**클라우드 #5주차 과제**

21660019 3A김세진

**1. 트랜잭션**

**2. SATA vs PATA ,(SAS, SCSI방식)**

**3. DW개념, 환경 세팅**

**4. mount (가상머신에서 유형)**

**5. 네트워크 스토리지(CL측면)**

**6. 백업(PC, 스마트폰)**

**7. 랜섬웨어**

**8. 스냅샷**

**9. VTL,PTL**

**10. 풀 백업, 증분백업**

**11. L1~L7 OSI7계층**

**12. 베어메탈**

**13. SLA**

**1. 트랜잭션**

1-(1) 트랜잭션의 개념

트랜잭션(Transaction)은 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.

1-(2) 내가 생각하는 트랜잭션의 개념

데이터 베이스에서 하나의 값을 바꿈으로써 그것과 관련된 값이 같이 바뀔 수 있는데, 바뀌는 도중에 그 값을 참조하여 데이터를 만들면, 계산에 실제와는 다른 값이 들어갈 수도 있다. 따라서 바뀌는 모든 동작을 하나의 동작이 수행되는 것처럼 묶어서 실행하는데 이것을 트랜잭션이라고 하는 것 같다.  
 또한 어떠한 오류에 의해 연산이 도중에 멈추면 관련된 모든 값을 원상태로 되돌리는 기능도 트랜잭션이 처리한다고 보면 될 것이다.

1-(3) 효율적으로 이용, 효율적 세팅, 활용 방안

트랜잭션은 여러 연산이 복잡하게 수행될 때, 그 과정에서 여러 문제가 터지는 것을 예방하거나 막는데 이용이 될 것이다.   
  
첫 번째로, 계좌를 생각해보자, 10,000을 보내는데, 내 계좌에서는 빠져나갔지만, 받는 계좌에서 돈이 들어오지 않는다면? 10,000원이라는 돈은 갑자기 사라진 것으로 처리된다. 이는 심각한 오류이므로 우리는 트랜잭션으로 내 계좌 -10000의 연산과 받는 계좌 +10000연산을 하나로 묶는다. 이것을 하는 방식은 첫 번째 처리를 임시로 하고, 두 번째 처리를 마무리 지으면, 그제서야 DB에 실제 값이 바뀌는 식으로 처리된다.

그렇다면, 이러한 연산을 하고 있는데 그 값을 또 쓴다면??

위의 예시에서 트랜잭션을 처리하다가 그 관련된 값을 다른 트랜잭션에서 쓴다고 생각을 해보자, 내 계좌에서 10000원을 빼고 다른 계좌에 더할 때, 다른 트랜잭션에서 내 계좌의 잔여 금액을 요구하면 10000원이 빠진 금액을 줄 것이고, 첫 번째 트랜잭션이 롤백되면, 모든 연산에 이상한 데이터가 들어갈 것이다.

이러한 상황에 맞게 트랜잭션을 미리 세팅해두어야 하는데, 한 트랜잭션이 값을 사용하고 있을 때는 다른 트랜잭션이 이 값을 사용하지 못하게 lock을 시킨다. 이로 인해 내 계좌가 한 트랜잭션이 차지하고 있으면 다음 트랜잭션은 연산이 끝날 때까지 내 계좌를 이용 할 수 없는 것이다.

이러한 lock에는 치명적인 약점이 있는데 여러 트랜잭션이 서로 lock을 시키는 deadlock을 유발할 수도 있다는 것인데, 이러한 결과를 막거나 해결하기 위해서 초기 세팅으로 deadlock 발생시 하나의 트랜잭션을 롤백하고 이후에 처리하는 방식으로 세팅할 수 있다.

**2. SATA vs PATA ,(SAS, SCSI방식)**

2-(1) SATA, PATA의 개념

ATA : (Advanced Technology Attachment)의 줄임말이며, 직역하면 고급 기술 결합, 컴퓨터에 하드디스크, CD-ROM드라이브와 같은 기억장치를 연결하는 표준 인터페이스이다.

PATA : 병렬로 연결하는 방식의 ATA이다.

SATA : 직렬로 연결하는 방식의 ATA이다.

SCSI : (small computer system interface) 주변 장치를 컴퓨터에 연결하기 위해 사용되는 표준 병렬 인터페이스이다.

