

하드디스크 데이터 I/O 속도 측정용 유틸리티 blockwrite 개발과 응용

김효령* · 송민규**

Development and Application of HDD I/O Measurement Utility Blockwrite

Hyo-Ryoung Kim* · Min-Gyu Song**

요 약

하드디스크의 속도 성능을 탐색해 볼 수 있는 유틸리티를 개발하였다. 하드디스크에 적용하여 속도 곡선을 자세히 볼 수 있음을 보이고, 이동평균 방법을 적용하면 하드디스크의 실린더 구조와 속도 곡선의 변화를 선명하게 볼 수 있도록 하였다. 확장성을 위해 일반 대용량 스토리지에 적용하였으며, 최근에 새로운 저장 매체로 각광 받고 있는 SSD의 I/O 성능을 측정하였다. 10Gbps급 이상의 속도를 보이는 M.2 NVME를 이용하여, 리눅스 O/S에서 제공하는 cp와 속도 비교 실험을 수행함으로써, 유틸리티의 신뢰성을 검증하였다.

ABSTRACT

In order to investigate the speed profile of data I/O of HDD, we have developed an utility program. The application to HDD reveals the detail properties of the speed profile of HDD and the relation between the cylinder structure of HDD and the velocity profile. For the extent application, the experiment of the large volume storage was performed, and the profile of SSD media, which is known as the new rapid media, was measured. The new M.2 NVME SSD, which has the ability of over 10Gbps, we can compare the velocities between cp under linux O/S and the utility, and shows that the performance of the utility can be reliable.

키워드

HDD I/O Speed, Storage Speed, SSD, NVME

하드 디스크 입출력 속도, 스토리지 입출력 속도, 에스에스디, 엔브이엠이

1. 서 론

한국우주전파관측망(KVN:Korea VLBI Network)은 세계 최초로 4 채널 동시 관측을 할 수 있는 시스템을 개발하여 운용하고 있다. 데이터 생성은 한 채널마다 1024M 샘플링을 수행하여 2비트 양자화 데이터로

저장한다[1]. 즉 데이터 생성 속도는 채널당 2Gbps이며 4채널 동시 관측 시스템이므로 총 8Gbps의 속도로 데이터를 생성하고 이를 하드디스크 어레이에 기록한다[2]. 각 관측소에 기록된 데이터는 크레오넷망을 이용하여[3] 대전 데이터 센터로 전송되어 저장된다[4,5]. 데이터 센터에 도착한 데이터는 모두 모아지

* 한국천문연구원 전파천문본부(hrkim@kasi.re.kr)

• Received : Oct. 28, 2020, Revised : Nov. 22, 2020, Accepted : Dec. 15, 2020

** 교신저자 : 한국천문연구원 전파천문본부(mksong@kasi.re.kr)

•Corresponding Author: Hyo-Ryoung Kim

• 접수 일 : 2020. 10. 28

Division of Radio Astronomy, Korea Astronomy and Space Science Institute

• 수정완료일 : 2020. 11. 22

Email : hrkim@kasi.re.kr

• 게재확정일 : 2020. 12. 15

면 해당 관측별로 처리되어 관측자에게 제공된다.

처음 데이터가 기록되는 관측소에서 모든 데이터가 손실없이 기록되는 것이 아니라, 관측 결과에 영향을 미칠 정도는 아니지만 가끔씩 데이터가 기록되지 못하는 현상이 발생하여 그 원인을 분석하고자 하였다. 모든 과정이 순차적으로 순조롭게 이루어지더라도 데이터가 최종 기록되는 곳은 8개로 이루어진 하드디스크 어레이이며, 모든 데이터는 다시 8개로 분할되어 개별 하드디스크에 기록된다. 결국, 최말단에서 데이터를 기록해 주는 하드디스크의 성능을 자세히 알면 데이터 손실의 현상을 구체적으로 이해할 수 있으리라는 기대를 갖고, 하드디스크의 기록 성능을 분석해 볼 수 있는 유틸리티를 개발하기로 하였다.

