Regular paper

Journal of the KIECS. pp. 75-84, vol. 12, no. 1, Feb. 28 2017, t. 81, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2017.12.1.75

초고속 관측 데이터 수신 및 저장을 위한 기록 시스템 설계 및 성능 최적화 연구

송민규* · 강용우** · 김효령**

The Study on the Design and Optimization of Storage for the Recording of High Speed Astronomical Data

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

요 약

초고속 환경에서 대용량 데이터에 대한 안정적 기록 및 효율적인 데이터 접근의 필요성은 갈수록 높아지고 있다. 이와 관련된 기초과학의 한 분야로 방대한 천체 관측 데이터를 생산하는 VLBI(: Very Long Baseline Interferometer)가 있는데 고분해능, 고감도 관측 연구를 수행하기 위해서는 고성능의 데이터 저장 시스템이 요구된다. 하지만 시장에 출시된 대다수 클라우드 기반 스토리지는 일반 IT, 금융, 행정 서비스 지원을 위한 저용량, 복수 스트림의 비정형 데이터에 최적화되어 있기 때문에 빅 스트림 데이터 기록을 위한 최적의 대안이 될 수 없다. 본 논문에서는 이를 극복하기 위한 방안으로 데이터 입출력 처리에 있어 고성능, 동시성에 최적화된 데이터 저장 시스템을 설계하고자 한다. 이를 위해 멀티 코어 CPU 환경에서 libpcap, pf_ring 등의 API 호출을 통해 패킷 입출력 모듈을 구현하였고 외부로부터 유입되는 데이터를 효율적으로 처리할 수 있도록 소프트웨어 RAID(: Redundant Array of Inexpensive Disks) 기반의 확장성 있는 스토리지를 구축하였다.

ABSTRACT

It becomes more and more more important for the storage that supports high speed recording and stable access from network environment. As one field of basic science which produces massive astronomical data, VLBI(: Very Long Baseline Interferometer) is now demanding more data writing performance and which is directly related to astronomical observation with high resolution and sensitivity. But most of existing storage are cloud model based for the high throughput of general IT, finance, and administrative service, and therefore it not the best choice for recording of big stream data. Therefore, in this study, we design storage system optimized for high performance of I/O and concurrency. To solve this problem, we implement packet read and writing module through the use of libpcap and pf_ring API on the multi core CPU environment, and build a scalable storage based on software RAID(: Redundant Array of Inexpensive Disks) for the efficient process of incoming data from external network.

키워드

Storage, Ring Buffer, Packet Capture, VLBI
스토리지, 링 버퍼, 패킷 캡처, 초장기선 전파 간섭계

** 한국천문연구원 전파천문본부
(byulmaru@kasi.re.kr, hrkim@kasi.re.kr)
* 교신저자 : 한국천문연구원 전파천문본부
· 접 수 일 : 2017. 01. 07
· 수정완료일 : 2017. 02. 13
· 게재확정일 : 2017. 02. 24
· Received : Jan. 07, 2017, Revised : Feb. 13, 2017, Accepted : Feb. 24, 2017
· Corresponding Author : Min-Gyu Song
Radio Astronomy Division, Korea Astronomy and Space Science Institute.
Email : mksong@kasi.re.kr

Ⅰ. 서 론

초고속 환경에서 대용량 데이터에 대한 안정적 기록 및 효율적인 데이터 접근의 필요성은 갈수록 높아지고 있다. 수 Gbps의 입출력 성능을 필요로 하는 HEP(: High Energy Physics: 고에너지물리), VLBI는 그에 대한 대표적인 분야로, 해당 연구시설에서 생산되는 데이터 속도는 이미 수년 전 10 Gbps를 넘어섰고 용량 또한 수 십~수 백 TB에 이를 정도로 방대하다[1]. 본 논문에서 다루고자 하는 VLBI에서 고속의 데이터 기록이 필요한 이유는 그에 대한 구현을 통해 관측 대역폭을 용이하게 확장시킬 수 있고 기존에는 볼 수 없었던 우주의 초미세 구조에 대한 심도있는 관측 연구 수행이 가능하기 때문이다[2]. 뿐만 아니라 빠른 데이터 처리로 인해 전송, 저장 및 분석 처리 과정에서 효율성을 배가시킬 수 있는 이점도 얻을 수 있다.

