Regular paper

Journal of the KIECS. pp. 817-828, vol. 12, no. 5, Oct. 31 2017, t. 85, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2017.12.5.817

빅 스트림 기반 초고속 데이터 스토리지 성능 평가

송민규* · 강용우** · 김효령**

Performance Evaluation of Big Stream based High Speed Data Storage

Min-Gyu Song* · Yong-Woo Kang** · Hyo-Ryoung Kim**

요 약

단일 스트림 기준으로 10Gbps에 달하는 데이터 처리를 수행하는 시스템은 매우 제한적이고, 그에 관한 애플리케이션 역시 극소수에 불과한 실정이다. 하지만 이러한 시스템은 물리, 천문 등 과학연구 분야에서 더욱 그 중요성이 부각되고 있으며, 업그레이드 된 성능을 요구하고 있다. 이를 위해 해당 분야의 중소기업 지원을 위해 1년 전 10Gbps 수준으로 네트워크 입력되는 고속의 패킷 수신, 저장을 수행하는 초고속 스토리지가 개발되었다. 하지만 실제 연구 현장에서 사용되기 위해서는 객관화된 성능 평가가 수반되어야 하며 최적화가 이뤄져야 한다. 본 논문에서는 이에 필요한 시스템 환경을 구축한 것을 시작으로 객관화된 성능 지표를 얻기 위한 실험 과정 및 방법론에 대해 논의하였다.

ABSTRACT

It is very hard to find the system which processes single 10Gbps stream, and the related application is also rare. But in the field of science such as physics and astronomy, these high speed systems have been widely used and now more upgraded performance is expected. For this reason, high speed network based storage which captures and records 10Gbps level of packets was developed for the support of small astronomical company in KASI. But for the use of the system in research, system performance should be not only evaluated but also optimized. In this paper, we first implement system environment for the performance evaluation and discuss the experiment procedure and solution to acquire numerical results.

키워드

VDIF, Big Stream, Ring Buffer, Thread, Multicore CPU, RAID
VDIF, 빅 스트림, 링 버퍼, 쓰레드, 멀티코어 CPU, 레이드

Ⅰ. 서 론

IT 기술의 급격한 발전 및 활용 범위의 증가에 따라 네트워크 상에서 유입되는 초고속 데이터를 효과적으로 처리할 수 있는 스토리지의 비중 역시 높아지고 있다. 웹, SNS를 비롯 인터넷 상의 각종 IP 관련 서비스를 처리하는 클라우드 기반 일반 스토리지 분야의 경우 단위 노드 당 수 백 ~ 수 천명에 달하는 불특정 다수 사용자의 의해 발생하는 데이터를 처리한다[1]. 이 경우 각 사용자에 의해 발생하는 데이터

** 한국천문연구원 전파천문본부
(byulmaru@kasi.re.kr, hrkim@kasi.re.kr)
* 교신저자 : 한국천문연구원 전파천문본부
· 접 수 일 : 2017. 09. 30
· 수정완료일 : 2017. 10. 02
· 게재확정일 : 2017. 10. 18
· Received : Sep 30, 2017, Revised : Oct 02, 2017, Accepted : Oct 18, 2017
· Corresponding Author : Min-Gyu Song
Radio Astronomy Division, Korea Astronomy and Space Science Institute.
Email : mksong@kasi.re.kr

는 소규모이고 산발적이지만 방대한 사용자를 커버하는 특성으로 스토리지가 커버하는 전체 용량은 증가한다.

반면 과학 연구에서 사용되는 스토리지는 일반 IT 사용자의 그것과 다소 차별화된 특성을 필요로 한다. 이들 스토리지는 불특정다수의 사용자에게 통상적인 IT 서비스를 제공하는 것이 아닌 특정 분야 종사자의 연구 개발에 그 초점이 맞춰져 있다. 때문에 사용자 수가 극히 제한적이다. 하지만 각 개별 연구자에 의해 생산되는 데이터 용량이 단일 스트림 기준으로 수 ~ 수십 Gbps에 육박하기 때문에 일반 사용자의 데이터 발생량과는 비교 자체가 무의미하고 일반 스토리지로 처리함에 한계가 있다[2]. 이와 관련된 대표적인 연구 분야로는 HEP(: High Energy Physics : 고에너지 물리), VLBI(: Very Long Baseline Interferometry: 초장기선 전파간섭계)를 들 수 있다. 이 중에서 VLBI는 날로 늘어가는 대용량 데이터에 대한 효율적인 처리를 위해 지난 수 십 년 간 인터페이스, 데이터 포맷 측면에서 별도의 스토리지를 자체 개발해 사용해 왔다. 하지만 이더넷 기술이 기존 1GbE을 넘어 10GbE, 100GbE로 급속하게 발전하고 저변 확대가 신속히 이뤄짐에 따라 최근 시스템은 광 네트워크를 접목하는 방향으로 업그레이드되고 있으며 이에 기반한 데이터 포맷을 채택해 활용하는 추세이다[3].

네트워크 상에서 고속으로 유입되는 대용량 데이터를 효과적으로 처리함에 있어 주목받고 있는 기술로 Circular Buffer라고도 불리는 Ring Buffer가 있다[4]. 이는 VLBI 시스템 개발 분야에서도 적극 활용되고 있는데 4-5년 전부터 관측 데이터 처리를 위한 입출력 인터페이스로 10GbE로 사용됨에 따라 해당 기술의 적용 및 활용은 더욱 커질 것으로 전망된다. 기본적으로 링 버퍼는 외부로부터 유입되는 고속의 패킷을 메모리 상에 보다 효율적으로 관리, 저장하기 위한 기술이다. 따라서 링 버퍼를 통해 패킷 버퍼링이 원활히 이뤄진다고 해서 스토리지 종단에서의 데이터 무결성이 보장될 수는 없다. 왜냐하면 최종적으로 데이터가 저장되는 HDD, SSD 등 기록 매체는 메모리에 비해 아무리 우수하더라도 입출력 성능이 수 십 분의 1에 불과하기 때문이다[5].