본 논문에서는 개발된 유틸리티를 이용하여 실제 관측에 사용하는 하드디스크의 전체 속도 속성을 자세하게 분석해 볼 수 있었다. 유틸리티의 활용 범위를 확장하기 위하여 일반 대용량 저장매체, SSD 와 NVME에 대한 속도측정 실험을 수행하였다. 10Gbps 이상의 속도 특성을 갖고 있는 NVME를 이용하여 리눅스 cp와의 속도 비교 실험을 실시하여, 본 논문에서 개발된 유틸리티의 성능에 신뢰성도 있음을 보이고자 한다.

II. 데이터 I/O 속도 측정용 유틸리티 개발

2.1 데이터 I/O 기준 속도

하드디스크에 데이터를 쓰거나 읽는 것을 흔히 I/O로 표현한다. 이러한 I/O의 순간 속도를 편리하게 측정하기 위해서, I/O에 사용하는 데이터의 양 dQ을 일정한 양으로 고정하기로 하였다. 측정에 사용되는 데이터의 양이 고정됨으로써, 소요되는 시간 dt만 측정하면 되는 것이다.

KVN에서 사용하는 하드디스크의 I/O 속도가 대략 2Gbps 이하의 값을 가지기 때문에, 속도의 기준을 1Gbps(=1024 Mbps)로 정하였다. 1Gbps는 데이터의 양 128MB를 1초에 기록하는 속도이다. 따라서 1GB(=1024MB) 데이터 양을 8초에 처리했다고 하면 속도는 1Gbps에 해당되며, 만약 소요시간 1초라면 무려 8Gbps에 해당된다. 즉 1Gbps의 속도를 기준으로 삼아 소요시간 초를 Gbps로 단위 변환없이 숫자만

을 인식하여 편리하게 환산하기 위한 것이다. 이러한 기준을 바탕으로 데이터 I/O 함수는 128MB의 데이터를 처리하도록 되어 있고, 처리 전후에 소요되는 시간을 측정하여 속도를 계산하도록 되어 있다.

2.2 버퍼 크기 선택

보통 스토리지의 동작 환경은 데이터 I/O 함수를 사용할 때 버퍼 크기를 최적화하여 사용할 필요가 있다. 이러한 기능적 탐사를 위하여 버퍼 크기를 선택할 수 있도록 하였다. 버퍼 크기는 512B, 1KB, ..., 512KB, 1MB, ..., 512MB의 21가지 크기를 가질 수 있도록 하였다. 즉, 동일한 양의 데이터를 기록하면서, 사용하는 버퍼 크기를 변화시킴으로써 데이터 I/O 속도가 어떻게 달라지는지를 살펴볼 수 있다.

2.3 데이터 I/O 함수

데이터 I/O 함수는 일반 C/C++에서 제공하는 함수를 사용하였다[6,7]. 하드디스크에 데이터 I/O를 실행하는 방법은 일반 C++에서 제공하는 fwrite와 fread 함수를 이용하면 된다. 데이터 I/O는 버퍼 단위로 처리되며 구현에 사용된 구문은 다음과 같다.

```
fp.write ((char *)uch, this->bsize);
```

```
fp.read ((char *)uch, this->bsize);
```

지정된 버퍼를 한 번에 쓰게 되며, 횟수는 128MB/(버퍼 크기)가 된다. 만약 버퍼 크기를 128KB로 선택하였다면, 1024번 반복하여 동작하게 된다.

2.4 시계함수

시계함수는 <sys/time.h>에서 제공하는 gettimeofday 함수를 이용하였다. 구현된 시계함수는 다음과 같다.

```
struct timeval tv;
```

```
struct timezone tz;
```

```
gettimeofday (&tv, &tz);
```

```
esecond=(tv.tv_sec-tz.tz_minuteswest*60)+tv.tv_usec/1.0E6;
```

유틸리티에 사용하는 시간은 마이크로 초단위까지 저장하여 사용한다.

