

# 서버의 가용성 분석에 관한 연구

정성재\*, 오선우\*\*, 성경\*\*\*, 배유미\*\*\*\*

## 요약

최근 클라우드 컴퓨팅에 대한 관심이 높아지고, 서버 가상화 기술을 이용한 시스템 구성이 보편화되고 있다. 그러나, 기존에 운영중인 서버의 가용성(Availability)에 대한 분석은 매우 부족한 실정이다. 이러한 가용성은 서버의 자원(Resource)과 부하(Load)와 밀접한 관계가 있다. 본 논문에서는 물리적으로 분할되어 운영중인 WAS(Web Application Server), WEB, Database 서버의 자원과 부하(Load)를 분석하기 위해 5가지 항목인 CPU 사용률, 메모리 사용량, Disk I/O, Network 대역폭, Load Average를 기준으로 설정한다. 5가지 항목과 관련된 상태를 수집 및 통계를 위해 sar를 이용하고, ksar를 사용하여 도식화된 분석을 한다. 이 내용을 토대로 운영중인 서버들의 자원과 부하를 분석하여 서버 가상화 환경 구축의 기반 지식을 제공한다.

## A Study on Availability Analysis of Server

Sung-Jae Jung\*, Sun-Woo Oh\*\*, Kyung Sung\*\*\*, Yu-Mi Bae\*\*\*\*

## ABSTRACT

Interest is elevated about cloud computing lately, System line-up by server virtualization is becoming common method. But the actual circumstance is that to analyze existing operated system server Availability is too insufficient. The availability is closely related to Resource and Load. This paper will set the 5 item as criterion to analyze Resource and Load that is CPU usage, memory usage, Disk I/O, Network bandwidth, Load Average to analyze physically divided operated WAS(Web Application Server), WEB, Database server. It is used that sar for 5 item related status collecting and statistics and for the graphical analysis, ksar will be used. Based on these contents, operated server Resource and Load is analyzed and the knowledge from this is offered as server virtualization basic setup information.

Key Words : Server Virtualization, Resource Usage, sar, ksar, System Analysis

---

\* (주)스کم씨엔에스 (✉posein@paran.com)

\*\* 숭실대학교 정보과학대학원 정보통신융합학과

\*\*\* 목원대학교 컴퓨터교육과

\*\*\*\* 한남대학교 컴퓨터공학과

· 제1저자(First Author) : 정성재 · 교신저자(Correspondent Author) : 배유미

· 접수일(2011년 6월 8일), 수정일(1차 : 2011년 7월 7일), 게재확정일(2011년 7월 11일)

## I. 서론

클라우드 컴퓨팅에 대한 관심이 높아지면서, 여러 대의 물리적 시스템을 하나로 통합해주는 서버 가상화(Server Virtualization)기술을 활용한 시스템 구축이 보편화되고 있다[1]. 서버 가상화 기술을 이용하면 자원의 효율성, 관리의 편리함, 공간 절약 등의 많은 장점이 가지고 있지만, 기존에 여러 대의 서버를 운영하는 곳에서는 가상화 환경으로의 전환이 쉽지 않다[2]. 특히 서버를 통합하기 위해서는 시스템의 유휴자원(Idle Resource)과 시스템 및 네트워크 부하(Load) 등 서버의 가용성 분석이 선결되어야 한다.

본 논문에서는 국내의 한 업체에서 물리적으로 분할되어 운영중인 Web Application Server(이하 WAS), WEB, Database(이하 DB)의 가용성 분석을 위해 서버의 자원 및 부하와 관련된 5가지 기준항목을 설정한다. 5가지 기준 항목은 CPU 사용률, 메모리 사용량, Disk I/O, Network 대역폭, Load Average로 정하고 관련된 값의 수집과 통계를 위해 sar를 사용한다[3]. sar로 수집된 데이터는 ksar를 이용하여 도식화된 자료로 변환하여 좀 더 보기 쉽게 가용성을 분석할 수 있도록 한다[4]. 이러한 내용을 토대로 운영중인 서버를 통합하여, 서버 가상화 환경 구축의 기반 지식을 제공한다.

