1강. IT인프라의 이해

**<클라우드의 진화>**

* 클라우드는 다양한 장치에서 생성된 대규모 데이터 수집, 분석하는 지능형 플랫폼으로 진화하고 있음

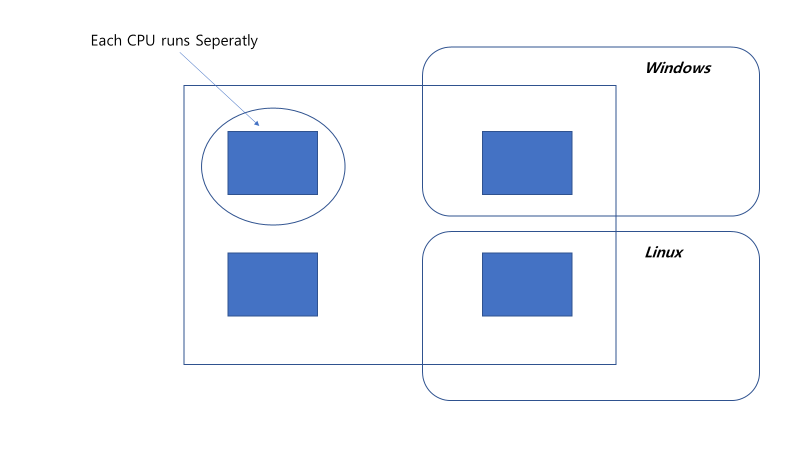
**<클라우드 기술 역량의 특징>**

|  |  |
| --- | --- |
| **전통적 IT의 역량** | **클라우드 역량** |
| * 분야별 전문역량 * 공급자의 표준화된 기술 * 중장기적 기술 혁신 주기 | * 전 분야 통합 역량 (모든 분야가 사용) * 단일 공급자의 고유 기술 * 짧은 기술 혁신 주기 * 기술이 지속적으로 바뀌어서 적응해야 함 |

**<강의내용(일부 빠진 내용 있을 수 있음)>**

1. 엔터프라이즈 컴퓨팅(Enterprise Computing)
   1. 다른 시스템보다
      1. 장애가 생겨도 서비스 유지
      2. 비교적 싸게
   2. IT 기술의 원칙(두 가지 강조하신 거!) 🡪 아래에 설명 다시 있음
      1. 모든 기기는 고장이 난다.
      2. 경제성 있게.(되도록이면 가격을 저렴하게)
   3. 신뢰성, 안전성 등이 고려되어야 한다.
2. 인프라 IT
   1. Infrastructure(인프라): 기반  
      생활을 지탱하는 바탕이나 토대
   2. IT Infrastructure(IT 인프라): IT 서비스를 지탱하는 기반
3. 서버
   1. 컴퓨터의 구조: CPU + RAM + Disk
   2. CPU의 원리  
      반도체의 특성을 이용: 0과 1을 구분 (트랜지스터) 🡪 집적회로 (IC) 🡪 CPU
   3. CPU 발전 🡺 CPU의 집적도를 높이며 발전
      1. Core가 많아지고 Thread가 증가하여, 병렬처리 가능
      2. 가상화 (CPU나 thread 개수만큼 가상의 컴퓨터 (CPU – 메모리 – 디스크) 구성)

* 이것이 클라우드의 등장 배경이자, 비용절감의 이유가 됨



[Figure 1. Virtualization(가상화의 이해도)]

* 1. IT 기술의 원칙
     1. 모든 장비는 고장 난다
     2. 기술은 경제성에 기반을 둔다
  2. 캐쉬 메모리
     1. 비용관리에서 중요
     2. 중간에 빠른 것이 중재하는 역할
     3. 비용을 이용한 성능 최적화
  3. 메모리의 속도 단위(성능기준): IOPS (Input Output Per Second)
  4. 저장 장치 중 테이프는 가격이 저렴하고, 데이터의 유실 염려가 없음
  5. RAID 5: 중간에 Parity를 삽입하여, 경제적이며, 중간 장애에도 복구 가능

1. 네트워크
   1. 포트번호: 동시에 들어온 데이터를 구분하는데 사용
   2. IP 주소의 클래스
      1. A클래스
         * 첫번째 주소비트는 0
         * 따라서, 255 – 128 = 127
         * (범위: 0 ~ 127)
      2. B 클래스
         * 첫,두번째 주소비트는 10
         * 따라서, 255 – 64 = 128
         * (범위: 128 ~ 191)
      3. C 클래스
         * 첫,두번째 주소비트는 110
         * 따라서, 255 – 32 = 192
         * (범위: 192 ~ 255)
2. IT 인프라 아키텍처
   1. 아키텍처: 구조, 기본적인 물리 설계
   2. IT 아키텍처: 어떤 기술에 대한 물리적 기초 설계
      1. 컴포넌트 아키텍처: 단일 아키텍처
      2. 애플리케이션 아키텍처: 컴포넌트간 연계
      3. 통합 아키텍처: app간 연결
      4. 엔터프라이즈 아키텍처: 종합 (1~3번 합친 형태)
3. 3계층형 시스템
   1. 2계층형 시스템: app (client) 🡪 DB (server)