SAS : (Serial Attached SCSI) 직렬연결 SCSI로 병렬연결의 특성상 데이터 전송의 한계로 고성능의 인터페이스 필요에 따라 개발됨  
  
2-(2) 각각의 장단점

PATA vs SATA

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 속도 | 케이블 크기 | 장비 호환성 |
| PATA | 최고 133MB/s | 40~80핀 | 최신 장비 호환 X |
| SATA | 최고 600MB/s | 7핀 | 최신 장비 호환 O |

PATA와 SATA 차이를 요약 해 본다면, PATA의 병렬 데이터전송으로 인해 속도의 한계가 존재하여, 발전한 형식이 SATA의 직렬 데이터 전송이다. 따라서 SATA의 데이터 전송의 속도가 훨씬 빠르며, 케이블의 크기 차이로 인해 탈부착의 편리함과 전력의 소모 감소, 컴퓨터 내부 공기 흐름 개선에 도움이 된다. 또한 최신 블루레이, 하드디스크 등에 호환성이 좋다.

2-(3) 어떻게 쓸지

이 4가지는 컴퓨터의 성능을 결정하는 인터페이스 규격을 나타내는 것 같은데, 첫번째로 병렬 방식은 느리기 때문에 요즈음에는 거의 쓰이지 않을 것 같다. 찾아 본 바에 의하면, PATA, SCSI방식의 인터페이스를 가지는 것은 최신 장비에는 찾아보기 힘들다고 나온다, 따라서 내가 컴퓨터를 새로 맞춘다면 SATA, SAS방식의 인터페이스를 지원하는 장비를 살 것 같다.

그러면 SATA와 SAS가 남는데, 이 둘은 서로 호환이 가능하다. 서로의 차이는 연결할 수 있는 장치의 수, 확장성, 속도, 발열, 가격 등이 있다.   
 연결 가능 장치 수는 SAS가 훨씬 많고, 확장성도 SAS가 더 좋고, 확장 시 신뢰성도 뛰어나다. 속도의 면에서도 SAS가 SATA에 비해 2배 정도 빠른 것 같다. 이러한 성능 때문에 발열은 더 심하며, 가격도 2배정도 비싸다.   
 정리하자면, 성능의 면에서 확연한 차이를 보이지만, 가격이 만만치 않고, 소음이나 발열이 심하다는 것을 따져보면, 가정에서는 SATA규격을 사용하는 것이 가장 합리적인 것 같으며, 서버같은 환경에서 안정성, 기능을 따진다면 당연히 SAS규격을 사용하여 많은 처리, 확장성을 노리는 것이 좋아 보인다.

**3. DW개념, 환경 세팅**  
 3-(1) DW의 개념

DW : (Data Warehouse)란 사용자의 의사 결정에 도움을 주기 위하여, 기간시스템의 데이터베이스에 축적된 데이터를 공통의 형식으로 변환해서 관리하는 데이터베이스를 말한다.  
  
 3-(2) 내가 생각하는 DW의 개념  
 DB에 값을 넣고, 그 값들을 이용하여 필요한 정보로 만들어 내는 것은 가능하지만, 그렇게 하면 DB의 처리속도가 느려지고 데이터의 안정성에 문제가 생길 수도 있을 것이다. 따라서 데이터를 계속 넣어주는 DB를 존재시키고, 이 DB에서 데이터를 모두 DW로 옮겨서 필요한 정보를 만드는 것을 할 것이다. 짧게 말하면, 데이터를 저장하는 DB와 저장한 데이터를 사용하는 DB의 차이점이라고 보면 될 것 같다.   
  
 3-(3) 환경 세팅

정확하게 이해했는지 모르겠지만, DW에 대해서 더 정확한 설명을 해보자면, 기존의 DB의 값을 가지고 특정 원하는 정보를 얻고자 특별한 데이터베이스를 구축했다고 보면 되는데,

방식은 다음과 같다. 기존 DB를 복사하여 그 데이터를 ODS(Operational Data Store)를 통해 DW로 전해지고, ELT등의 도구로 데이터를 추출, 변형하여 DW에 저장된다. 그 과정에서 데이터는 활용되기 위해 여러 조건에 의해 나뉘는데 이렇게 나뉜 데이터는 각각의 data mart라는 형식으로 스토리지에 저장되고, 이로 인해 데이터를 분석하는데 더 유용하게 사용되어 진다.

**4. mount (가상머신에서 유형)**  
 4-(1) mount의 개념  
 mount : 마운트는 디스크와 같은 물리적인 장치를 특정위치 즉 디렉터리에 연결시키는 것이다.  
  
 4-(2) 내가 생각하는 마운트의 개념  
 쉽게 말하면, USB를 연결했을 때, E:드라이브 이런 식으로 뜨게 하는 작업을 마운트라고 하는 것 같다. Window 운영체제에서는 운영체제가 자동으로 해주기 때문에 사용자가 처리를 하지 않아도 되지만 Linux체제에서는 명령어를 통하여 처리를 해주어야 된다고 한다.   
 그렇다면 가상머신에서 마운트는 무엇일까?