시계함수를 이용한 부가 기능으로써, 시계함수를 반복 호출하는 방법으로 사용하는 컴퓨터의 시간측정에 소요되는 시간을 알 수 있도록 하였다. 시계함수는

I/O 시작 전에 호출되어 시작 시간을 기억하고, I/O가 종료된 뒤 다시 호출되어 종료 시간을 기억한다. 프로그램에서는 두 시간의 차이를 소요시간으로 간주한다. 이러한 과정을 정밀하게 분석해 보면, 데이터 I/O에 사용되었다고 생각하는 소요시간에는 종료시 시계함수를 1번 호출하는데 사용된 시간이 포함되어 있다. 그러나 대부분 아주 작은 값이어서 무시되어 고려되지 않을 뿐이다. 본 연구 실험에 사용된 PC는 ASUS Maximus XII Hero 마더보드를 사용하였고 테비안 10.3.0 버전을 설치하였다[8]. 이 PC에서 시계함수가 초당 약 4,400 만번 호출됨을 알 수 있었고, 이것은 시계함수 1회 호출에 대략 23 nsec 정도 걸린다는 것을 말해 준다, 이 정도의 소요시간은 데이터 I/O에 소요되는 시간에 비하면 매우 작은 값으로서 충분히 무시될 수 있다고 여겨진다.

2.5 데이터 I/O 속도 출력

하드디스크의 I/O 속도는 데이터 양 dQ 를 기록하는데 걸린 시간 dt 를 이용하여 $v=dQ/dt$ 의 간단한 식으로 계산된다. 128MB의 데이터 양의 처리가 끝나면, 화면에 속도와 날짜 등의 정보를 출력한다. 출력되는 항목은 mode, bsize, filename, date, eseccond_1970, v, vepoch, dt, dN, tobs, N, Q 12개 항목이다. mode는 I/O가 쓰기인지 읽기인지를 나타내며 write와 read로 구분된다. bsize는 버퍼의 크기를 나타내며 기본값은 4K이다. 표기의 간편성을 위하여 마지막 단위인 바이트 표시 B를 생략하였다. 날짜는 '년-월-일 시:분:초'의 형식으로 표시되어 일상적인 시각으로 눈으로 쉽게 이해할 수 있도록 출력된다. eseccond_1970은 1970년 1월 1일 0시부터의 경과시간이다. 이 경과시간을 마이크로 초단위까지 표시하도록 하였다. v는 순간속도를 나타내며 출력되는 단위는 Mbps를 사용하였다. 1Gbps일 경우 '1024'로 표시된다. vepoch는 처음 시작부터 현재까지 처리한 데이터 양을 시간으로 나눈 전체 평균 속도값을 나타내 준다. dt는 128MB 데이터의 양을 처리하는데 소요된 시간이며 단위는 초를 사용하였다. dN은 128MB의 데이터 양을 처리하기 위하여 버퍼 단위로 몇 번을 동작하였는지를 나타내 주는 값이다. tobs는 측정을 시작하면서 현재까지 경과된 시간을 표시하며 초 단위를 사용하되 마이크로 초까지의 정확도를 갖도록 하였다. N은 현재까지 사용한

버퍼의 누적 횟수이다. 마지막으로 Q는 버퍼크기와 누적횟수로 계산된 데이터 양을 나타낸다. 크기가 증가함에 따라 M, G, T의 단위가 자동 사용된다. 출력의 편의를 위하여 바이트 단위 B가 생략된 형식을 갖는다.