데이터 기록 및 저장을 위한 스토리지 관점에서 VLBI는 일반 IT, 금융, 행정 서비스와는 명확히 구분되는 한 가지 특징이 있다. 일반적으로 스토리지에 저장되는 데이터는 인터넷 상의 불특정 다수 사용자가 접근해 사용하며 그 단위 용량은 소규모로 제한적이다[3]. 그 대표적인 사례로 웹, SNS, 클라우드 서비스 등을 들 수 있으며 이를 처리하는 스토리지에 있어 Throughput으로 지칭되는 전체 처리량은 대단히 중요한 요소이다[4]. 이와 달리 기초과학 분야에서 필요로 하는 스토리지는 단일 스트림 기준, 10Gbps에 육박하는 방대한 성능을 요구하며 과학 연구의 특성 상 사용자 수 역시 소수로 제한된다[5]. 뿐만 아니라 처리 데이터에 있어서도 구분되는데 클라우드 기반의 일반 스토리지가 비정형 데이터를 주 기록 대상으로 삼는 반면 VLBI 등 과학 연구 분야에서 다루는 데이터는 상당부분 프레임 헤더와 포맷이 고정되어 있는 형태로 정형 데이터에 해당한다[6]. 하지만 그럼에도 대부분의 스토리지 사용자가 일반 웹, 클라우드 서비스에 편향되어 있고 전체 스토리지 시장에서 상당한 비중을 차지하기에 스토리지 업체는 저용량, 복수 스트림 방식의 비정형 데이터에 최적화된 제품 위주로 출시하고 있다.

빅 스트림 데이터 처리의 경우 기존의 이러한 스토리지 구조 하에서 성능 저하는 물론 시스템 병목 현상이 불가피하다. 이에 따라 VLBI 분야에서는 그동안 기성품 형태의 스토리지 대신 특수 제작된 시스템에 관측 데이터를 기록하였다. 한 예로 지난 10여 이상 전세계 VLBI 연구기관에서 관측 데이터 저장을 위한 용도로 미국 MIT 부설 Haystack 천문대에서 개발된 Mark5B+/MarkC가 표준으로 사용되었으며 2013년에는 16Gbps의 초고속 데이터 처리를 위한 Mark6가 출시되었다[7-8]. 하지만 이들 시스템은 일반 스토리지와 달리 별도로 개발된 독립된 형태기이에 호환성 문제가 있고 데이터 분석 처리 과정에서 성능 이슈 역시 부각되고 있다.

이에 따라 본 논문에서는 이러한 문제점을 극복할 수 있는 방안을 모색하고 최적화된 시스템을 설계 및 개발하고자 한다. 초고속 입력 데이터를 효과적으로 분산 처리하기 위한 입출력 모듈 설계를 기반으로 다수의 하드디스크 어레이에 소프트웨어 RAID 기술을 적용하여 확장성 있는 데이터 저장 시스템 개념을 접목할 것이다. 나아가 대용량 데이터를 안정적으로 기록할 수 있는 시스템 안정화에 대해 논의하고자 하며 그 결과에 대해 논의하고자 한다.

본 논문은 다음의 순서에 따라 구성되었다. 본 서론에 이어 2장에서는 VLBI 분야에서 고속의 데이터 저장을 위한 시스템 현황 및 개발의 필요성에 대해 살펴볼 것이고 이를 기반으로 3장에서 시스템 개발 목표를 제시한다. 4장에서는 외부로부터 유입되는 10Gbps 수준의 데이터를 효과적으로 수신, 저장할 수 있는 시스템 설계 및 개발에 대해 기술하고자 하며 5장에서 구현된 시스템이 정상적으로 구동하는지 유효성 검증과 성능 평가 결과에 대해 서술하고자 한다. 그리고 마지막 6장에서 본 논문의 결론을 맺고자 한다.