우리는 하드디스크를 추가적으로 연결할 때, 무엇 때문에 하는 것이냐고 묻는다면, 당연히 추가적인 용량이 필요해서라고 대답할 것이다. 따라서 가상머신에서 마운트를 하는 것은 추가적인 용량을 늘리는 것을 의미할 것이고, 이러한 마운트는 실제 물리적인 디스크를 추가하는 것이 아닌, 가상의 자원을 더 붙인다는 개념일 것이다.   
  
4-(3) 현업에서는 어떻게 쓰일지  
 아무래도 서버에서 리눅스 운영체제를 사용하고 있고, 서버의 스토리지 용량을 증가시키려고 가상, 물리적인 스토리지를 추가했을 때, 리눅스 운영체제가 해당 스토리지를 인식하기 위해서 쓰일 것이다.

**5. 네트워크 스토리지(CL측면)**

5-(1) DAS, NAS, SAN의 개념(CL측면)

NAS(Network Attached Storage) : 네트워크 결합 스토리지로, LAN을 통해 Storage와 Server를 접속하는 방식이다.  
DAS(Direct Attached Storage) : 서버와 외장형 저장장치를 전용 케이블로 직접 접속하는 방법이다  
SAN(Storage Area Network) : 광 저장장치영역네트워크라고 부르며, 서버가 fiver channel switch를 통해 storage를 연결하는 방법이다.  
  
   
5-(2) 각각의 장단점   
 DAS 장점 : 설치가 쉬우며, 비용이 저렴하다. 이미 규격이 표준이 되있기에 성능이 최적화되어 있다.   
 DAS 단점 : 접속 확장이 가능한 SCSI 버스의 수나 처리가능 데이터 용량의 확장성이 제한된다.  
 시스템간 파일 공유가 불가능하여 같은 데이터를 여러 시스템이 가져야 한다.  
 대규모의 데이터, 서버를 관리 시 비효율적인 관리 비용이 늘어난다.

NAS 장점 : 설치와 관리가 용이하다. 총 소유비용의 관점에서 저렴하다.

이기종 시스템간의 파일공유와 파일전송이 가능하다.

대용량을 수용할 만큼 확장이 가능하고, 사용중에 다운할 필요없이 확장이 용이

스냅샷 데이터 보호 솔루션으로 강력한 데이터 보호/ 백업이 가능하다.

NAS 단점 : 랜 대역폭의 한계가 있다. -> 랜은 작은 사이즈의 파일을 신속하게 보내는 방식에 유리하지만, 사이즈가 큰 데이터 전송에 상대적으로 불리하다.  
DAS, SAN에 비해 애플리케이션 서버~ 전용파일 서버, 전용파일 서버~ 스토리지 사이의 접속단계가 늘어서 Latency Time이 더 걸린다.

SAN 장점 : 서버에 직접 연결할 수 있는 스토리지 용량을 서버와 관계없이 대규모 확장이 가능하며, 서버의 다운없이 확장시킬 수 있다.  
 기가비트 이더넷의 전송량보다 빠른 이론상 초당 100MB의 전송률을 가진다.

SAN 단점 : 가격의 면에서 부담이 크다. -> 단순한 스토리지 디스크뿐만 아니라 네트워크 인프라를 고가의 파이버채널 스위치 장비를 사용해야 하는 측면에서 부담이 간다.

5-(3) 나라면 어떤 상황에서 쓸지

장단점을 고려해 볼 때, DAS는 개인 서버나 클라이언트에서 스토리지 용량을 늘릴 때, 사용하는 것이 바람직하다고 본다. 왜냐하면, 단점을 보면 대규모의 데이터를 관리하려면, 많은 물리적인 스토리지를 계속 탈부착해야 되기 때문에 소규모의 스토리지 확장에 용이하다고 생각한다.   
 NAS는 주로 내부 서버망에 용이 할 것 같다. NAS는 서버의 파일을 나누는 데 특화 되어 있다고 생각한다. 즉 중앙의 서버의 스토리지에 필요한 파일, 데이터를 보관하고, 그 서버로부터 파일을 받아오는 식으로 동작하여, 업무의 편리성, 속도를 보장하고 데이터 보호 솔루션으로 데이터를 보호. 백업에 용이하기 때문이다.

