

2010.02.03 sysadminstudy 발표자료

24시간 365일 서버/인프라를 지탱하는 기술 (요약&발표 엄기성<jurist@kldp.org>)

Chapter 06: 서비스의 무대 뒤 (자율적인 인프라, 다이나믹한 시스템 지향)

## 6.1 Hatena의 내부

### 1. Hatena의 인프라

1. "자전주의(自前主義)", "오픈소스 주의" 와 같은 운영 방침과 같이 일관된 방향을 계속 유지하는 것이 Hatena만의 강점이 될 수 있다.

### 2. 확장성과 안정성

#### 1. 리버스 프록시

- 직접 개발한 mod\_dosdetector 사용하는 점에 주의. (필요하면 직접 개발해야 함)

#### 2. DB

- 이제는 mysql cluster edition 사용가능 (community & commercial edition)

#### 3. 파일서버

- NFS/CIFS가 아닌, cluster 환경에서 동시에 파일을 접근하려면 필수적으로 OCFS2를 고려하자. DBRD는 여전히 2-node로 제한됨

### 3. 운영효율 향상

#### 1. 키스타트에 의한 설치

- WDS와 함께 공존해서 사용해야 한다. 각종 boot image, firmware update, system rescue cd 등도 PXE 부팅해서 사용하자.

#### 2. 패키지 관리와 Puppet

- Asset/Inventory Management 도구가 필요하다. Hatena에서는 자체 개발을 선호함.

#### 3. 서버관리와 감시

- 마찬가지로 Asset/Inventory Management 도구가 필요하다.

#### 4. Capistrano에 의한 deploy

- 더 많이 자동화시킬수록 더 편해진다. 하지만, 몸으로 겪기 전엔 알수 없는 것이 deployment.

### 4. 전원효율 - 리소스 이용률 향상

#### 1. 1A당 성능을 증시한다.

- 전력 총 사용량에 대한 정기적인 측정이 필요하다. How ??

#### 2. 서버 1대당 성능을 최대한 끌어낸다.

- 저가형 머신 여러대가 결코 답이 아니다. 시스템의 수가 N개 증가하면, 스위치 포트/케이블/전원 포트/kvm 등의 요구항목이 얼마나 증가하는가?

#### 3. 불필요한 부품은 설치하지 않는다.

1. Diskless 환경이 아닌 client라면 무조건 RAID1+0를 사용해야 한다. 특정한 디스크의 내용을 100% 알고 있다고 확신하는 것은 무모함이다.

### 5. 자율적인 인프라 지향

1. 인프라의 최종 진화형을 어떻게 바라볼 것인지 상상(!)한다. 10을 상상하되 0.1 정도 구현하기.

## 6.2 DSAS(Dynamic Server Assign System)의 내부

### 1. DSAS란

#### 1. DSAS의 특징

#### 2. 하나의 시스템에 여러 사이트 수용

1. 단일 서버가 복수개의 서비스 제공하되, 서비스별 시스템 재구성이 신속해야 한다. 안그러면 노가다와 복잡도 증가. 그리고, 3 tier 시스템에서는 어떻게?

#### 3. OSS로 구축

1. (개발/운영/기업) 문화가 충돌하는 지점이다.

#### 4. 어딘가 끊어져도 멈추지 않는 네트워크

1. 복잡도와 신뢰도, 비용 사이에서 균형 잡기. 원칙이 중요해진다.
5. 간편한 서버중설
  1. deploy 성숙도에 비례한다. 그 이상도 그 이하도 아님. --> 모르면 모두 노가다!
6. 간단한 장애복구
  1. 모니터링 수준과 비례한다. 내게 보이는 것 이상으로 장애 복구를 할 수는 없다. 모르면 재부팅하게 될 뿐.
2. 시스템 구성 상세
  1. Bonding 드라이버를 이용하는 이유
    1. 포트 비용도 생각해야 한다. 그리고, multi-path 구성을 어디까지 할 것인가?
  2. DBRD를 장애극복할 때의 주의점
    1. 2-node 클러스터로 생각하기.
  3. SSL 가속기
    1. 어딘가에서는 SSL offload를 처리해야 한다. Appliance, Server의 CPU, 또는 ???
    2. L7 layer 에서 패킷을 제어하고 싶은 need가 있는가?
  4. 헬스체크 기능의 확장
    1. 생략
  5. 간편하고 안전하게 운용할 수 있는 로드밸런서
    1. 생략
  6. 세션 데이터 처리 (memcached, repcached)
    1. 개발팀과 꼬옥 상의하고 잘 꼬셔서 적용하자.
3. DSAS의 미래
  1. Dynamic 이란 말을 남용하지 말자.
  2. 80% Automated / 20% Manual 정도가 좋아 보인다.