이에 따라 본 논문에서는 링 버퍼 상에서의 버퍼링은 물론 스토리지의 물리적 종단에 해당하는 디스크 어레이에서 데이터가 손실없이 기록되었는지를 모니터링함으로써 스토리지의 유효성을 검증하고자 한다. 본론에서 세부적으로 논의하겠지만 이를 위한 유용한 수단으로 메모리 상에서 버퍼링되는 VDIF 스트림의 실시간 헤더 분석을 채택하였다.

본 논문의 구성은 다음과 같다. 먼저 본 서론에 이어 2장에서는 초고속 입력 데이터 버퍼링의 핵심적 역할을 하는 링 버퍼와 그에 기반한 스토리지에 대해 간단히 살펴보고자 한다. 3장과 4장에서는 우리가 목표로 하는 초고속 데이터 스토리지의 성능 요건 및 성능 실험을 위한 실험 환경에 대해 기술하고자 한다. 그리고 5장에서 시스템 평가 수행 및 그 결과에 대해 고찰할 것이고 6장에서 결론을 맺고자 한다.

II. 링 버퍼 기반 패킷 처리

컴퓨터 시스템 내부의 데이터 입출력 과정에서 메모리는 일시적으로 임의의 데이터를 보관하는 역할을 한다. 다음 단계 또는 최종 목적지로 데이터 전달함에 있어 해당 데이터는 메모리에서 누적 또는 감소 과정을 거치게 된다. 이를 통해 사용자는 송수신 시스템 간 보다 유연하고 신뢰성 있는 데이터 전달을 구현할 수 있다. 시스템 내의 메모리는 물리적으로 시작과 끝이 존재하고 수용 가능한 용량에 있어서도 제한적이다. 이러한 상황에서 대다수 일반 컴퓨터 사용자의 경우 메모리의 상당 부분은 프로그램 프로세스 관리와 인터넷 트래픽 처리용으로 사용된다. 데이터 전송이 이뤄지더라도 용량과 속도면에서 제한적이고 수 시간 이상 지속적이지 않기에 메모리 용량이 작더라도 큰 문제가 되지는 않는다. 하지만 Gbps 급의 초고속 데이터가 최소 수 시간 이상 지속적으로 입력되는 경우에도 동일한 시스템 환경, 메커니즘을 적용한다면 어떻게 될까? 기존 방식 하에서 한정된 메모리는 점유율이 급속도로 상승하여 몇 초도 안 되어 시스템 오버플로우가 발생할 것이다. 그리고 이는 시스템 셧다운으로 이어질 가능성이 크다.

링 버퍼는 이와 같이 초고속 데이터가 대량으로 연속 유입된다고 가정할 경우 메모리 상에서 해당 데이터를 보다 효과적으로 처리하기 위해 제시된 개념이다[5]. 링 버퍼라는 이름에서 알 수 있듯이 메모리의

시작과 끝은 논리적으로 서로 연결되어 메모리 종단에 다다르면 자연스럽게 시작 위치로 반환된다. 이에 따라 대용량 데이터가 연속적으로 유입되더라도 한정된 용량의 메모리에서 보다 효율적인 처리를 구현할 수 있다. 다만 이에 대한 사용을 위해서는 한 가지 전제조건이 충족되어야 한다. 그것은 메모리 상에서 현재 수신되는 데이터가 이전 데이터를 덮어쓰지 않도록 디스크 블록 레벨에서 해당 데이터의 처리가 보장되어야 한다는 것이다. 이를 위해 링 버퍼를 경유한 로컬 상의 파일 저장 또는 원격지로 데이터를 전송하는 경우 쓰기 및 읽기 주소를 통한 메모리 관리가 이뤄진다. 한 예로 외부로부터 고속의 패킷이 유입되어 링 버퍼에 누적되면 쓰기 주소는 그에 비례하여 전진한다. 마찬가지로 읽기 주소 역시 링 버퍼 상의 패킷을 디스크 어레이로 파일 이동시키는 과정에서 순차적으로 증가하며 두 주소 간의 데이터 처리를 통해 오버플로우가 발생하지 않도록 관리한다,

본 논문에서 성능 평가가하고자 하는 초고속 스토리지는 기본적으로 링 버퍼의 이러한 기본 메커니즘에 기반하여 작동한다[6]. 또한 보다 효율적인 데이터 처리와 명확한 작업 분배를 위해 쓰레드를 활용하였다.

III. 초고속 데이터 저장 시스템

초고속 데이터 스토리지는 단일 스트림 기준 10Gbps급으로 입력되는 대용량 데이터에 대한 무손실 저장을 목표로 한다. 해당 시스템의 구성 및 목표로 하는 성능에 대해 간단히 기술하고자 한다.

3.1 시스템 구성

개발 완료된 초고속 데이터 스토리지는 하드웨어 측면에서 그 기능에 따라 크게 16개의 1TB HDD와 2개의 10GbE NIC, 1개의 RAID 카드로 분류된다. 하지만 데이터 수신이 실질적으로 이뤄지는 네트워크 인터페이스는 멀티모드 광 패치케이블을 경유해 데이터 소스와 직결되는 10GbE 1포트이다. 10Gbps 급 대용량 패킷을 손실없이 처리하기 위해서는 네트워크 입력 단에서 해당 패킷을 수신 처리하는 링 버퍼는 물론 최종적으로 데이터가 저장되는 디스크 어레이의 성능 역시 고도화되어야 한다. 이러한 성능 이슈를 감안해 네트워크 인터페이스는 는 인텔 x520 칩셋 기반으로 구성하였고 RAID0 기반으로 16TB 용량의 가상 디스크를 형상화하였다. 패킷 입출력이 실질적으로 이뤄지는 메모리의 용량은 크면 클수록 좋다. 용량에 비례해서 보다 많은 양의 패킷을 긴 시간 동안 버퍼링할 수 있고 최적화를 위한 기타 용도로 할당할 수 있기 때문이다. 본 논문에서 성능 측정할 스토리지에는 4개의 16GB DDR3 램이 탑재되었고 NUMA(Non-Uniform Memory Access) 모드 적용을 통해 최종 25GB 크기의 링 버퍼를 생성하였다. 10Gbps 급 패킷을 처리함에 있어 해당 용량이 부족하다고 느껴질 수도 있다. 하지만 실험 결과 실제 패킷 처리에 전혀 모자람이 없었으며 이에 대해서는 성능 평가가 이뤄지는 5 장에서 보다 상세히 논의하고자 한다.