SAN은 스토리지 사이의 네트워크를 통해 연결했기 때문에, 하나의 서버가 여러 스토리지를 연결하는데 가장 최적화된 모델이라고 생각한다. 물리적으로 연결하여 탈부착하는 방식이 아니라, 네트워크를 통하여 연결되있기 때문에 확장성도 유연하다. 따라서 SAN은 클라우드 서버의 가상화라는 곳에서 많은 도움을 주는 스토리지 방법인 것 같다.

**6. 백업(PC, 스마트폰)**  
  
6-(1). 백업의 개념  
 백업 : 임시 보관을 일컫는 말로, 데이터를 미리 임시로 복제하여, 문제가 일어나도 데이터를 복구 할 수 있도록 준비해 두는 것을 말한다.   
  
6-(2) 백업의 방안  
 컴퓨터에서 백업 방안 : 요즈음에는 매주 혹은 앱, 드라이버를 설치하는 경우에 PC에서 자동으로 복원 지점을 설정하는데, 시스템에 문제가 발생시 운영체제를 통해서 복원을 하는 방법이 있다. 따로 전체를 백업하는 방식이 부담이 간다면, 사용자가 주기적으로 필요한 폴더나 파일등을 다른 기억장치에 보관하는 방식으로 백업을 해두는 방법도 있다.   
   
 휴대폰에서 백업 방안 : 휴대폰은 여기저기 들고 다니기에 분실이 잦고, 물리적으로 손실이 일어날 가능성이 많은데, 기존에는 미리미리 사용자가 일일이 USB나 SD카드에 필요한 데이터를 저장해 두는 방식을 많이 썼다. 하지만 요즈음 휴대폰을 잃어버리거나 손상이 되도 바로 바로 이전 휴대폰의 데이터를 받아오는데, 이는 SAMSUNG어플, 카카오톡, 구글, 통신사 어플등에서 사용자의 데이터를 받아올 수 있기 때문이다.   
 또한 요즈음엔 네이버 클라우드 등에 자신이 찍은 사진이나 동영상, 파일등이 자동으로 업로드 되기 때문에 분실을 해도 클라우드에 보관되는 경우가 있어서 편리하다.   
  
6-(3) 백업의 중요성, 필요성  
 9.11 테러를 예로 들어보겠다. 9.11테러 당시에 350여 기업의 전산시스템도 건물붕괴와 함께 파괴되었다. 그 당시 데이터복구를 할 수 있던 150여개의 기업을 제외한 나머지 기업은 도산했다고 전해진다. 이렇게 기업은 생각치도 못한 방법으로 기업의 전산시스템, 데이터, 기밀 등을 날려버릴 수 있고, 그에 대한 결과는 기업의 몰락으로 이어 질 수도 있기 때문에 데이터 백업에 시간과 돈을 들여서 보험을 들여놓는 것이 필수적이고, 중요하다고 생각한다.   
  
 **7. 랜선웨어**

7-(1) 랜선웨어의 개념  
랜섬웨어 : 랜섬웨어(ransomware)는 컴퓨터 시스템을 감염시켜 접근을 제한하고 일종의 몸값을 요구하는 악성 소프트웨어의 한 종류이다. 랜섬웨어는 몸값을 뜻하는 Ransom과 Software(소프트웨어)가 더하여진 합성어이다.

7-(2) 내가 생각하는 랜선웨어의 개념   
랜섬웨어는 해커가 어떠한 프로그램에 악성 바이러스를 심어 다운받은 컴퓨터의 일부 기능을 마비시키는 방법 같다. 이런 랜섬웨어를 심는 방법은 여러가지인데, 메일이나 특정 사이트를 클릭함으로써 침투가 될 수 있고, 특정 파일을 다운받으면서 침투가 될 수 있다. 침투가 되면 컴퓨터의 데이터가 다 날라가도 된다고 봐도 되기 때문에 중요한 데이터가 있는 컴퓨터에 심으면, 해커가 돈을 요구하는 방식으로 진행이 되는 것 같다.

7-(3) 현업에서 어떻게 쓰일지  
랜섬웨어의 사례를 찾아보다가 최근의 예시를 보게 되었는데, 대학교나 여러 강의들이 사이버강의를 통해 진행됨으로써, 특정 소프트웨어가 필요하여 다운받아야 되는 일이 발생하는데, 자신의 랜섬웨어를 그런 특정 소프트웨어처럼 둔갑시켜서 다운받게 하는 것이다. 우리 연암공대의 경우를 보면, webex프로그램을 깔았는데 컴퓨터의 모든 파일들이 날라가는 것을 상상하면 된다. 이러한 랜섬웨어를 막는 방법으로는 이상한 메일, 사이트를 들어가는 것에 거리를 두는 방식과 랜섬웨어가 깔려도 다시 파일을 복구할 수 있게 백업을 시켜두는 방식을 들 수 있다.