: 원래는 DB서버만 있고, app은 개인이 설치하던 형태

* 1. 3계층형 시스템: web + app + DB (server)

: 서버의 부하가 생겨 app을 Computing하는 서버가 필요함

* web세계로 넘어가면서 2게층에서 3계층 시스템으로 발전
* 클라이언트의 app이 web + app(계산) 서버로 발전
  1. 주요개념
     1. 프로세스: 프로그램 자체와 상태가 메모리 상에서 실행되는 작업 단위
     2. 스레드: 프로세스 내에서 실행되는 흐름의 단위

1. 인프라르 지탱하는 기본 이론
   1. 직렬과 병렬
      1. 직렬: 여러 개의 물건이 일직선으로 나열돼 있는 것
      2. 병렬: 두 줄 이상으로 나열돼 있는 것
   2. **병렬과 분산(중요)**
      1. **병렬: 같은 일을 나눠서 하는 것**
      2. **분산: 다른 일을 나눠서 하는 것**
   3. 동기와 비동기
      1. 동기: 즉흥적인 커뮤니케이션, 상대방의 대답 대기  
         예) 전화
      2. 비동기: 즉흥적인 커뮤니케이션이 아니어도 무방  
         예) 카톡
2. 인프라를 지탱하는 응용이론
   1. 캐시: 사용 빈도가 높은 데이터나 값을 고속으로 액세스 할 수 있는 위치에 미리 복사해 놓는 임시 저장소
   2. 복제: DB나 저장소 등에서 자주 사용되는 기술, 원본을 다른 위치에 복사(실시간 동기화)
   3. 마스터와 슬레이브
      1. 마스터: 데이터의 트랜잭션이 이루어지는 주 서버
      2. 슬레이브: 마스터를 참조하여 대기하면서 마스터의 데이터를 참조 복제하는 보조 서버

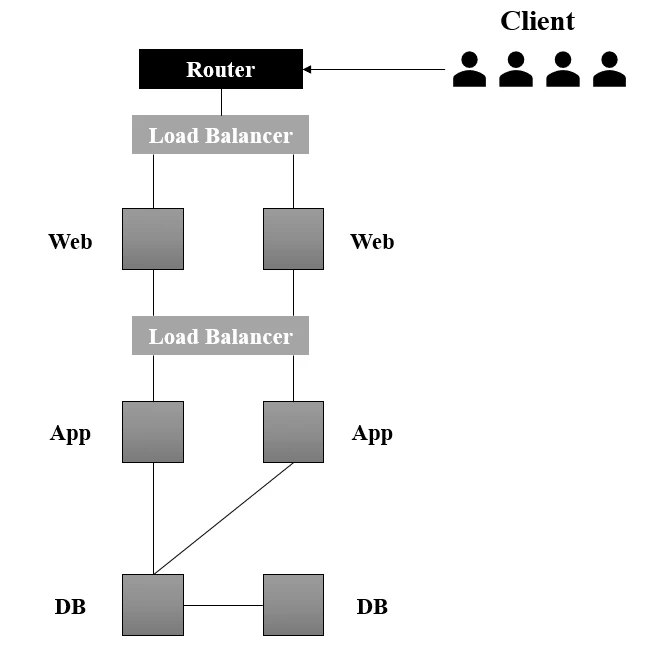
* 마스터가 정지할 경우 슬레이브를 마스터로 대체

1. 무정지를 위한 인프라 구조
   1. 고가용성(안정성)과 이중화
      1. 고가용성(High Availability): 시스템 서비스가 되도록 멈추지 않도록 하는 것
      2. 이중화: 병렬로 여러 개 나열, 하나에 장애가 발생해도 다른 것을 가동하여 서비스를 지속하도록 함, 여러 개 가동하여 부하 분산
         * 이중화의 장점

대체 가능

속도 증가

* 1. 서버의 이중화(web + App + DB)



**웹 데이터 흐름: 웹 서버 (웹 표현) 🡪 App 서버 (프로그램을 돌아 감) 🡪 DB 서버(데이터)**

<Figure 02. 웹 서버의 이중화>

* 1. 백업 🡪 특정 시점에 있는 것을 그대로 가지고 옴
     1. 복구의 목표  
        RTO(Recovery Time Objective): 복구 목표 시간 🡪 작을수록 좋음  
        RPO(Recovery Point Objective): 복구 기준 시점 🡪 주기가 잦을수록 좋음
     2. 시스템 백업
     3. 데이터(DB) 백업

1. 성능 행상을 위한 인프라 구조
   1. 응답(response): 요청에 대한 회신이 오기까지 걸리는 시간(0.5 ms)
   2. 처리량(throughput): 정해진 시간동안 요청에 대한 응답 건수(1,000 건/초)