2.6 tint 옵션

일반 하드디스크의 경우 최고 속도는 2Gbps 이하이다. 하드디스크의 속도 측정을 시작하면 초반에 대략 초당 2번씩 출력되다가 점차 속도가 느려져 초당 1회의 속도로 출력될 것이다. 일반 저장장치의 경우, 저장 장치의 I/O 속도가 대부분 8Gbps 이상이거나 수십 Gbps의 성능을 갖고 있다. 만약 저장장치의 I/O 속도가 10Gbps 라고 가정해 보았을 때, 이 저장장치의 데이터 I/O 속도를 측정하면 초당 10번씩 출력하게 된다. 화면에 출력되는 정보를 파일에 저장한다면 실험 후에 자세히 살펴볼 수도 있으나, 현장에서 눈으로 직접 살펴보기에는 너무 빠른 출력 횟수이다. 이러한 경우를 대비하여 tint 옵션을 사용할 수 있도록 하였다. 데이터 I/O 속도가 1Gbps보다 크다고 예상될 때, 대략 Gbps 단위의 값을 tint의 값으로 설정하게 되면 출력횟수가 줄어들어 대략 초당 1회 꼴로 보이게 된다. 예를 들면, SSD의 경우 대략 4Gbps라는 I/O 성능을 갖고 있다. 만약 기본값 tint=1을 설정하면 초당 4번의 출력이 되고 눈으로 읽기에는 무척 빠른 편이다. 이러한 상황에서 tint=4나 tint=8의 값을 설정해 줌으로써 1초당 또는 2초당 한 번 꼴로 출력되게 함으로써 쉽게 눈으로 확인할 수 있게 된다.

III. 저장장치별 데이터 I/O 속도 측정

3.1 하드디스크 속도 측정

개발된 유틸리티를 이용하여 하드디스크의 쓰기 성능을 측정하였다. 실험에 사용된 하드디스크는 도시바 제품을 사용하였고 용량은 8TB 급이다. 화면에 출력되는 정보를 파일에 저장한 후 gnuplot 유틸리티[9]를 이용하여 그림 1과 같이 만들었다.

실험에 사용된 컴퓨터의 구성은 메인보드는 ASUS ROG MAXIMUS XII HERO (인텔 Z490/ATX)에 인텔 코어 10세대 i7-10700 CPU를 사용하였고, 메모리

는 삼성 DDR4 16GB PC4-21300을 4개를 장착하여 총 64GB가 되도록 하였으며, 설치한 O/S는 테비안 10.3.0이다. 그림 1에서 볼 수 있듯이 도시바 8T 하드디스크의 I/O 속도는 1950 Mbps에서 시작하여 전반적으로 줄어들어 대략 980 Mbps의 속도를 갖는다. 다만, 하드디스크를 XFS 포맷을 사용하도록 하였기 때문에 끝부분에 가서 속도가 높아지는 부분이 약간 있다. 이것은 널리 알려진 EXT4 포맷에는 없는 XFS 포맷만의 특징이다.

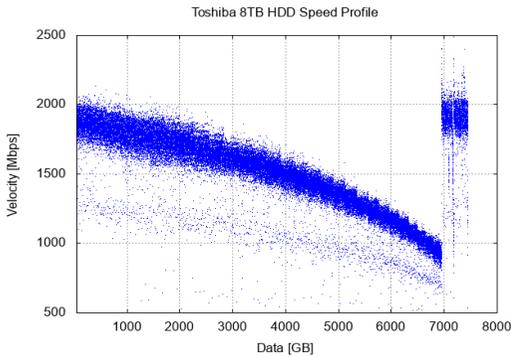


그림 1. 도시바 8TB 하드디스크 속도 곡선
Fig. 1. Speed profile of Toshiba 8TB HDD

도시바 하드디스크의 속도 특성을 알기 위하여 기본값을 사용하였다. 기본값일 경우 버퍼 크기는 4KB를 사용하고, tint=1을 사용한다. 그림 1은 이러한 측정 결과를 잘 보여주고 있다. 하드디스크는 실린더 구조라는 방식으로 데이터를 저장하며 바깥 쪽 실린더부터 사용하여 안쪽으로 이동한다. 하드디스크의 회전 속도는 일정하기 때문에 이 경우 rpm 7200-, 실험 초기의 속도에 비해 나중 속도가 점점 느려지는 것은 당연한 결과이다.