**8. 스냅샷**

8-(1) 스냅샷의 개념  
 랜섬웨어가 이슈화 된 이후, NAS서버 제조사들은 랜섬웨어를 예방하기 위해서 데이터 백업 솔루션으로 스냅샷 기능을 만들었다.  
  
 8-(2) 내가 생각하는 스냅샷의 개념  
 약간 백업과 스냅샷의 개념이 비슷하다고 생각하는데, 약간의 차이점이 있다면, 속도나 저장위치가 아닐까 생각해본다. 백업은 어떤 물리적인 곳에 같은 데이터를 복사해두는 것이라고 생각한다면, 스냅샷은 기존의 디스크의 일부 공간을 복구용 저장소로 생각하는 것 같다. 이로 인해, 스냅샷은 백업보다 복사 및 복구의 속도가 빠른 것 같다. 단점으로는 해당 디스크가 물리적으로 손상되면 스냅샷은 데이터를 불러올 수가 없다는 점이다.   
  
 8-(3) 현업에서 어떻게 쓰일지

현업에서는 속도의 면에서 스냅샷이 빠르고 안정성이나 정확도의 측면에서 백업이 좋기 때문에, 기업에서는 랜섬웨어를 방지하기 위한 방안으로 하루에 몇번씩 짧게 짧게 스냅샷을 할 것 같고, 물리적인 측면의 손상이나 다른 바이러스 같은 부분을 방지하기 위해서 일, 주 단위의 대량 백업을 하는 방식으로 많이 쓰일 것 같다.

**9. VTL, PTL**  
 9-(1) VTL, PTL개념  
 VTL : (virtual tape library) 가상 테이프 라이브러리란 데이터의 백업 및 복구를 위해 가상화 기술을 이용함으로써 디스크를 마치 테이프처럼 인식하여 데이터를 저장하는 시스템이다.

PTL : (physical tape library) 물리 테이프 라이브러리란 데이터의 백업 및 복구를 위해 물리적인 형식의 테이프를 이용하여 데이터를 저장하는 방식이다.  
  
 9-(2)내가 생각하는 VTL, PTL의 개념  
 예전에 우리가 첩보영화나 드라마를 보면 서버에서 정보를 빼내는 장면을 볼 수 있는데, 그 때 생각나는 것은 어떤 데이터가 들어있는 스토리지를 빼서 넘기는 것이다. 이 때 이 스토리지가 테이프의 형태를 띄었는데, 이렇게 테이프를 이용해 저장해놓는 방식을 PTL이라고 생각하면 될 것이다. 하지만 최근의 트랜드를 보면 데이터의 양이 엄청 많기 때문에 이러한 테이프의 형태로 데이터를 저장하는 방식에는 많은 물리적인 요소가 필요하게 된다. 따라서 우리는 SATA방식의 디스크를 마치 테이프인듯이 사용할 수 있는 방법을 찾게 되었는데 이것을 VTL이라고 한다. 이 VTL방식은 데이터를 저장하는 곳이 물리적인 디스크이지만 컴퓨터는 이 디스크를 테이프라고 인식하게 어떠한 엔진을 단다고 생각하면 된다.   
  
 9-(3) 현업에서 어떻게 쓰일지

현업에서 PTL방식은 기존의 백업시스템이라고 보면, 소규모의 데이터를 다루고, 백업을 해야하는 경우에 단순히 테이프 형식의 스토리지를 이용하는 것이 좋아 보인다. 하지만 최근에는 수 많은 데이터가 오가고, 그것을 처리하면 백업해야 되기 때문에 테이프 방식으로 백업을 한다는 것은 수많은 테이프를 물리적으로 보관해야 한다는 것이고, 장애복구 시 일일이 해당하는 테이프를 가져와 복구를 해야 되기에 시간의 소모가 크다. 회사는 이러한 PTL방식을 사용하는 것이 꺼려질 것이다. 그에 비해 VTL방식은 디스크 방식이므로 테이프보다 더 많은 데이터 수용과 물리적으로 작다는 점에서 보관에 용이하다. 따라서 VTL방식은 많은 데이터를 저장하고, 빠른 복구를 지원할 수 있는 곳에 많이 쓰일 것이다. 내 생각에 요즈음의 물리적인 백업은 모두 VTL방식으로 쓰일 것 같다.

**10. 풀 백업, 증분 백업**  
 10-(1) 풀 백업, 증분 백업의 개념  
 풀 백업 : 전체 백업은 데이터의 변경 유무에 관여하지 않고 전체 데이터의 복사본을 만드는 백업 방식이다.  
 증분 백업 : 증분 백업은 정해진 시간을 기준으로 그 이후에 변경된 파일만을 백업하는 방식이다.  
  