## II. 관련연구

### 2.1 시스템의 부하와 자원 분석

운영중인 시스템들을 서버 가상화 기반 환경으로 통합하려면 시스템의 자원(Resource)과 부하(Load)에 대한 분석이 필수적이다[5]. 자원 분석의 대상은 크게 CPU와 Memory를 들 수 있고, 시스템에 발생하는 부하에는 CPU 부하, I/O(Input/Output)가 있다. 또한 원활한 서비스와 관련된 Network 부하가 있다. 시스

템의 가용성을 분석하기 위해서는 CPU, Memory, Disk I/O, Network, Load Average 등 5가지 항목에 대한 분석이 요구된다. 분석하는 기준을 살펴보면 CPU, Memory, Network, Disk I/O는 사용량을 기반으로 분석하고, 발생하는 부하는 Load Average 수치 값을 분석함으로써 알 수 있다.

### 2.2 시스템 부하 및 자원 분석 도구

#### 2.2.1 sar

시스템의 자원과 부하를 분석하는 명령어는 iostat, sysstat, mpstat, pidstat, sar, tload, vmstat 등을 들 수 있다. 이 명령어들은 sysstat와 procs라는 패키지에 포함되어 있는데, 텍스트 기반에서 실행시 상태의 값을 출력해준다[6]. 이 중 CPU 사용률과 I/O 대기률 등을 통합해서 출력해주는 명령어가 sar(system activity reporter)이다. sar는 시스템의 상태를 수집하고 활동 상태를 누적시켜 프로그램 실행한 날짜 기준으로 10분 주기로 해당 값을 출력한다. sar를 이용하여 모니터링(Monitoring) 가능한 항목은 다음의 표 1과 같다.

표 1. sar 모니터링 가능 항목  
Table 1. The Monitoring Items of sar

- I/O 전송량
- 페이징
- 프로세스 생성 숫자
- 블록 디바이스 활동
- 인터럽트
- 네트워크 통계
- run 큐 및 시스템 부하 평균
- 메모리와 스왑 공간 활용 통계
- 메모리 통계
- CPU 이용도
- 특정 프로세서에 대한 CPU 이용도
- inode, 파일, 기타 커널 테이블에 대한 상태
- 시스템 스위칭 활동(context switch)
- 스와핑 통계
- 특정 프로세스 통계
- 특정 프로세스의 자식 프로세스 통계
- TTY 디바이스 활동

당일의 기록은 스케줄링 프로그램인 크론(Cron)에 의하여 /var/log/sa 디렉토리안에 기록된다[7]. 예를 들어 30일에 누적한 데이터는 /var/log/sa/sar30 이라는 텍스트파일에 기록되어져서 수시로 확인할 수 있다.

### 2.2.2 ksar

ksar는 sar에서 수집한 데이터를 도식화하여 GUI 형태로 전환시켜주는 도구로 사용자가 그래프를 통한 추이변화를 쉽게 확인할 수 있도록 지원해준다. Java 환경만 지원된다면 보편적으로 사용하는 윈도우 환경에서도 손쉽게 분석가능하다.

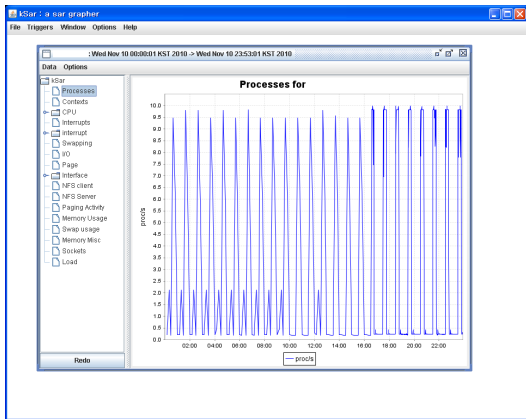


그림 1. ksar 실행 화면  
Fig 1. Execution Screen of ksar

ksar에서 지원하는 데이터 항목들은 다음의 표 2부터 표 6와 같다.