링 버퍼는 오픈소스 기반으로 여러 가지 패키지가 소개되어 사용되고 있지만 그 중 PF_RING이 전세계적으로 가장 널리 사용되고 두터운 사용자 층을 형성하고 있다. 리눅스 배포판에서도 PF_RING이 패키지 형태로 기본 제공되고 있는 점을 감안하여 이에 기반으로 시스템을 구현하였다. 성능 극대화를 위한 설정치를 정리하면 표 1과 같다.

표 1. PF_RING 설치 내역 및 설정 옵션
Table 1. Installed PF_RING's version and other configuration options

Items	Values
PF_RING Version	6.7.0
Ring slots	8192
Capture TX	No [RX only]
IP Defragment	No
Socket Mode	Quick

3.2 성능 요건

데이터 생산 및 소비 증가에 따라 IT 시장에서 스토리지의 비중은 해마다 증가해 왔다. 하지만 시장의 수요는 이미 오래전부터 보다 많은 사용자에게 다양한 네트워크 서비스를 제공하는 범용 형태로 발전해 왔다. 이러한 추세는 향후 클라우드 컴퓨팅 환경에서

더욱 심화될 것으로 예측된다. 실제로 델, HP, IBM 등 메이저 업체에서 출시되는 제품의 상당수는 클라우드 기반에 초점이 맞춰져 있다[7].

하지만 물리, 천문, 기상, 생명 등 다수의 과학연구에서 사용되는 스토리지는 해당 분야의 특성상 결코 많은 사용자가 필요치 않다. 대신, 각 사용자가 하루 발생시키는 데이터 양은 수 ~ 수 십 TB 수준으로 일반 사용자의 그것과 차원이 다른 상반된 특성이 있다. 이러한 방대한 데이터는 일반적으로 네트워크 상의 원격지로부터 전송되기에 최종 데이터가 저장되는 스토리지는 빅 스트림에 최적화되어 있어야 한다[6].

링 버퍼를 사용할 경우 NIC를 통해 입력되는 데이터는 보다 효율적으로 대용량 내부 메모리에 의해 관리되며 데이터 처리에 있어 이점을 제공한다. 하지만 일반 소켓 통신과 비교 시 소켓 기술자에 의한 소켓 개설이 없고 TCP 스트림이라 할지라도 재전송에 의한 오류제어, 흐름 제어를 거치지 않는다. 이는 우리가 개발한 스토리지가 링 버퍼 상에서 데이터 입출력 성능 극대화는 물론 디스크 어레이에서의 데이터 손실 최소화 요건을 동시에 만족시켜야 함을 의미한다. 본 논문에서 추구하는 스토리지의 성능은 10Gbps 수준에서 10분 단위로 시스템 운용 시 네트워크 패킷 손실을 0.05% 이하로 낮추는 것이다. 데이터 저장에 있어서는 단 1바이트의 누락도 발생해서는 안 되며 해당 실험 진행 및 결과에 대해서는 다음 장에서 세부적으로 논하기로 한다.

Ⅳ. 평가 방법 및 테스트베드 구축

기 개발된 스토리지의 성능을 정량적으로 평가하고자 한다. 다양한 사용 환경에서 시스템이 어떠한 성능 변화 추이를 보이는지 확인하기 위해 평가 조건을 달리 하였고 이를 위한 실험 환경을 구축하였다.

4.1 목적

본 논문에서 성능 평가를 진행하고자 하는 스토리지는 초고속 데이터 처리를 위해 개발되었다. 방대한 데이터를 기록, 저장하고 기초 과학용으로 사용되기 위해서는 무엇보다 데이터 품질이 보장되어야 한다. 더욱이 HEP, VLBI 등 우리가 스토리지 활용을 목표로 하는 분야는 방대한 양의 데이터가 초고속으로 유입되며 사용 빈도가 월등하다. 따라서 실전에서 사용되기 위해서는 오차가 최소화되어야 하고 이것이 수치적으로 제시되어야 할 것이다.

4.2 평가 항목

평가 지표로는 데이터 유실, 프린지 획득 2가지 방식을 선택하였다. 프린지는 임의의 신호에 대해 FFT(Fast Fourier Transform) 처리를 수행한 결과 얻어지는 특정 주파수 신호로서 이에 대한 검출을 통해 데이터가 정상 여부를 진단할 수 있다. 하지만 기록 데이터에 다소 결함이 있더라도 프린지가 검출될 가능성이 있기 때문에 이를 보완할 보다 세부적인 검증이 필요하다[8]. 이에 따라 본 논문에서는 먼저 송신 시스템에서 초고속 스토리지로 전송된 패킷 헤더 검출, 메모리 상에서 버퍼링된 스트림과 기록된 데이터의 비교 분석을 통해 데이터 유실 여부를 검증하고자 한다.

4.3 테스트베드 구축

다양한 실험 장비로 구성되는 테스트베드는 그 기능에 따라 데이터 생성, 전송, 저장 시스템으로 분류 가능하다. 각 부분에 해당하는 시스템에는 샘플러, 이더넷 프레임 소스, 컴퓨터 시스템이 있는데 그림 1에서 구축 완료된 테스트베드를 확인할 수 있다.