그림 1에서 보면, 안쪽 실린더의 경우 속도 곡선에 약간의 계단 모양이 있음을 알 수 있다. 이러한 현상을 더욱 자세히 볼 수 있도록 하기 위하여, 출력된 정보에 이동평균을 취하여 속도 정보를 그림 2와 같이 다시 그리는 방법을 사용하였다. 본 그림에서 보여주는 이동평균은 $Navg=128$ 이라는 값을 사용한 경우이다. 이것은 데이터 양은 $dQ=128MB*128$ 이라는 128배의 값을 갖게 되고, 소요시간은 과거 128번째부터의 시간을 적용하기 위하여 $dt=t[i]-t[i-128]$ 을 사용하였

다. 그림 2에서 볼 수 있듯이 도시바 8T 하드디스크의 실린더의 수가 33개로 이루어져 있음을 선명하게 확인할 수 있다.

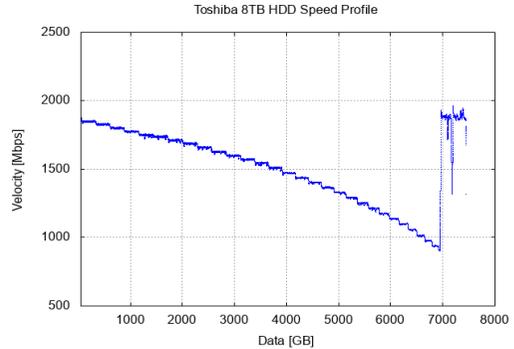


그림 2. 이동평균을 적용한 도시바 8T 하드디스크 속도 곡선.
Fig. 2. Speed profile of Toshiba 8T HDD, applied by the moving average method.

3.2 일반 대용량 저장장치 속도 측정

유틸리티의 적용 분야를 확대하기 위하여 대용량 저장장치의 I/O 속도를 측정하였다. 실험에 사용된 저장장치는 240TB 저장장치이며, 측정에 사용된 용량은 25TB 정도이다. 저장장치의 속도가 8Gbps 급으로 예측되어 tint=8로 조정되어 측정되었다. 측정 결과와 이동평균 $Navg=128$ 을 적용한 결과를 그림 3에 동시에 보여주고 있다.

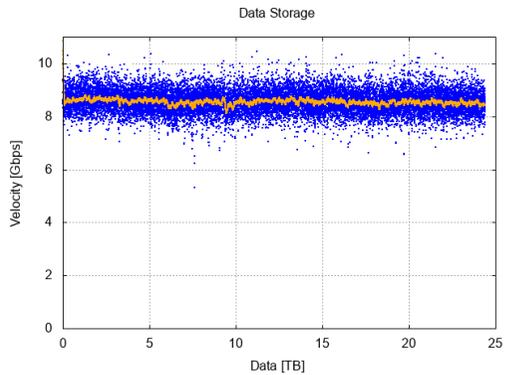


그림 3. 240TB 대용량 저장장치의 데이터 I/O 속도곡선.
Fig. 3. Speed profile of the large volume storage with 240TB

대부분 저장장치의 성능을 속도로만 볼 수도 있으나, 본 프로그램의 특성을 활용하면, 데이터 I/O 순간 속도의 분포를 그림 4와 같이 알 수 있다. 이 저장장치의 경우, 평균속도는 8Gbps 이상이다. 만약 8Gbps의 속도로 생성되는 데이터를 기록한다면 그림 4에서 보듯이 데이터 손실이 발생할 수 있다고 예측할 수 있다. 결국, 이 저장장치의 경우 6Gbps 급의 데이터를 저장하는 데는 비교적 안전하다고 권장할 수 있을 것이다.