 10-(2)내가 생각하는 풀 백업, 증분 백업의 개념  
 먼저 풀 백업의 경우, 모든 데이터를 복사한다는 개념으로, 한번 백업을 시켜 놓는데 많은 시간이 드는 백업이고, 증분 백업은 풀 백업 이후에 바뀐 데이터만을 백업 시켜놓는 것을 의미한다.

두 방법에는 각각 장단점이 있는데, 차이점은 백업과 복구의 속도차이에 있다. 풀 백업은 모든 데이터를 백업하는 것이니 당연히 백업을 하는데 시간이 오래 걸리지만 복구를 하는데에 모든 데이터를 그대로 옮기면 되기에 복구 속도는 빠르다. 반대로 증분백업을 한 경우에 그때 그때 백업해 놓는 시간은 빠르지만, 복구를 하려면 풀 백업에 여러 증분 백업 데이터를 순차적으로 복구 해야하기 때문에 시간이 오래 걸린다.   
  
 10-(3) 현업에서 어떻게 쓰일지

현업에서는 풀 백업과 증분 백업을 같이 사용하는 방법이 많이 쓰일 것 같다. 일단 백업은 거의 하루에 한번은 해야 한다고 생각하기에, 예시를 1주일동안 백업울 한다는 것으로 두었을 때,

일요일을 풀 백업하는 날로 두고 이전의 모든 데이터를 저장해 놓는 것이다. 이후에 월요일~ 토요일은 해당 일자에 바뀐 데이터를 백업 해 두고, 다시 일요일이 오면 전체 데이터를 저장하면, 월~토요일의 데이터는 초기화하고 다시 같은 패턴으로 백업하는 것이다.   
 이렇게 함으로써 1주일 중 6일동안은 백업을 빠르게 하고 일요일에는 백업이 느리지만, 모든 데이터를 백업시켰기 때문에 장애를 복구하는데 시간을 최소화 할 수 있기 때문이다.

**11. L1~ L7 Layer계층**  
 11-(1) OSI7 계층 개념  
 국제 표준화 기구에 의해 통신이 일어나는 과정을 각 단계별로 파악하기 위해 7단계로 구분해 놓은 컴퓨터 통신 구조의 모델이다.

11-(2)각각의 계층별 설명

1계층 물리 계층 : 물리매체를 통해 bit흐름을 전송, 물리적 장치와 인터페이스가 전송을 위해 필요한 기능과 처리절차 규정

2계층 데이터 링크 계층 : 노드와 노드 사이의 데이터 전달, 물리층에 신뢰성을 더하기 위한 흐름제어 및 오류제어 기능, LLC, MAC 두 개의 서브레이어로 구성

3 계층 네트워크 계층 : 송신 측에서 최종 목적지까지 패킷을 전달, 패킷의 결오를 배정하는 라우팅 기능을 수행, 데이터 링크의 물리 주소는 패킷이 시스템을 이동할 때마다 변경되지만, 네트워크 주소는 목적지 까지 변하지 않음.

4 계층 전송 계층 : 네트워크가 종단까지 전달한 패킷을 전송층은 종단 내에서 최종 수신 프로세스로의 전달(포트주소)를 담당한다. 분할/재조립, 연결/흐름제어, 오류제어 기능

5계층 세션 계층 : 통신하는 프로세스 사이의 대화제어 및 동기화를 담당

6 계층 표현 계층 : 데이터의 변환, 압축, 암호화를 담당

7계층 응용 계층 : 사용자에게 서비스 제공 역할, SMTP, FTP, HTTP 등 사용자가 원하는 최종 목표에 해당.  
  
 11-(3) 계층별 각각의 사례

1 물리 계층 : 시스템의 물리적인 것으로 케이블의 종류, 핀 배치, 전압이 맞는지, 옳게 연결됬는지, 전원 플러그가 빠지지 않았는지를 살펴 본다.

장비로는 허브, 리피터가 있다.

2 데이터 링크 계층 : 예를 들어 컴퓨터, 교환기, 라우트가 직렬로 연결되어 있다면, 이 경우에는 노드가 3개, 링크가 2개인 것이다. 이 계층에서는 물리의 bit와 네트워크의 패킷의 중간단계에서 데이터를 header와 trailer로 감싸는데, header에는 시작을 알리는 플래그, 목적지의 주소, trailer에는 에러 검출을 위한 체크썸, 끝을 알리는 플래그로 구성된다.