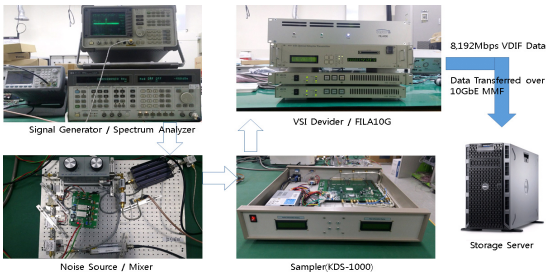


그림 1. 초고속 데이터 스토리지 성능 평가를 위한 테스트베드
Fig. 1 Testbed for performance evaluation of high speed data storage

위 그림을 통해 데이터 신호 생성부터 저장에 이르기까지 대략적인 흐름을 파악할 수 있다. 먼저 신호발생기에서 출력되는 200MHz 기준 주파수는 노이즈 소스와 믹서에서 결합되어 1,024Msps 샘플링, 2Bit 양자

화, 64MHz 클럭으로 동작하는 샘플러를 통해 2,048Mbps의 디지털 신호로 2포트 출력된다. 해당 두 신호는 복제, 출력 기능을 수행하는 두 대의 VSI 디바이더로 전달되어 4개의 2,048Mbps VSI 신호가 만들어지고 이는 다시 FILA10G에 입력되어 최종적으로 8,192Mbps 스트림으로 출력된다. 테스트 베드를 구성하는 각 장치의 간략한 사양 및 인터페이스는 표 2을 통해 확인할 수 있다.

표 2. 성능 평가 테스트베드를 구성하는 인스트루루먼트의 규격 및 기본 기능
Table 2. The Specifications & basic features of the instruments which implement the testbed for performance evaluation

Instruments	Specifications	Functions
Signal Generator	HP 8665A 0.1~4200MHz	Generation of 200MHz / -40dbm reference frequency
Waveform Generator	Keysight 33600A Frequency: 1Hz, Amplitude: 2.5Vpp, Offset: 1.25V	Generation of 1PPS tone signal
Spectrum Analyzer	HP 8563A 9kHz ~ 26.5GHz	Analysis for synthesis of reference frequency with noise
Sampler	KDS-1000 1GSps/2bit/64MHz	Conversion output signals from analog to digital type
VSI Devider	Tokyo Elecs VSI Interface	Duplication of a VSI signal and output the same two channel of data
VSI to VDIF Converter & Transmitter	FILA10G XFP + 2Ports, Xilinx Platform Cable USBⅡ	Conversion from VSI to VDIF and transmit 10GbE based VDIF frame
Server	Dell T630 2CPU, Intel Zeon E5-2623 v3 3.0GHz, 10M Cache, SFP+	Receiving of 10GbE based VDIF frame and save it as a file

Ⅴ. 성능 측정 및 결과 분석

패킷 유실, 프린지 획득을 위한 세부 실험을 통해 스토리지 성능을 정량적으로 진단하고자 한다. 보다 자세하고 즉각적인 결과를 얻을 수 있는 패킷 유실 점검에 대해 먼저 기술하기로 한다.

5.1 패킷 유실

본 논문은 초고속 스토리지 구현 단계에서 PF_RING을 주 라이브러리로 활용하였다. PF_RING은 네트워크 상에서 유입되는 프레임을 보다 효과적으로 수신 및 메모리에 저장하는 다양한 함수를 지원하며 주요 함수로는 pfring_open, pfring_get_bound_device_address, pfring_recv가 있다. 이 중에서 pfring_recv 함수는 실행 후 세 가지 경우에 따라 다르게 반응한다. 즉, 아무 패킷도 입력되지 않으면 0, 패킷이 정상적으로 수신되면 1, 수신 실패 시 -1을 반환하며 함수 실행 단계에서 반환값을 통해 패킷 누락 발생 여부를 파악할 수 있다[9].

메모리상에서 버퍼링되는 스트림의 누적 용량은 수신 패킷의 총합을 패킷 크기로 곱한 결과에 해당한다. 이 수치가 실제 스토리지에 저장된 데이터 용량과 일치하는지 비교함으로써 스토리지 상의 데이터 기록 과정에서 데이터 유실이 발생했는지를 파악하는 것이 가능하다. 보다 세부적으로 일정한 시간 간격마다 어떠한 프레임이 얼마나 많이 유실되었는지 알아내기 위해서는 각 패킷에 관한 추가 정보가 필요하다. 수신된 패킷마다 순차적으로 증가하는 수치가 헤더에 포함된다면 이를 구현하는 것이 보다 용이할 것이다. 하지만 일반 TCP/IP 패킷에서는 이에 부합되는 헤더 정보가 없다[10]. 다행히 현재 KVN이 사용하고 있는 VDIF 데이터에는 매 프레임마다 순차적으로 증가되고, 매 초마다 0으로 초기화되는 프레임 넘버가 포함된다. VDIF는 전파천문학에서 VLBI 데이터 처리를 위해 제정된 포맷으로 각 프레임마다 32바이트 크기의 헤더가 정의되어 있다. 이에 대한 헤더 구조를 간략히 도시하면 그림 2와 같다[3].