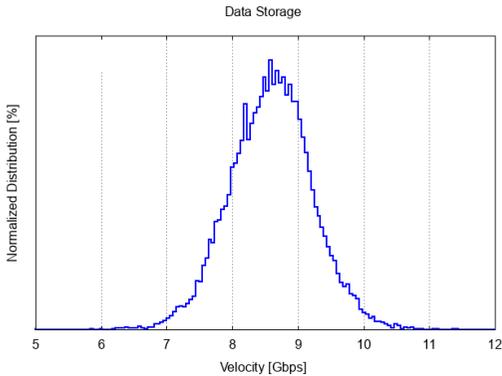


그림 4. 240TB 대용량 저장장치의 데이터 I/O 속도 분포도

Fig. 4. Speed distribution of the large volume storage with 240TB

3.3 SSD 속도 측정

다음으로 요즘 새로운 저장장치로 각광을 받고 있는 SSD의 I/O 속도를 측정하였다. 실험에 사용된 SSD는 삼성 860 EVO 1TB이며 생산지는 Product of KOREA이고, 인터페이스는 SATA3(6G)이다. 측정 옵션은 기본값을 사용하였으며, 이동평균 $N_{avg}=64$ 를 적용한 결과는 그림 5와 같다. SSD 1TB의 데이터 I/O 성능을 측정해 본 결과, 쓰기 속도는 4060 Mbps이며 이를 제품 표시 방식으로 바꾸어 환산하면 507 MB/sec가 된다. 읽기 속도는 쓰기보다 약간 낮은 4050 Mbps의 값을 보였다.

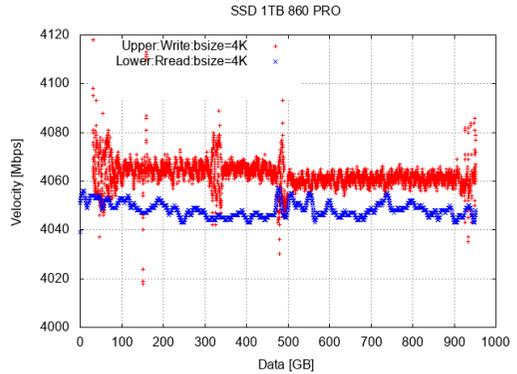


그림 5. SSD 1TB의 속도 곡선 (상단:쓰기, 하단:읽기).

Fig. 5. Speed profile of SSD 1TB (upper side:write, lower side: read)

3.4 10Gbps 이상 저장매체에 대한 적용

이제까지 본 연구에서 개발된 유틸리티로 측정된 장치의 속도는 대략 10Gbps 급의 수준이었다. 프로그램의 활용 범위의 확장을 위해서는 보다 빠른 장치에 대한 실험이 필요하였다. 이를 위하여 SSD 보다 더 빠른 저장장치로 각광 받고 있는 M.2 NVME 장치 [10]에 대한 측정을 수행하였다. 실험에 사용된 M.2 NVME는 삼성 NVME 970 PRO PLUS 2TB이며 생산지는 Product of KOREA이다. 속도 측정은 기본값을 사용하였으며, 측정 결과는 그림 6에 보였다. NVME 2TB에 대한 측정결과 쓰기 속도는 평균 10Gbps이고, 읽기 속도는 평균 18Gbps임을 알 수 있었다.

3.5 >10Gbps 매체에서 신뢰성 검증 실험

데이터 I/O 속도가 10Gbps 이상이 되는 NVME 2TB 장치에 대한 측정실험이 이루어짐에 따라 본 연구에서 개발된 유틸리티의 성능과 신뢰성을 검증해 보기 위하여 리눅스 O/S에서 제공하는 cp와 비교하는 실험을 하였다. 리눅스 O/S에서 제공하는 cp의 성능을 측정하기 위하여 파일을 /dev/null에 쓰는 방식을 이용하였다. 즉, 'cp <file> /dev/null' 과 같이 명령을 수행하고, 이에 대한 소요시간을 측정하였다. 시간 측정은 date 함수를 사용하여, 쉘 환경에서 'tnow='date +%s''와 같이 처리함으로써 시작 시간과 종료 시간을 저장한 다음 소요 시간을 계산하였다