표 2. ksar의 CPU 및 swap 항목  
Table 2. CPU & swap Items of ksar

|                    |  |
|--------------------|--|
| Process (proc/s)   | 초당 새롭게 만들어져 활동하는 process의 수                    |
| Contexts (cswch/s) | 초당 context switching 수                         |
| CPU (User)         | 사용자 레벨(application level)에서 실행중 일때의 CPU 사용률(%) |
| CPU (System)       | 시스템레벨(kernel)에서 실행중일때의 CPU 사용률(%)              |

|                      |   |
|----------------------|---|
| CPU (Waiting I/O)    | 시스템 disk I/O 요청 처리중에 CPU가 idle인 시간의 % (%iowait)     |
| CPU (Steal)          | CPU들 혹은 가상 CPU에 의해 소요된 대기 시간 (%)                    |
| CPU (Nice)           | 사용자 레벨(application level)에서 nice 가 중치를 둔 CPU 사용률(%) |
| CPU (Idle)           | CPU가 쉬고있는 시간의 %                                     |
| Interrupts (intr/s)  | 초당 발생한 interrupt 수                                  |
| Swapping (pswpin/s)  | 초당 swap in 수  |
| Swapping (pswpout/s) | 초당 swap out 수                                       |

표 3. ksar의 I/O 및 Network 항목  
Table 3. I/O & Network Items of ksar

|                     |  |
|---------------------|--|
| I/O (Transfer/s)    | 물리적 디스크에서 발생되어진 초당 block(512bytes) 전송수       |
| I/O (Block read/s)  | 드라이브 안의 블럭에서 초당 읽은 데이터의 총합                   |
| I/O (Block write/s) | 드라이브 안의 블럭에서 초당 쓰여진 데이터의 총합                  |
| I/O (Writes/s)      | 물리적 디스크로부터 발생된 초당 쓰기 총 요청 횟수                 |
| I/O (Read/s)        | 물리적 디스크로부터 발생된 초당 읽기 총 요청 횟수                 |
| Page (frmpg/s)      | 시스템에서 초당 idle된 memory page의 양                |
| Page (bufpg/s)      | 시스템에서checkd buffer에 추가적으로 더해진 memory page의 양 |
| Page (campg/s)      | 초당 캐쉬되어 추가된 memory page이 양                   |
| Interface (rxpck/s) | 초당 받은 패킷 수                                   |
| Interface (txpck/s) | 초당 전송한 패킷 수                                  |
| Interface (rxbyt/s) | 초당 받은 bytes                                  |
| Interface (txbyt/s) | 초당 전송한 bytes                                 |
| Interface (rxcmp/s) | 압축된 패킷을 초당 받은 수                              |
| Interface (txcmp/s) | 압축된 패킷을 초당 전송한 수                             |
| Interface (rxmst/s) | 초당 받은 다중 패킷 수                                |

표 4 ksar의 NFS 항목  
Table 4. NFS Items of ksar

|                        |                           |
|------------------------|---------------------------|
| NFS client (call/s)    | 패킷전송중 초당 발생한 총돌수          |
| NFS client (retrans/s) | 초당 재전송되어야할 RPC 요청 수       |
| NFS client (read/s)    | 초당 RPC calls을 'read'한 수   |
| NFS client (write/s)   | 초당 RPC call을 'write'한 수   |
| NFS client (access/s)  | 초당 RPC call을 'access'한 수  |
| NFS client (getatt/s)  | 초당 RPC call을 'getattr'한 수 |
| NFS Server (scall/s)   | 초당 수신된 RPC 요청 수           |

|                        |                               |
|------------------------|-------------------------------|
| NFS Server (packet/s)  | 초당 수신된 N/W 패킷수                |
| NFS Server (udp/s)     | 초당 수신된 UDP 수                  |
| NFS Server (tcp/s)     | 초당 수신된 TCP 수                  |
| NFS Server (hit/s)     | 초당 cache hits의 응답 수           |
| NFS Server (miss/s)    | 초당 cache misse의 응답 수          |
| NFS Server (sread/s)   | 초당 수신된 RPC call을 'read'한 수    |
| NFS Server (swrite/s)  | 초당 수신된 RPC call을 'write'한 수   |
| NFS Server (saccess/s) | 초당 수신된 RPC call을 'access'한 수  |
| NFS Server (sgetatt/s) | 초당 수신된 RPC call을 'getattr'한 수 |

|                  |  |
|------------------|--|
| Sockets (rawsck) | 현재 사용중인 RAW sockets 수                          |
| Load (runq-sz)   | CPU run queue(Dispatch queue)에 기다리고 있는 thread수 |
| Load (plist-sz)  | 프로세스 리스트에 있는 프로세스 와 쓰레드 총 수                    |
| Load (load 1mn)  | 최종 1분동안 평균 System load                         |
| Load (load 5mn)  | 최종 5분동안 평균 System load                         |
| Load (load 15mn) | 최종 15분동안 평균 System load                        |

