최소화 시킨 시스템에서는 일반적으로 읽기 성능과 쓰기 성능이 동일하게 구현된다. 데이터의 입력속도가 만족되지 못한 시스템으로는 연구환경이나 미디어 작업 환경을 구현할 수 없는 것은 당연한 이치이다.

기상청에서 사용되는 슈퍼컴퓨터 환경에서는 사용자의 요구 환경이 최소 10GB/s 이며, 방송 환경에서는 40대 가량의 작업자가 HD-SDI(1.485Gbps)과 같은 고해상도의 미디어를 실시간으로 저장하고, 편집을 한다. 이러한 환경에서 읽기 성능만 우선적으로 고려가 된다면, 데이터의 원본을 안정적으로 확보하기가 상당히 어렵다. 따라서, 쓰기 성능이 우선적으로 고려대상이 되어야 데이터 원본 확보가 만족되는 것이다.

① 안정성 확보

데이터 스토리지의 첫 번째도 두 번째도, 선결 조건은 '안정성'이다. 성능을 논하기 이전에 안정성이 가장 먼저 논의되어야 한다. 안정성을 확보하기 위한 스토리지의 조건은 RAID LEVEL 및 데이터 페리티 관리이다. RAID LEVEL은 주로 고속 환경에서 사용되는 경우에 3이나 6을 채택한다. RAID 5의 경우에는 회전식으로 모든 디스크에 페리티를 기록하고 모든 데이터의 싱크를 맞추기 위해 디스크의 스핀들에 LOCK을 걸고 회전수를 제어하게 된다. 따라서, 데이터 기록 시에는 고성능 구현이 불리하다. 이러한 제약 사항으로 고속 연산 및 미디어에 RAID 5를 적용하기란 어렵다. 각 업계마다 RAID 구현 방법이 조금씩 차이가 있게 마련인데, 일반적으로 RAID 3(Data Disk Group + 1 parity)를 사용하고, 안정성을 좀 더 높이기 위해서 RAID 6(Data Disk Group + 2 parity)를 적용한다. RAID 6를 구현하게 되는 경우 2개의 페리티를 구현하게 되면 RAID 3 구현 환경보다 스토리지의 컨트롤러가 많은 연산을 해야 하기 때문에

속도가 느리게 된다. 이러한 기존 RAID 6 구현 방법으로 인하여 각 제조 업체에서 자기만의 RAID 6을 개발하기 시작했고, 최근 들어 별도의 패리티 채널을 통하여 Parity Data를 기록하는 방법인 RAID 6 방식을 구현한 업체도 등장하였다. 독립적인 패리티를 구현한 RAID 6인 경우에는 최대 10GB/s의 성능이 단일 시스템에서 구현이 된다.

이러한 RAID 6의 등장으로 자원의 20% 가 동시에 장애를 발생시켜도 고속 환경에서 성능 저하 없이 슈퍼컴퓨팅 및 미디어 기록을 가능하게 한다.

일반적인 RAID 구현 방식은 단 몇 개의 디스크 장애에 의해 전체 성능이 저하 되지만, 비정형 데이터 스토리지 시스템은 독립적인 패리티 엔진과 채널로 인해 장애 유발에 대해 보완책을 충분히 가지고 있는 셈이다.

데이터 보호는 쓰기 시 패리티에 의해 보호가 되고, 읽기 시에 2차적으로 보완이 된다. 디스크의 장애에 의해 발생되는 데이터 손실(Data Corruption)은 NAS 서비스나 호스트들이 데이터를 요구할 때 체크되지 못한다. NAS 파일 시스템 및 SAN 공유 파일 시스템은 전송에 관련된 데이터 패킷이나 데이터 블록을 관리하지만, 손실된 데이터는 감지하지 못한다. 이러한 데이터 오류를 정정하기 위해서는 스토리지 컨트롤러가 처리를 해 주어야 한다. 데이터 오류를 감지하기 위해서는 읽기 시에도 패리티 검증을 하여 오류를 자동으로 정정을 해 주어야 호스트에게 전달을 할 때 데이터 무결성을 확보하게 된다.

Ⅳ. 결 론

많은 스토리지 제조사에서 발표하고 있는 비정형 데이터용 스토리지들이 있다. 그 구현 방법들도 상당한 차이가 있으나, 일맥 상통하고 있는 사용자의 요구 사항은 "안정적이고, 빠른 스토리지" 이다. 안정성 및 성능을 확보한 스토리지는 아키텍쳐가 단순하여야 하며, 장애에도 성능 저하가 없어야 하고, 데이터의 무결성을 100% 제공할 수 있어야 한다. 스토리지 선별 과정에서 기존 정형 데이터 스토리지 제조사들이 주장하고 있는 스토리지 사양들에 사용자들은 이끌려 가고 경험을 해 본 후에 다시금 비정형 데이터들에 대한 욕구를 느낀다. 시스템에 대한 중복 투자가 여러 기업에서 발생이 되고 예산이 낭비가 되고 있다. 비정형 데이터 환경에 맞는 스토리지 선정은 연구에 필요한 슈퍼컴퓨팅 및 미디어 환경 구현을 위한 필수요소로 투자 보호의 지름길이자 연구 결과 및 작업 프로세스를 안정적으로 구현할 수 있는 유일한 방법이다.

필 자 소 개

김 정 림

- *홍익대학교 기계공학 전공*
- *현재 : 한국DDN 차장*