[11]. 유틸리티의 파일 읽기 옵션을 선택하고 화면에 출력되는 것을 /dev/null로 보냄으로써, cp의 속도 측정과 유사한 환경이 되도록 실험을 진행하였다. 실험에 사용된 파일은 앞의 실험에 사용된 NVME 2TB에 약 960GB의 파일을 만들어 사용하였다. cp의 쓰기 성능은 측정하기 어렵기 때문에, 읽기 성능만을 측정하기로 하였으며, 총 실험시간은 약 20시간 정도 걸렸다. 그림 7에 비교 실험 결과를 보이고 있다.

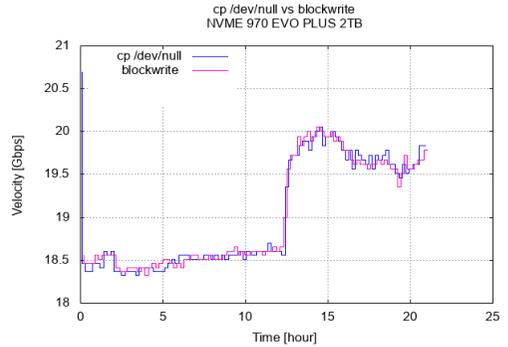


그림 7. cp와 유틸리티 read 기능의 읽기 속도 비교.

Fig. 7. Speed profile of cp vs read of utility.

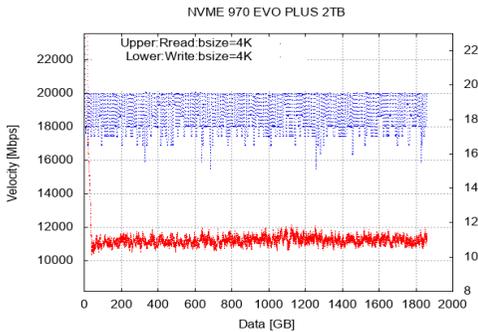


그림 6. NVME 2TB의 속도 곡선 (상단:읽기, 하단:쓰기). y축 단위(좌측:Mbps, 우측:Gbps).

Fig. 6. Speed profile of NVME 2TB (upper side:read, lower side:write).

그림 7의 측정 결과에서 보듯이 리눅스 cp의 읽기 속도와 개발된 유틸리티의 읽기 속도는 거의 동일함을 알 수 있다. 이것은 개발된 유틸리티의 신뢰성이 매우 높다는 것을 말해 준다. 실험을 진행하는 초기 13시간 동안에는 약 18.5 Gbps의 속도를 보이다가, 이후 속도가 19Gbps 이상으로 올라가는 현상이 발생하였다. 본 실험은 개발된 유틸리티가 10Gbps 급 이상의 장치에서 신뢰할 수 있는지를 살펴보는 것에 있으므로, 별다른 장치의 이상이 아닌 것으로 간주하여 이에 대한 자세한 논의는 본 연구 범위에서 제외하기로 한다.

V. 결론 및 향후개선 방향

본 연구의 목적은 KVN의 데이터 손실이 발생하는 현상을 이해하기 위하여, 최종 데이터 기록을 담당하는 하드디스크의 기록 속성을 자세히 이해하고자 출발하였다. 개발된 유틸리티가 작은 용량과 낮은 속도를 갖는 하드디스크에만 적용되는 것을 넘어, 일반적 저장매체와 대용량 저장장치에도 적용될 수 있음을 보였다. 10Gbps급 이상의 매체에 대한 신뢰성을 검증해 보기 위해 실시한 NVME에 대한 읽기 속도 비교 실험은, 20Gbps 이하의 매체에 대한 자세한 속도 곡선이 신뢰할 수 있다는 것을 보여 주었다. 이렇게 여러 저장 장치에 적용되어 얻어진 결과와 검증된 신뢰성을 바탕으로 일반적으로 어떠한 곳에서도 컴파일하면, 누구나 사용자 입장에서 저장장치의 쓰기와 읽기 성능과 속성을 자세히 측정해 볼 수 있게 되었다고 여겨진다.