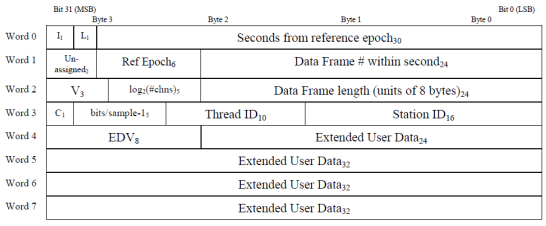


그림 2. 고속의 데이터 처리를 위한 VDIF 프레임의 헤더 구조
Fig. 2 Header structure of VDIF frame for high speed data processing

4바이트 헤더를 포함해서 VDIF 데이터는 8,224Mbps 속도(헤더 제외하고 실질적으로 VDIF 페이로드만 부각시켜 8,192Mbps라고 논하기도 함)로 FILA10G에서 스토리지로 전달된다. 이를 매 초 당 전송되는 VDIF 페이로드 용량으로 환산하면 1,024MB이고 다시 단위 페이로드 크기로 나누면 1초 당 전송되는 패킷 개수 125,000개를 얻을 수 있다. 즉, 0부터 124,999까지의 프레임 넘버를 갖는 VDIF 패킷이 스토리지의 네트워크 인터페이스로 입력되며 그 과정에서 각 패킷의 헤더를 분석함으로서 유실 여부를 파악할 수 있는 것이다. 이와 관련된 코드 일부를 나타내면 아래와 같다.

```
// 프레임 포맷이 VDIF에 부합되는지 검사
if (glob.stream.format == VDIF)
 {
    glob.stream.t_vdif = pvdh->seconds;
    printf("pvdh->seconds: %d, pvdh->frame: %d₩n",
pvdh->seconds, pvdh->frame);

// 프레임 개수 증가
    fnum++;
// 헤더 내의 시각 정보가 변경되면 누적된 프레임 개수 출력 및
초기화
    if (p_time != pvdh->seconds)
    {
        if (p_time == 0)
        {
            printf("Let's Start Performance Test₩n");
        }
        else
        {
            printf("Time: %d, Total:
%d/125000₩n",(pvdh->seconds - 1), fnum - 1);
            fnum = 1;
        }
        p_time = pvdh->seconds;
    }
}
```

성능 실험은 스토리지 말단에 해당하는 디스크 어레이까지 단번에 수행하기 보다는 데이터 소스로부터 어레이 측으로 단계적으로 범위를 넓혀가며 진단하는 방식을 준수하였다. 우선 데이터 발생 장치에 해당하는 FILA10G와 10GbE의 성능 프로파일을 얻을 필요가 있다. FILA10G로부터 한 치의 오차없이 패킷 전달이 이뤄진다면 맞은편 종단에 해당하는 스토리지에서 단 하나의 패킷 유실도 발생하지 않을 것이다. 하지만 소스 자체에 다소의 손실이 존재하고 광 네트워크 상의 송수신이 원활하지 않다면 아무리 스토리지의 성능이 안정적이라도 누락되는 패킷이 발생할 것이다. 이점에 착안해서 FILA10G와 스토리지의 네트워크 인터페이스 간 패킷 송수신 특성을 모니터링하였고 해당 프로파일을 그림 3에 도시하였다.

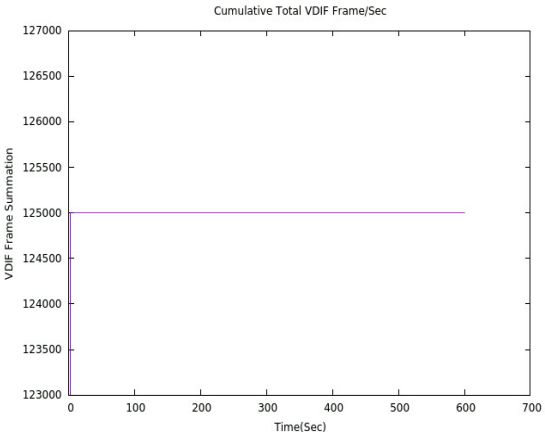


그림 3. FILA10G와 스토리지의 네트워크 패킷 송수신 특성 프로파일
Fig. 3 The Profile of packet transfer characteristic between FILA10G and high speed data storage

위 결과는 10분 동안 이뤄진 패킷 송수신에 관한 것으로 성능이 정상궤도에 진입하는 초기 2초 간 누락이 발생한 것을 제외하면 125,000개의 패킷이 매 초마다 안정적으로 네트워크 입력되었음을 알 수 있다. 네트워크 인터페이스를 지나 메모리까지 측정 범위를 확대해도 이러한 특성이 재현되는 것을 확인하였다. 메모리까지 고속의 패킷 입력 시 메모리까지의 성능 프로파일을 얻기 위해서는 이를 관장하는 CPU에서 디스크 어레이가 배제되어야 한다. 이에 따라 우리는 다음과 같이 디스크 어레이 기능을 담당하는 쓰레드를 일시적으로 비활성화시켰다.

```
#define RAID_ARRAY_OPERATING
#ifndef RAID_ARRAY_OPERATING
    pthread_create(&pthread, NULL, rb_read, (void *) NULL);
#endif
```

상기 조치를 통해 링 버퍼 기반 패킷 쓰기만 활성화되었고 이전에 비해 CPU에 가해지는 부하가 다소 늘어났다. 이에 따라 패킷 누락 증가로 이어질 가능성 역시 소폭 상향되었는데 그림 4에 이에 대한 패킷 송수신 실험 결과를 도시하였다.

Time	Frame_Num/Sec		Total_Frame
Buf_Stream_Size	Data_Size	Packet_Loss	
10025185	4094/125000		4094
33669056	0	120906	
10025186	28514/125000		32608
268168192	0	96486	
10025513	123892/125000		40906500
336415056000	0	1108	

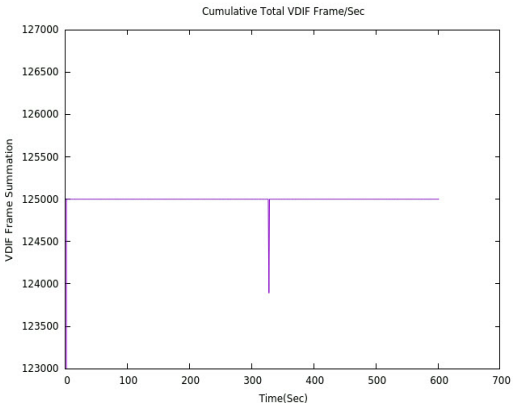


그림 4. 링 버퍼의 쓰기 기능만 활성화되었을 경우 패킷 유실 측정 결과
Fig. 4 The result of packet loss measurement in case of only ring buffer's write function enabled

먼저 그림 4 상단에 위치한 시각 및 패킷 누적 정보는 매 초당 125,000개의 패킷이 수신되지 않운 경우를 필터링 한 것이다. 602초 동안 총 3번에 걸쳐 패킷 유실이 발생했다는 것을 알 수 있다. 그 아래에서는 이를 그래프로 나타낸 것으로서 그림 3에서 확인한 FILA10G의 패킷 전송 특성과 별반 차이가 없고 스토리지 내의 메모리에서 스트림 버퍼링이 안정적으로 이뤄지고 있음을 알 수 있다.