표 5. ksar의 메모리 항목  
Table 5. Memory Items of ksar

|                            |  |
|----------------------------|--|
| Paging Activity (pgpin/s)  | 디스크로부터 초당 paged in된 page의 총 수                |
| Paging Activity (pgout/s)  | 디스크에 초당 paged out된 page의 총 수                 |
| Paging Activity (fault/s)  | 초당 fault된 page이 총 수                          |
| Paging Activity (majflt/s) | 하드디스크로부터 메모리 page를 요청할때 초당 중요한 system fault수 |
| Memory Usage (memfree)     | 사용가능한 총 메모리의 양(kbytes)                       |
| Memory Usage (memused)     | 사용중인 총 메모리의 양(kbytes)                        |
| Memory Usage (%memused)    | 사용된 메모리의 %                                   |
| Swap usage (kbswpfree)     | 사용가능한 스왑공간의 양 (kbytes)                       |
| Swap usage (kbswpused)     | 사용된 스왑공간의 양 (kbytes)                         |
| Swap usage (%swpused)      | 사용된 스왑공간의 양(%)                               |
| Swap usage (kbswpcad)      | kbytes인 캐시된 swap memory 양                    |
| Memory Misc (buffers)      | buffer로 사용되어지는 메모리양 (bytes)                  |
| Memory Misc (cached)       | 커널에 의해 cache data로 사용되어지는 메모리양 (bytes)       |

표 6. ksar의 소켓 및 부하 항목  
Table 6. Sockets & Load Items of ksar

|                   |                        |
|-------------------|------------------------|
| Sockets (totsck)  | 총 사용된 socket 수         |
| Sockets (ip-frag) | 현재 사용중인 IP fragments 수 |
| Sockets (tcpsck)  | 현재 사용중인 TCP sockets 수  |
| Sockets (udpsck)  | 현재 사용중인 UDP sockets 수  |

### III. 사례를 통한 가용성 분석

#### 3.1 개요

국내의 한 업체에서 가장 중요한 역할을 수행하고 있는 WAS(Web Application Server), WEB, DB Server 를 선정하여 1분 단위로 24시간동안 기준으로 하고, 요일 및 월별 등의 로그를 분석한다. 분석하는 요소는 CPU와 Memory의 사용량, Disk I/O, Network 사용량, Load Average를 선정하여 중점적으로 검사한다.

표 7. 시스템 개요  
Table 7. Overview of System

|      |  |
|------|--|
| 대상   | 현재 서비스중인 WAS, WEB, DB 서버                     |
| 요소   | CPU, Memory, Disk I/O, Network, Load Average |
| 분석기간 | 1분 단위로 24시간 분석,                              |
| 특이사항 | 요일별 분석, 월별 분석                                |

분석은 sar로 데이터를 수집하고, ksar를 통해 도식화된 결과값을 가지고 통계 및 분석한다.

#### 3.2 시스템 분석 단계

시스템의 가용성을 분석하기 위해서는 최적의 상태 인지를 점검해야 한다. 시스템의 성능 분석을 위한 단계는 다음의 그림처럼 4단계로 나눌 수 있다.

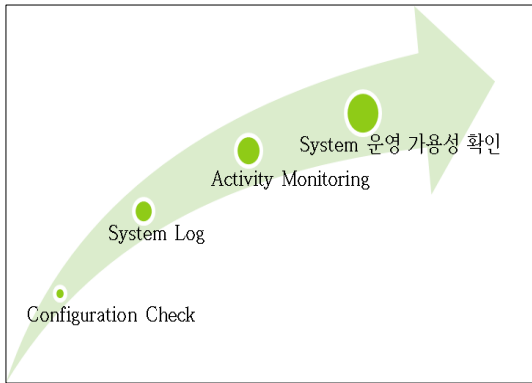


그림 2. 성능분석 흐름도  
Fig 2. Flowchart of Performance Analysis

Configuration Check는 운영중인 응용프로그램의 환경 설정파일이나 운영체제의 커널 파라미터(Kernel Parameter)분석이 해당된다. 환경 설정 파일에는 성능 향상을 위한 항목들이 존재하여 사용자들에 선택할 수 있도록 되어 있고, 리눅스 운영체제에서는 /proc/sys 디렉토리에 존재하는 파일값들을 조정함으로써 더 나은 성능향상을 고려해볼 수 있다. 관련 항목값을 최적화함으로써 시스템의 가용성을 더욱 높일 수 있다.