앞으로도 대용량 저장장치의 성능 측정과 새로이 개발되어 보급되고 있는 각종 저장장치의 성능을 측정해 봄으로써, 저장장치 입출력 속도 측정과 이해에 큰 도움이 되리라 기대한다.

감사의 글

이 논문은 2020년 한국천문연구원 기관고유사업의 지원에 의한 논문임

References

- [1] Y.-W. Kang, M.-G. Song, S.-O. Wi, S.-M. Lee, and S.-R. Kim, "Development and Observation Result of High Speed Digital Conversion System of Astronomical Radio Signal," *J. of the Korea Institute of Electronic Communication Sciences*, vol 12, no. 6, Dec. 2017, pp. 1009-1018.
- [2] M.-G. Song, Y.-W. Kang, and H.-R. Kim, "The Study on the Design and Optimization of Storage for the Recording of High Speed Astronomical Data," *J. of the Korea Institute of Electronic Communication Sciences*, vol 12, no. 1, Feb. 2017, pp. 75-84.
- [3] M.-S. Lee and K.-J. Yoo, "Implementation of a Platform for the Big Scientific Data Trnsfers," *J. of the Korea Institute of Electronic Communication Sciences*, vol 13, no. 4, Aug. 2018, pp. 881-886.
- [4] M.-G. Song, H.-R. Kim, Y.-W. Kang, D.-H. Je, S.-O. Wi, and S.-M. Lee, "Implementation of Ring Buffer basd Massive VLBI Data Stream Input/Output over the Wide Area Network," *J. of the Korea Institute of Electronic Communication Sciences*, vol 14, no. 6, Dec. 2019, pp. 1109-1120.
- [5] M.-G. Song, H.-R. Kim, and Y.-W. Kang, "A Study on the Performance Monitoring and Optimization of a High Speed Network for the Transfer of Massive VLBI Data," *J. of the Korea Institute of Electronic Communication Sciences*, vol 14, no. 6, Dec. 2019, pp. 1097-1108.
- [6] U. Kirch-Prinz and P. Prinz, *A Complete Guide to Programming in C++*. Massachusetts: Jones and Bartlett Publishers, 2002.
- [7] Bjarne Stroustrup, *The C++ Programming Language (4th Edition)*, Addison-Wesley, 2013.
- [8] Thomas Down, *Installing Debian GNU/Linux*, Carmel Ind., Sams, 1999.
- [9] Philipp K. Janert, *Gnuplot in Action, Second Edition*. NY: Manning Publications Co., 2016.
- [10] J. Ahn, C. Hyun, and D. Lee, "Persistent Page Table and File System Journaling Scheme for NVM Storage," *J. inst. Korean. electr. electron. eng.* vol 23, no. 1, Mar. 2019, pp. 80-90.
- [11] Richard Siddaway, *Power Shell in Practive*, NY: Manning Publications Co., 2010.

저자 소개



김효령(Hyo-Ryoung Kim)

1990년 서울대학교 천문학과 졸업 (이학사)
 1996년 부산대학교 대학원 천문학과 졸업(이학석사)
 2003년 부산대학교 대학원 천문학과 졸업(이학박사)
 1990년 ~현재 한국천문연구원 연구원
 ※ 관심분야 : 전파천문, 외부은하, 클러스터



송민규(Gyu-Min Song)

2001년 강원대학교 전기공학과 졸업(공학사)
 2003년 강원대학교 대학원 전자공학과 졸업(공학석사)
 2002년 ~현재 한국천문연구원 연구원
 ※ 관심분야 : 대용량 데이터 처리, 초고속 네트워크, 병렬 시스템