즉 물리, 네트워크 사이에서 오류를 검출하거나 목적지를 설정하는 흐름을 수행한다.

브릿지, 스위치 같은 장비들이 해당된다.

3 네트워크 계층 : 네트워크 계층에서 가장 중요한 기능은 라우팅 기능인데 이는 패킷을 어떻게 해당하는 목적지에 빨리 보낼지를 생각해 경로를 설정하는 기능이다. 대표적인 프로토콜은 흔히 들어본 IP를 들 수 있고, 장비로는 라우터를 들 수 있다.

4 전송 계층 : 패킷이 유효한지 확인하여 실패한 패킷은 다시보내는 등 신뢰성 있는 통신을 보장한다. 프로토콜로는 TCP가 있으며, 장비는 게이트웨이를 예로 들 수 있다.

5 세션 계층 : 전송받은 패킷을 어떤 프로세스로 전달할지에 대한 계층이며, 사용자간의 포트연결(세션)이 유효한지 확인, 설정하는 것을 예시로 들 수 있다.

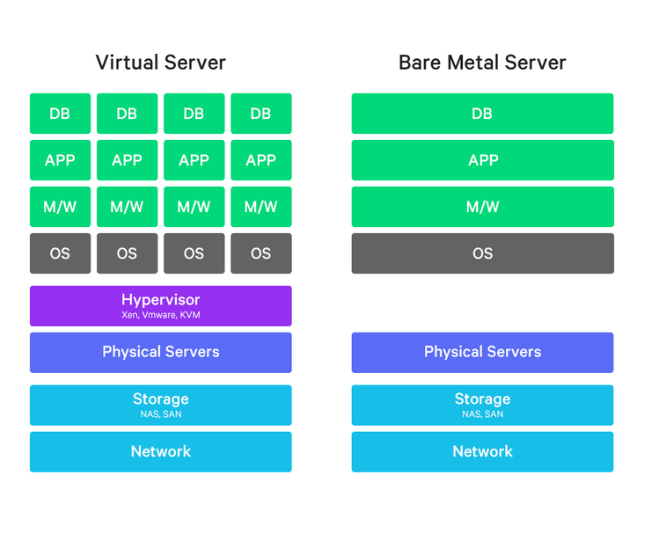
6 표현 계층 : 두 장치가 일관되게 전송 데이터를 서로 이해할 수 있게 하는 것을 나타내며, 예시로 UTF-8 과 EUC-KR 같은 제어코드의 규격, 프로그램의 확장자(jpg,gif,mpg)등이 인식되게 하는 것을 의미한다.

7 응용 계층 : 사용자가 네트워크에 접근할 수 있도록 해주는 기능을 말하며, 사용자 인터페이스라고 보면 된다. 인터넷을 사용하기 위해 브라우저를 사용한다라는 느낌. 프로토콜로는 HTTP, FTP등이 있다.

**12. 베어메탈**

12-(1) 베어메탈의 개념  
 베어메탈은 어떤 소프트웨어도 설치되어 있지 않은 상태의 하드웨어를 의미한다.

12-(2)베어메탈의 사례

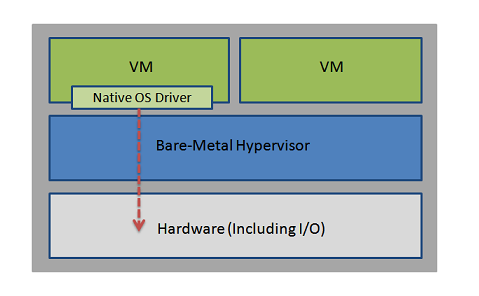
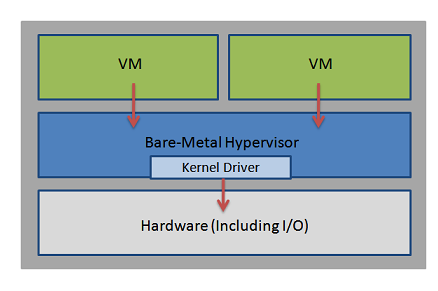


클라우드 환경에서 아키텍처는 크게 호스트형, 베어메탈형 방식이 있는데, 위에서 보듯이 차이점은 하이퍼바이저 OS의 유무에 있다. 전자는 하드웨어의 리소스를 각자 공유하는 방식으로 썼다면, 후자는 하드웨어의 모든 성능을 사용할 수 있는 방식이다.