메모리의 성능은 일반 저장 매체인 하드 디스크, SSD에 비해 수 십배 이상 뛰어나다. 따라서 메모리단에서의 성능은 안정적이라도 디스크 어레이 상의 데이터 기록 과정에서 패킷 누락이 발생할 수 있다. 이에 대한 확인 및 문제 해결에 있어 메모리에 입력되어 버퍼링 되는 스트림과 디스크 어레이에 저장되는 데이터 용량의 크기 비교는 매우 유용하다. 왜냐하면 스토리지 말단에서 데이터 손실이 없다면 메모리에서 입출력된 스트림 용량과 스토리지 상의 파일 용량은 같아야 하기 때문이다. 이에 대한 실험을 진행하기에 앞서 하드 디스크 기반 어레이의 경우 기록 전반부 성능이 후반부에 비해 높다는 사실이 고려되어져야 한다. 하드 디스크의 경우 데이터는 플래터라 불리는 여러 장의 원형 매체에 기록되는데 각속도 차이에 따라 안쪽으로 진입할수록 느려져 성능이 저하된다[11]. 따라서 스토리지 전체를 사용하기 위해서는 후반부에 해당하는 디스크 안쪽에서도 안정적으로 데이터가 기록되어야 한다. 이에 따라 본 논문에서는 그림 5에서 나타난 바와 같이 잔여 스토리지 용량 12%를 남겨둔 상태에서 데이터 기록을 실시하였고 해당 결과를 그림 6에 나타내었다.

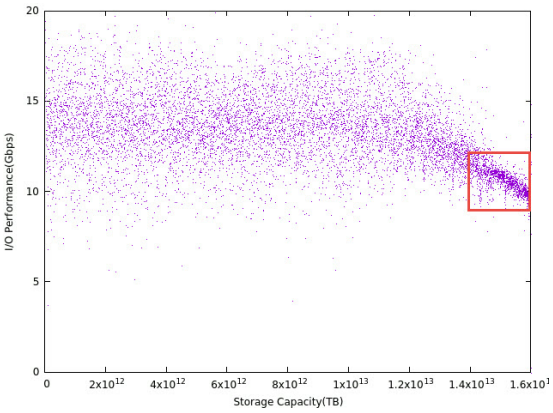


그림 5. 성능 평가를 위해 초고속 데이터가 기록된 디스크 어레이 말단 (빨간색 사각형으로 표시)
Fig. 5 The last part of disk array on which high speed data was written for the performance evaluation(displayed with a red rectangle)

Time	Frame_Num/Sec	Total_Frame	Buf_Stream_Size	Data_Size	Packet_Loss
10030651	8190/125000	8190	67354560	9992160	116810
10030652	44352/125000	52542	432105408	419670720	80648
10030653	125000/125000	177542	1460105408	1448863200	0
10030654	125000/125000	302542	2488105408	2478055680	0
10030655	125000/125000	427542	3516105408	3507248160	0
~~					
10030674	125000/125000	2802542	23048105408	23031928800	0
10030675	125000/125000	2927542	24076105408	24061121280	0
10030676	123635/125000	3051177	25092879648	25080321600	1365
10030677	125000/125000	3176177	26120879648	26109514080	0
10030678	125000/125000	3301177	27148879648	27138706560	0
~~					
10030738	125000/125000	10801177	88828879648	88820310240	0
10030739	125000/125000	10926177	89856879648	89829518400	0
10030740	121642/125000	11047819	90857263456	90848718720	3358
10030741	125000/125000	11172819	91885263456	91867919040	0
10030742	125000/125000	11297819	92913263456	92897111520	0
~~					
10031249	125000/125000	74672819	614109263456	614098169280	0
10031250	125000/125000	74797819	615137263456	615127361760	0
10031251	125000/125000	74922819	616165263456	616146562080	0
10031252	125000/125000	75047819	617193263456	617175754560	0

```
=================== SUMMARY ===================
        Stream Good Packet:75047819
        Buffered Stream Size: 617193263456
        Saved File Name: wrtest_fila10g_2017y298d02h17m30s.vdif
        File Capacity: 617193263456
================================================
```

그림 6. 초고속 데이터 기록 과정에서 실시간 패킷 캡쳐/ 기록 모니터링 및 요약된 결과 출력
Fig. 6 Real time monitoring during high speed packet capture & recording on our system and summary of the results

그림 6에서 표시된 숫자는 좌측 칼럼부터 각각 프레임 시간, 초당 프레임 개수, 누적 프레임 개수, 스트림 용량, 파일 용량 그리고 매 초당 유실된 패킷 개수를 실시간으로 보여준다. 프레임 시간은 해당 프레임이 생성 및 전송된 시간을 의미하는 것으로 본 실험에서는 10,030,651초부터 10,031,253초까지 총 602초 동안 이뤄진 패킷 수신 및 데이터 저장 결과를 포괄하고 있다. 측정 시간을 600초로 끊지 않고 602초로 잡은 이유는 FILA10G로부터 패킷이 유입되어 시스템이 본격적으로 가동하는데 최소 2초 정도의 시간이 필요하기 때문이다. 이는 본 실험에서도 확인되고 있는데 전체 패킷 유실 개수 202,181개 중 프로그램이 가동되기 시작한 초반 2초 동안 97.7%에 해당하는 197,458개의 패킷 유실이 집중되었다. 이후 시스템은 빠른 속도로 안정 성능에 진입하게 되어 실험 종료가 이뤄진 10여 분 동안 추가로 발생한 패킷 유실 건수와 개수는 각각 2건, 4,723개로 나타났다. 단 하나의 패킷 유실도 발생하지 않는다고 가정할 경우 수신 가능한 패킷 개수는 75,250,000개이다. 여기에 각 패킷의 크기 8224바이트를 곱하게 되면 이상적인 스트림/데이터 용량, 618,856,000,000 바이트를 얻을 수 있다. 하지만 우리의 경우는 602초 동안 4,723개의 VDIF 프레임 누락이 발생한 사실이 적용됨에 따라 초고속 스토리지에 입력되어 처리된 패킷의 개수는 다음과 같아야 한다.