System Log 단계에서는 응용프로그램이나 시스템 로그를 분석하여 오류 제거, 접속자 제어 등을 하고, 불필요한 프로그램 제거를 통한 자원의 효율성 증대 시킬 수 있다.

Activity Monitoring은 운영중인 시스템 성능에 중요한 영향을 미치는 요소들의 사용량 변화 추이를 살펴보는 단계이다. 서버의 가용성과 연관이 있는 Processor, Memory, Disk I/O, Network, Load Average 등의 사용률과 패턴을 분석한다. 이 단계에서 분석 요소를 정하고, 분석기간과 시간에 대해 기준을 잡는다. 서버는 24시간동안 연중 운영되고, 짧은 시간 동안 접속자가 급격히 늘어나는 경우가 있다. 정확한 결과값을 도출해 내기 위해서는 1분 단위로 24시간 분석이 필요하고 요일별 및 월별 분석 등도 병행해야 한

다. 분석대상이 되는 요소별로 가장 사용률이 높은 시점을 대상으로 시스템의 가용성을 분석한다.

시스템 분석을 위한 순서도는 다음의 그림과 같은 단계를 거쳐 결과를 도출한다.

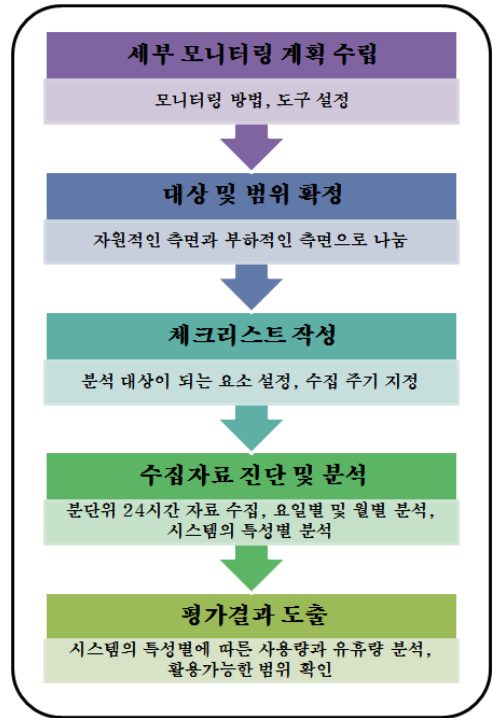


그림 3. 시스템 분석 순서도  
Fig 3. Flowchart of System Analysis

### 3.3 시스템 분석

#### 3.3.1 CPU 분석

CPU의 사용률은 3대의 서버중에 DB 서버가 가장 높은 사용률을 보였다. 최대 사용량을 살펴보면 WAS의 2%, WEB 서버는 13%, DB 서버는 25%의 사용량을 보였다. 다음의 그림 4는 DB 서버의 CPU 사용량을 분석한 내용이다. %used, %niced, %idle 3개의 항목을 보여주며 맨 위의 항목이 사용량인 %used이고 시간대별로 확인할 수 있다. 마지막 항목이 사용가능한 CPU 자원인 %idle값이다.

