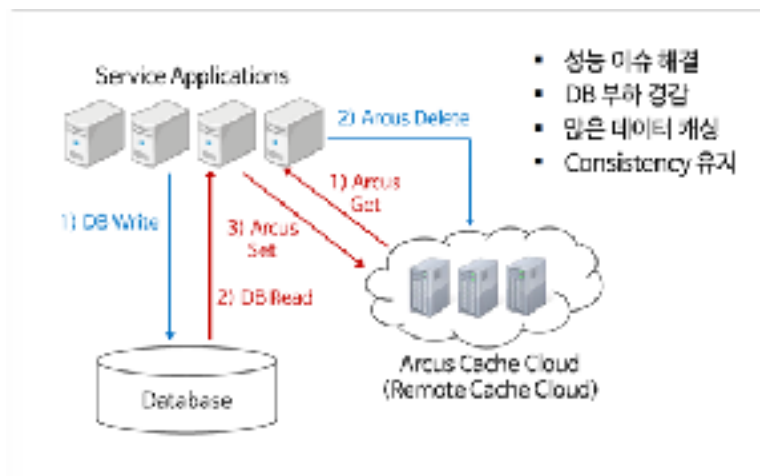


## [ ARCUS ]

ARCUS : memcached cloud with list,set,b+tree collections

- 아치형 구름
- Memcached & Zookeeper 기반의 확장한 cached cloud
- 필요성 : Large-scale web service 에서
  - 데이터와 요청 양이 증가하면
  - 성능 : 낮은 처리량 , 느린 응답
  - DB : 고비용 , scale-out이 힘들
- 결과 : DB Only -> DB + ARCUS
  - 고성능(높은 처리량 + 빠른 응답)
  - ARCUS 수평 확장으로 실시간 급증 조회 처리 가능
  - DB 부하 감소/안정화로 DB/인력 비용 절감



- 환경
  - OS : Linux Only
  - Clients : Java , C

## [ 도커 ]

- SPI 모델
  - 클라우드를 구축하기 위한 요소 기술(Enabling Tech)
    - 가상화 기술
      - 컴퓨터 자원(CPU,메모리,저장장치,네트워크 등)의 추상화
      - 서버의 리소스 가상화를 통해서 하나의 서버에 여러대의 OS를 동작시킬 수 있는 기술
      - 서버의 Utilization을 높일 수 있다.
    - 단점
      - 불필요한 기능의 중복 : host/guest OS 간의 기능 중복 in process scheduling
      - 상대적으로 무겁다 : 15~20% overhead
      - Deployment(배치)의 어려움
    - 레벨
      - API
      - ABI
      - ISA
  - 컨테이너 기반 기술 : 도커
- 클라우드 서비스
  - 예시
    - amazon aws
    - ms azure
    - n drive , drum cloud , google drive
- 도커
  - 배경
    - 하이퍼바이저
    - 도커
      - 하드웨어의 가상화가 아닌, 실행환경의 분리(isolation)
      - 각 컨테이너 간 영향을 분리
      - 하이퍼바이저와 달리 guest OS가 없기 때문에 가볍고 빠른 성능
      - 하이퍼바이저 기반의 우분투와 도커 기반의 우분투가 실제 설정/사용 방식이 상이
      - 하나의 Linux 시스템에 여러 Linux 시스템 운영을 위한 sw
  - 성능
    - 5% 이내 overhead
  - 특징
    - 모든 컨테이너가 동일 OS 커널 공유
      - 독립적인 스케줄링이나 CPU/MEM/디스크/네트워크를 가상화 하지 않음
    - 리눅스 특수기능(LXC)을 사용한 실행환경 격리를 응용
      - 리눅스에서만 사용가능
      - 다른 OS에서는 하이퍼바이저가 있어야 함
  - 사용 기술
    - 시스템의 분리 : LXC
    - 파일 시스템 : Aufs
    - 이미지 버전 컨트롤 : Git
  - LXC
    - 리눅스 컨테이너
    - 시스템 레벨의 가상화

- 요약
  - 도커란 : LXC or lib container + Aufs + 이미지,컨테이너 생성관리 + 각종 부가 기능
- .
- .
- .

## [ 멘토링 ]

- os : 리눅스가 제일 좋은 환경
- 도커 환경에서 아커스 컨테이너를 다운받으면 됨
- zookeeper : 인스턴스가 죽을 경우를 대비해서 active-standby 방식으로 백업 인스턴스를 가지고 있다.  
문제가 생기면 교체하는 방식으로 동작한다.
- 도커 테스트에 사용되는 데이터
  - mysql의 world-db 나 [lanunchpad.net](http://lanunchpad.net)의 test-db를 사용
- 스트레스 테스트를 하려면 웹 기반이 되어야 함
  - 웹 사이트를 제대로 만들면 좋지만 썰렁하다고해서 감점하지는 않음

#### [메뉴얼]

- docker mac version download
- ubuntu image download at terminal
- connect to ubuntu
- install arcus container

## [nBase-Arc]

- DB 클러스터
  - 하나의 DB를 복수개의 서버(혹은 가상서버)상에 구축하는 것
  - 고가용성 , 병렬처리 , 성능향상 등의 이유로 하나의 서버 만으로 DB를 구성하지 않음
  - 고가용성
    - 가용성 이란 , DB가 동작하고 있는 시간과 정지한 시간의 비율
    - DB 시스템을 구성할 서버, 스토리지 장비를 2대 이상으로 구성해서 하나의 장비에서 장애가 발생한 경우 , 단 기간에 운용을 재개할 수 있도록 함(고가용성 클러스터)
  - 병렬처리
    - 데이터의 양이 너무 많거나 sql문이 복잡한 경우 하나의 sql문 실행에는 막대한 시간이 걸린다. 이럴 경우 DB를 복수개의 작은 단위로 분해해서 각각을 분석하는 처리를 병렬로 실행해서 결과를 통합하는것
  - 성능향상
    - DB 유저수가 막대해지는 경우 , DB의 복사본을 만들어 참조 처리는 복사한 DB를 사용하게 하는 방법으로 문제 해결 가능