실험 시간(초) x 단위 시간 패킷 전송 개수 - 패킷 유실 개수 = 수신된 총 패킷 개수

602 x 125,000 - 202,181 = 75,047,819

스토리지의 저장 성능을 검증하는데 있어 수신된 총 패킷 개수는 숙지할 필요가 있다. 왜냐하면 메모리로부터 스트림을 전송받아 스토리지 말단에 데이터를 저장하는 기능이 완벽하다면 실제 저장된 파일의 크기가 메모리 상에서 버퍼링된 스트림의 용량과 항상 일치해야 마땅하기 때문이다. 그 일치 여부는 두 가지

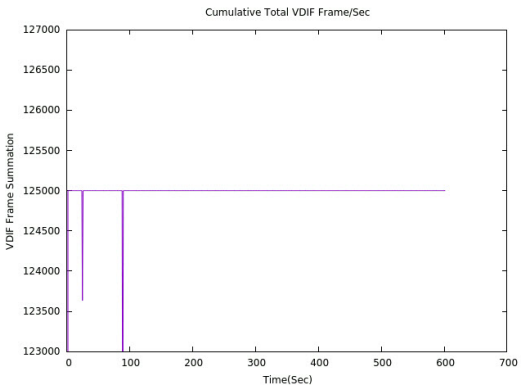


그림 7. 링 버퍼의 쓰기/읽기 기능 활성화되었을 경우 패킷 유실 측정 결과

Fig. 7 The result of packet loss measurement in case of ring buffer's write/read function enabled

지표에 대한 모니터링을 통해 확인이 가능하다. 먼저 그림 6의 하단부를 보면 실험 완료 후 처리된 패킷/데이터에 관한 정보를 볼 수 있다. 스트림 용량과 데이터 용량 두 수치가 617,193,263,456 바이트로 일치하는데, 이로부터 메모리 상에서 버퍼링된 스트림 데이터가 단 1바이트의 손실없이 디스크 어레이에 파일로 저장되었다는 것을 확인할 수 있다.

```
VDIF Time: 10031237 Total Frame: 73091844 Stream Capacity: 601107325056 File Name: wrtest_fila10g_2017y298d02h17m30s.vdif File Capacity: 601098369120

VDIF Time: 10031241 Total Frame: 73592158 Stream Capacity: 605221907392 File Name: wrtest_fila10g_2017y298d02h17m30s.vdif File Capacity: 605205146880

VDIF Time: 10031245 Total Frame: 74092468 Stream Capacity: 609336456832 File Name: wrtest_fila10g_2017y298d02h17m30s.vdif File Capacity: 609321916800

VDIF Time: 10031249 Total Frame: 74592776 Stream Capacity: 613450989824 File Name: wrtest_fila10g_2017y298d02h17m30s.vdif File Capacity: 613438686720

VDIF Time: 10031253 Total Frame: 75047819 Stream Capacity: 617193263456 File Name: wrtest_fila10g_2017y298d02h17m30s.vdif File Capacity: 617193263456

VDIF Time: 10031253 Total Frame: 75047819 Stream Capacity: 617193263456 File Name: wrtest_fila10g_2017y298d02h17m30s.vdif File Capacity: 617193263456
```

그림 8. 클라이언트에 실시간 제공되는 데이터 처리 정보

Fig. 8 Data processing information provided to client in real time

그림 8은 스토리지 상에서 패킷 수신/ 데이터 저장이 발생할 경우 클라이언트에서 구동되는 파이썬 스크립트의 실행 결과이다. 이는 3~4초마다 수신된 패킷 넘버 및 개수, 누적 패킷 개수, 스트림 용량, 그리고 마지막으로 파일 이름과 용량 정보를 출력한다. 스트림, 데이터 용량이 617,193,263,456 바이트로 동일하고 실제 저장된 파일 크기와도 일치하는 것을 통해 메모리 상의 스트림이 파일로 완벽하게 저장되었음을 알 수 있다.

프린지 검출을 위해서는 스토리지에 저장된 데이터를 mark5access라는 패키지를 이용해 FFT처리 하였다[12]. 아래 그림은 그에 대한 결과로서 동일한 패턴을 갖는 4개의 출력이(200MHz 주파수와 노이즈의 합성신호)가 연속적으로 검출되었음을 보여주고 있다. 앞서 이뤄진 패킷 유실 진단과 본 프린지 검출을 통해 개발된 스토리지가 실제 현장에서 사용될 수 있다는 것이 객관적으로 검증되었다.