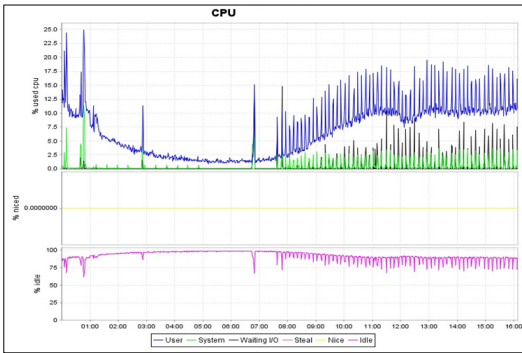


그림 4. DB 서버의 CPU 사용량  
Fig 4. CPU Usage of DB Server

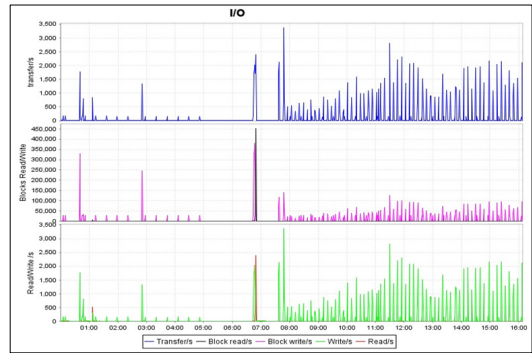


그림 6. DB 서버의 Disk I/O  
Fig 6. Disk I/O of DB Server

### 3.3.2 Memory 분석

메모리의 사용률도 역시 DB 서버가 가장 높은 수치를 보였다. 전체 메모리 용량인 64GB의 83% 정도를 사용한다고 나오지만, 사용량중 절반이 Swap과 Cache로 사용되어 실제 사용량은 많지 않음을 알 수 있다.



그림 5. DB 서버의 Memory 사용량  
Fig 5. Memory Usage of DB Server

### 3.3.4 Network

Network의 사용량은 WEB 서버가 가장 높은 수치를 보인다. 다음의 그림을 살펴보면 네트워크 사용량은 CPU 사용량과 같은 패턴으로 증가와 감소를 반복하는데, 순간 최대 사용량은 5MB를 넘지 않는 것을 알 수 있다.

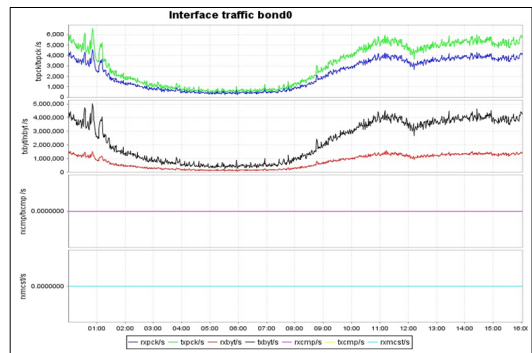


그림 7. WEB 서버의 Network 사용량  
Fig 7. Network Usage of WEB Server

### 3.3.3 Disk I/O

디스크에 대한 입출력 수치도 DB 서버에서 가장 높게 나타난다. 디스크에 대한 읽기 및 쓰기 작업이 지속적으로 일어나지만 Load Average의 수치가 높아지지 않는 것으로 보아 병목 현상을 발생시키기에는 미미한 수준으로 볼 수 있다.

### 3.3.5 Load Average

Load Average값은 3대의 서버중에 WEB 서버의 수치가 가장 높게 나타난다. 다음의 그림 8을 살펴보면 평균적으로 2이하의 값을 가지고 최대 부하시의 수치도 3.5를 넘지 않아 부하가 크지 않을 것을 알 수 있다.

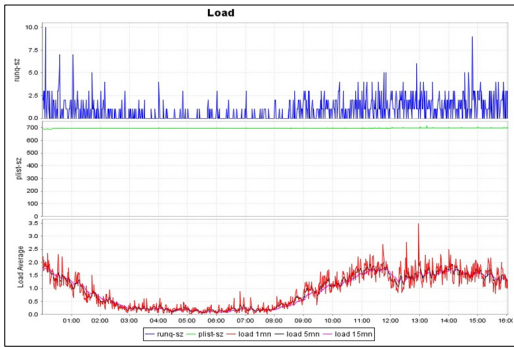


그림 8. WEB 서버의 Load Average  
Fig 8. Load Average of WEB Server

### 3.4 결과 분석

운영중인 WAS, WEB, DB 서버를 분석한 결과 WAS 서버는 CPU 사용률도 높지 않고, 메모리 사용량의 변화가 거의 없고, Disk I/O 및 Network 사용량, Load Average값도 일정하다. DB 서버는 CPU 및 Memory 사용량에서 높은 수치를 보이고 있고, WEB 서버는 네트워크 사용량에서 상당히 높은 수치를 보이고 있다. 각 항목별 값은 다음의 표 8과 같다.