네이버 클라우드도 이러한 베어메탈 서버를 제공하는데, 기존의 호스트형 가상서버에서는 디스크I/O가 부족해서 DB응답이 느려지는 Slow Query 등과 같은 성능 이슈가 발생했는데, 고객의 서버가 부하가 없어도 가상 서버가 탑재된 물리 서버의 부하로 인해 응답이 느려지는 현상이 발생했다. 이는 하나의 물리서버에 여러가상 서버를 공유해서 사용하면서 순간적으로 발생하는 자원 간섭 현상에 따른 영향이다. 그에 비해 베어메탈 서버는 이런 간섭이 없이 혼자 독점하기에 빠른 데이터 생성이나 응답이 가능했다, 그에 따라 비싸더라도 베어메탈형 서버를 원하는 고객들이 늘어나고 있다.

12-(3) 아키텍처에서 어떻게 적용할지

베어 메탈 가상화가 호스트 운영체제에 의존하지 않으므로 본 아키텍처를 사용하는 하이퍼 바이저는 I/O 디바이스와 직접 통신한다는 점을 유념해야 한다. VM(이더넷, 하드 드라이버등)간 I/O디바이스가 공유되기 위해서, 하이퍼바이저 소프트웨어는 로우 레벨 드라이버를 보유해야 한다. 또한 게스트 VM을 위해 각 공유되는 디바이스를 반드시 에뮬레이션할 수 있어야 한다. 베어 메탈 가상화 아키텍처에서 버추얼 머신간 공유되는 시스템 I/O 디바이스는 반드시 하이퍼 바이저의 커널 드라이드에 접근해야한다.

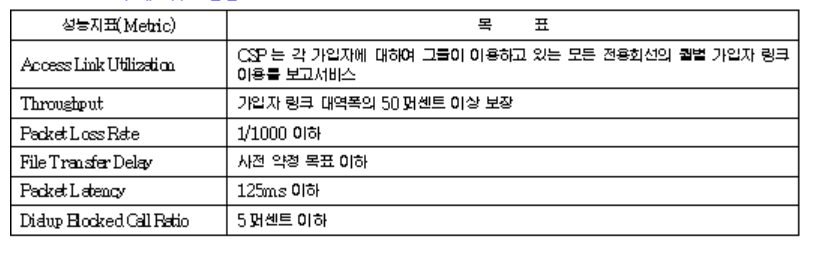


솔직히 찾긴 했는데, 이해가 너무 어렵다.

**13. SLA**  
 13-(1) SLA 개념  
 SLA란 어떤 서비스가 제공될 것인지를 측정이 가능한 지표로 명시한 문서이며, 서비스 공급자와 사용자간에 서비스에 대한 측정지표와 목표 등에 대한 협약서이다.  
  
 13-(2)SLA 실제 사례

ANX는 미국의 자동차협회(AIAG: Automotive Industry Action Group)가 발주하고 요구하는 고품질, 고도의 보안성이 확보되어야 하는 IP VPN서비스로서 자동차 산업의 발전을 위하여는 네트워크 환경을 일정 수준의 서비스 수준이 보장되는 안정적인 네트워크를 확보할 필요를 느끼게 되어 ANX를 추진하게 되었다.

ANX에서 CSP에 대하여 요구하는 서비스 성능들은 크게는 다음과 같은 8개 항목으로 구성되어 있다. 네트워크 서비스, 상호운용성, 네트워크 성능, 신뢰도, 네트워크 장애 복구, 보안, 고객지원, 문제해결로 구성되는데 그 중에 네트워크 성능 SLA를 보면,



네트워크에서 특정 성능에 대해 미리 목표를 세우고 해당 성능 지표에 가중치를 줘서, 성능의 총합이 협약에 만족했을 시 +a% 만큼의 인센티브를 CSP측에 주고, 불만족 시 –a% 만큼의 패널티를 부과하는 방식으로 쓰인다.

13-(3) 내가 클라우드 제공자라면 어떻게 SLA를 적용할지  
내가 클라우드 제공자라는 것은 물리적인 나의 서버를 여러 사람에게 나눠주는 것을 의미할 것 같은데, 이 때에, 실제로 내 서버가 상대에게 얼마나 빠른 속도로 얼마나 문제없이 돌아가느냐, 보안이나 백업관련을 얼마나 신경을 쓰고, 어떤 서비스를 제공했느냐. 정도의 관심이 쏠릴 것이다.   
이러한 경우에 속도는 IOSP를 10만을 보장하고 패널티를 9.95만부터 가하고, 실제 가동 시간을 24시간의 99.9% 미만인 경우에 패널티, 해킹관련 보안을 막았을 시에 인센티브, 뚫렸을 시 피해보상, 백업을 하루 몇 번했는지에 따른 인센티브 등을 협약서에 적용할 것 같다.