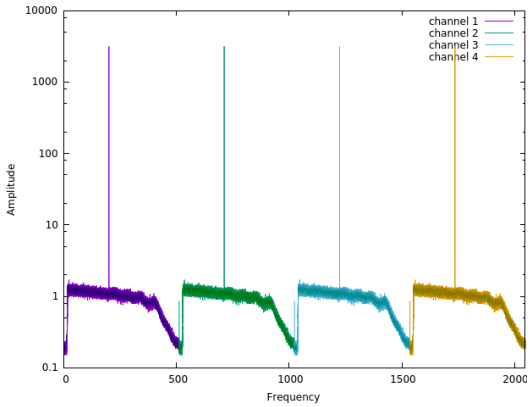


그림 9. 512MHz 대역 4채널로 구성된 8,192Mbps 기록 데이터에 대한 FFT 분석 결과(512MHz 대역폭 X 4 채널)

Fig. 9 Result of FFT processing for the 8,192Mbps recorded data(512MHz bandwidth X 4 channel)

Ⅵ. 결 론

본 논문에서는 10Gbps 수준으로 네트워크 입력되는 패킷의 수신, 기록을 위한 초고속 스토리지의 성능 평가를 진행함에 있어 이를 객관적으로 수치화하기 위한 방안에 대해 논의하였다. 이를 위해서는 매 초당

한 치의 오차없이 규칙적으로 패킷을 전송할 수 있는 차별화된 소스(Source)가 필요하고 해당 패킷에 각 프레임을 구분할 수 있는 헤더 정보가 포함되어 있어야 한다. 이에 따라 본 실험에서는 이러한 요건을 충족시키는 VDIF 데이터를 이용하였고 FILA10G를 기반으로 테스트베드를 구축하였다.

페이로드 기준으로 8,192Mbps로 입력되는 패킷의 경우 네트워크 인터페이스 상에서 손실이 없다고 가정할 경우 매 초당 125,000개가 입력되어야 한다. 해당 패킷의 메모리 버퍼링 이후 디스크 어레이에 저장되는 과정에서 손실이 없다면 링 버퍼 상의 스트림 데이터와 저장된 데이터의 크기는 정확히 일치해야 한다. 이를 검증하기 위해 우리는 디스크 어레이 상에서 가장 성능이 낮다고 평가되는 말단부에서 실험을 진행하였다. 그리고 이 두 가지 수치 정보의 일치 여부 검증을 통해 디스크 어레이 측면에서 단 1바이트의 손실도 발생치 않았음을 확인할 수 있었다.

데이터 저장에 있어서는 완벽하게 구동되고 1바이트의 손실도 없었지만 패킷 입력의 경우에는 간헐적으로 패킷이 누락되는 것을 확인했다. 600초 단위로 측정한 결과 거의 대부분 125,000개의 패킷이 입력되었지만 4~8번에 걸쳐 패킷 누락이 나타났다. 이러한 특성은 네트워크 입력되는 VDIF 패킷의 질을 검증하는 vdiftimeUDP.py를 이용한 패킷 수신에서도 간혹 확인되었는데 패킷 수신과 데이터 기록이 병행될 때 패킷 누락이 다소 증가되는 경향을 보였다[13]. 패킷 수신과 데이터 기록을 별도의 쓰레드 상에서 실행시키고 독립된 CPU 코어를 할당하였으나 부하 증가가 일부 패킷 누락으로 이어진 것으로 추측된다. 이를 해결하기 위한 방법으로 시스템 최적화가 필요하며, 단위 디스크를 기존 1TB에서 2TB 이상으로 업그레이드 한 후 성능 변화 추이를 모니터링 할 계획이다.

References

[1] J. Jang, D. Kim, and C. Choi, "Study on Hybrid Type Cloud System," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 11, no. 6, 2016, pp. 611-618.

[2] H. Hinteregger, A. Rogers, and R. Cappallo, "A high data rate recorder for astronomy," *IEEE Trans. Magnetics,* vol. 27, no. 3, 1991, pp. 3450-3460.

[3] A. Whitney, M. Kettenis, C. Phillips, and M. Sekido, "VLBI Data Interchange Format(VDIF)," In *Proc. IVS The 8th Int. e-VLBI Workshop*, Madrid, Spain, June 2009, pp. 1-7.

[4] A. Krizhanovsky, *Lock-Free Multi-Producer Multi-Consumer Queue on Ring Buffer.* Los Angeles: LINUX Journal, 2013.

[5] H. Kim and S. Ahn, "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage," In *Proc. USENIX Conf. on File and Storage Technologies,* San Jose, 2008, pp. 239-252.

[6] 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, 2017, pp. 75-84.

[7] M. Chen, S. Mao, Y. Zhang, and V. C. Leung, Big Data Related Technologies, Challenges and Future Prospects. Heidelberg: Springer, 2014.

[8] S. Kershaw, R. Spencer, R. H. Jones, R. Burgess, S. Casey, A. Rushton and P. Boven, *Protocols Performance Report - Final Report on Protocols and Network Infrastructure.* Manchester: EXPReS, 2009.

[9] R. Deri, *PF_RING API Revision:7479API documentation for PF_RING: high-speed packet capture, filtering and analysis framework.* Pisa: ntop, 2016.

[10] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols.* Boston: Addison-Wesley, 2001.

[11] A. A. Mamun, G. Guo and C. Bi, *Hard Disk Drive Mechatronics and Control.* Boca Raton: CRC Press, 2007.

[12] W. Brisken, *Walter Brisken's library for reading Mark5 and similar VLBI-format data.* Socorro: NRAO, 2014.

[13] J. Wagner, *INFORMATION ON WIDEBAND VLBI OPERATION Summaries for operating the Mark6 and FILA10G, DiFX correlation, and extracting Phase Cal.* Daejeon: KASI, 2015.

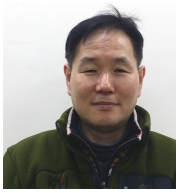
저자 소개

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

강용우(Yong-Woo Kang)

1988년 부산대학교 기계설계학과 졸업(공학사)
1990년 부산대학교 대학원 지구과학과 졸업(이학석사)
2000년 부산대학교 대학원 지구과학과 졸업(이학박사)
2000년 ~ 2001년 연세대학교 박사후연구원
2002년 ~ 2006년 연세대학교 연구전임교원
2006년 ~현재 한국천문연구원 연구원
※ 관심분야 : 전파백엔드시스템, 대용량 자료처리, 관측천문학



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