표 8. 결과 분석  
Table 8. Result Analysis

| 항목                    |    | WAS  | WEB       | DB      |
|-----------------------|----|------|-----------|---------|
| CPU 사용률               | 최대 | 1%   | 13%       | 25%     |
|                       | 평균 | 0.5% | 6%        | 17%     |
| Memory 사용량            | 최대 | 4GB  | 14.4GB    | 26.56GB |
|                       | 평균 | 4GB  | 12GB      | 19GB    |
| Disk I/O (Transfer/s) | 최대 | 2.25 | 68        | 3,500   |
|                       | 평균 | 1.50 | 30        | 1,000   |
| Network (bytes/s)     | 최대 | 750  | 5,000,000 | 400,000 |
|                       | 평균 | 500  | 3,000,000 | 300,000 |
| Load Average          | 최대 | 0.6  | 3.5       | 3.0     |
|                       | 평균 | 0.3  | 1.5       | 1.5     |

현재 물리적으로 분할되어 구성된 WAS, WEB, DB 서버의 CPU 사용률을 살펴보면 통합해서 운영해도

여유가 있다고 볼 수 있다. 아울러, 메모리 사용량과 네트워크 대역폭은 최대치를 합산하여 할당하면 운영상의 무리가 없고, Disk I/O나 Load Average도 서버 가상화 기술을 이용하여 통합해도 큰 문제가 없을 것으로 여겨진다.

## IV. 결론

운영중인 서버들을 자원의 통합과 효율적인 관리를 위해 서버 가상화 기반으로 전환하려면 자원과 부하에 정확한 분석이 필요하다. 자원적인 측면에서는 CPU와 Memory 사용량을 분석해야 하고, 부하 분석 측면에서는 Disk I/O, Network, Load Average 값의 분석이 요구된다. 5가지 항목을 기반으로 월별, 요일별, 시간대별로 최대 및 평균 수치를 구하고 가용성 분석한다면 자원의 효율적인 운영과 통합관리의 기반을 마련할 수 있다. 서버 가상화 기술을 활용한 서버 통합은 관리의 편리성과 자원의 효율성뿐만 아니라, 클라우드 컴퓨팅 환경 구축을 위한 초석이 될 수 있다.

본 논문에서 분석한 내용을 토대로 운영중인 서버들의 자원과 부하를 분석한다면 자원 통합 기반 환경 구축뿐만 아니라 클라우드 컴퓨팅 인프라 구축에도 많은 도움이 되리라 여겨진다.

## 참고문헌

- [1] A white paper from IBM, "가상화 기술백서 - IBM Systems Agenda와 가상화 엔진을 중심으로", pp. 10-13, 2006.
- [2] 정성재, 배유미, 소우영, 성경, "x86 시스템에 최적화된 서버 가상화 연구", 한국지식정보기술학회 논문지, 제5권, 제5호, pp 131-139, 2010년 10월.
- [3] sysstat, <http://sebastien.godard.pagesperso-orange.fr/>
- [4] ksar, <http://sourceforge.net/projects/ksar/>
- [5] 정성재, 배유미, 장래영, 성경, 소우영, "클라우드 컴퓨팅 인프라 구축을 위한 시스템 부하 및 자원에 관한 연구", 한국

해양정보통신학회 춘계학술대회 논문집, 제15권, 제1호, pp. 440-443, 2011년 5월.

[6] Procps, <http://procps.sourceforge.net>

[7] cronie, <https://fedorahosted.org/cronie>



배유미 (Yu-Mi Bae)

2005년 한남대학교 컴퓨터멀티미디어  
학과 학사

2007년 한남대학교 정보기술학과 석사

2008년~현재 한남대학교 컴퓨터공학과 박사과정

※ 관심분야: 리눅스, 보안, 서버가상화, 멀티미디어 등

### 저자소개



정성재 (Sung-Jae Jung)

1998년 한남대학교 컴퓨터공학과 학사

2003년 한남대학교 컴퓨터공학과 석사

2005년 한남대학교 국제IT교육센터

전임강사

2011년 한남대학교 컴퓨터공학과 박사

2011년~현재 (주)스컴씨엔에스 부장

※ 관심분야: 리눅스, 정보보호, 서버 가상화, 운영체제,  
클라우드 컴퓨팅 등



오선우 (Sun-Woo Oh)

2009년 건양대학교 정보전산학과

2011년 숭실대학교 정보과학대학원

정보통신융합과 석사

2011년~ 현재 유티모스트아이엔에스(주) 대리

※ 관심분야: 리눅스, 보안, 서버 가상화, 운영체제 등



성경 (Kyung Sung)

2003년 한남대학교 컴퓨터공학과 박사

1994년 동해대학교 컴퓨터공학과 교수

2004년~현재 목원대학교 컴퓨터교육과 교수

※ 관심분야: 정보보호 및 정보관리, 컴퓨터네트워크, 신  
경희로망, 컴퓨터교육