리소스 측정

오현주

숙명여자대학교
SOOKMYUNG WOMEN'S UNIVERSITY

1. CPU 성능 측정

- Sysbench를 이용한 성능 측정

```
$ sysbench --test=cpu --num-threads=4 --cpu-max-prime=2000 run
```

  - sysbench 옵션(※ 버전에 따라 파라미터명이 살짝 달라짐)
    - --test=cpu: 실행할 테스트의 이름(cpu는 CPU performance test 진행)
      - cpu test는 지정한 값까지 소수를 계산함(2~지정한값의 제곱근 사이의 숫자들로 standard division을 함)
    - --num-threads: 스레드 개수 지정.(CPU 코어 수만큼 설정. 하이퍼 스레딩이 이루어진다면 x2로 계산)
    - --cpu-max-prime: 지정된 값까지 소수 계산. 보통 20000을 이용.
    - run: 실제 테스트 수행
      - clean: 테스트 데이터 삭제

2. 메모리 성능 측정

- free 커맨드 사용
  - m 옵션: MB 단위로 표시

```
$ free  -m
```

  - 표시하는 정보
    - total: 어플리케이션에서 사용할 수 있는 총 메모리 양
    - used: 사용된 메모리 양(used = total - free - buffers - cache)
    - free: 사용하지 않은 메모리
    - buff/cache: kernel buffer와 page cache와 slabs에서 사용하는 결합된 메모리 양(w 옵션을 통해 buffer와 cache를 다른 column으로 출력할 수 있음)
      - 애플리케이션에 필요한 경우 언제든지 회수 가능
    - available: 새 어플리케이션을 시작하는데 사용할 수 있는 메모리 양의 추정치(swapping을 제외한)

참고 사이트: https://support.site24x7.com/portal/en/kb/articles/how-is-memory-utilization-calculated-for-a-linux-server-monitor

3. 스토리지 성능 측정

- hdparm buffered을 이용한 성능 측정

```
$ sudo hdparm -t /dev/mmcblk0
```

  - hdparm -t {장치명}
    - **버퍼링이 전혀 되어있지 않은 데이터**를 얼마나 빠르게 읽을 수 있는 가를 확인하기 위함
    - 의미있는 결과를 위해 사용 가능한 메모리(a couple of megabytes)가 있는 상태에서 아무 프로세스가 실행되지 않은 상태에서 **2~3번 정도 반복해서 실행한 후 출력**
    - 드라이브가 파일 시스템 오버헤드 없이 리눅스에서 **sequential data reads**를 얼마나 빨리 유지할 수 있는지를 나타냄
    - 실제로 디스크에 존재하는 데이터를 읽는 속도 확인
    - 정확한 측정값을 위해 버퍼 캐시가 실행 도중 flush됨

3. 스토리지 성능 측정

- dd write

```
$ dd if=/dev/zero of=${USER_HOME_PATH}/test bs=8k count=50k conv=fsync; rm -f ${USER_HOME_PATH}/test
```

- 한 장소(if)에서 다른 장소(of)로 데이터를 복사함
- 파일 시스템의 캐시가 충분히 클 경우 디스크 속도 비교를 하기에 좋지 않음
  - OS가 실제로 드라이브에 데이터를 쓸 수 있을 만큼 count parameter가 충분히 크게 설정되어 있는지 확인해야함
- 파라미터
  - bs: block-size, dd 명령이 사용할 블록의 크기를 나타냄
  - count: dd 명령에서 읽을 블록 수 (k=KBytes(x1024) 의미)
  - conv=fsync: dd 명령이 매번 쓸 때마다 디스크에 데이터를 flush하도록 함
    - conv=fdatasync: dd 명령이 종료되기 전에 데이터 flush

3. 스토리지 성능 측정

- dd write

```
$ dd if=/dev/zero of=${USER_HOME_PATH}/test bs=8k count=50k conv=fsync; rm -f ${USER_HOME_PATH}/test
```

  - **테스트하기 전**, cache data를 flush하기

```
$ echo 3 | sudo tee /proc/sys/vm/drop_caches
```

3. 스토리지 성능 측정

- iozone 4K random read/write

```
$ ./iozone -e -I -a -s 100M -r 4k -i 0 -i 1 -i 2
```

  - 옵션
    - a: 테스트가 자동으로 진행됨
    - s: file size를 정하는 옵션(디폴트로는 64KB~512MB까지 테스트가 진행됨)
    - r: record length를 정하는 옵션(디폴트로는 4K~16M까지 테스트가 진행됨)
    - i: 특정 테스트 종류를 지정할 수 있는 옵션
      - 0: write/rewrite
      - 1: read/re-read

```
                                                    random    random     bkwd   record   stride
      kB  reclen    write  rewrite    read    reread    read     write     read  rewrite     read   fwrite frewrite   fread  freread
  102400       4     3855     2948     8833     9029     8441     1843
```