Ⅱ. 시스템 현황 및 개발의 필요성

VLBI에서 데이터의 처리 속도는 관측 분해능 향상과 직결되며 광대역 관측의 성패는 결국 얼마나 빠른 속도로 데이터를 생성하고, 전송, 저장하느냐에 관한 문제로 귀결된다[9]. 이와 관련된 시스템에는 샘플러, 네트워크, 그리고 스토리지에 해당하는 기록기가 있다. 하지만 이 중에서 엔드 단에 위치하는 기록기는

데이터가 쓰여지는 테이프, 디스크 등 미디어의 한계로 인해 처리 속도의 고성능화 구현이 가장 어려운 시스템으로 인식되어져 왔다. 이러한 점을 고려하여 VLBI 데이터 저장을 위한 기록 시스템은 전 세계적으로 소수의 연구기관만이 시스템을 개발에 참여하고 있는 실정이며 아직까지는 Mark5/Mark6를 개발한 미국 MIT 부설 Haystack 천문대가 유럽 JIVE와 함께 VLBI 데이터 기록 분야에서 표준으로 인식되고 있다

.

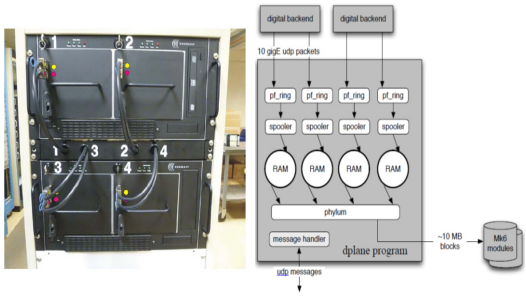


그림 1. Mark6 시스템 하드웨어(좌) 및 소프트웨어 블록 다이어그램(우)
Fig. 1 Mark 6 system hardware(Left) and software block diagram(Right)

그림 1은 Haystack 천문대에서 가장 최근에 개발된 Mark6 시스템을 소개하는 것으로 시스템 외형과 데이터 처리 메커니즘을 보여주고 있다[7,8]. Mark6 시스템은 16개의 디스크 사용을 기준으로 최소 8Gbps 이상의 데이터 기록을 지원하며 이를 안정적으로 구현하기 위해 SG(: Scatter/Gather)라는 별도의 파일 시스템을 채택해 활용하고 있다. 하지만 SG는 데이터 출력에 있어 입력 대비 일원화된 인터페이스를 지원하지 않는다. 따라서 시스템에 저장된 데이터의 분석 처리를 위해서는 리눅스 파일 형태로 별도의 변환을 진행해야 하는 번거로움이 있으며 이로 인한 데이터 처리 지연은 시스템 속도 및 용량 확장에 있어 걸림돌로 작용하고 있다.

비효율적인 외장 케이블, 디스크 팩 기반의 시스템 구조와 더불어 데이터 송수신에 있어 병목으로 작용하는 또 다른 부분으로 HBA(: Host Bus Adapter)가 있다. FC(: Fiber Channel), SAS(: Serial Attached SCSI), Infiniband 등 다양한 외부 인터페이스를 수용 및 지원하기 위한 컨버터로서 역할하는 HBA는 스토리지 상에서 입출력 성능을 저하시키는 병목의 하나로서 기존의 스토리지 구조 하에서 10Gbps에 준하는 데이터 기록은 불가능하다. 또한 버스 및 컨트롤러 기술 발전에 비례하여 시스템 성능을 개선함에 있어서도 한계가 있다[10]. 데이터 기록 관점에서 바라볼 때 데이터는 초고속 NIC(: Network Interface Card)를 통해 외부로부터 유입되어 시스템 버스를 경유하여 메모리에 수신된 후 디스크 풀에 저장되는 단순한 패턴이다. 때문에 기록 시스템 상에 굳이 HBA가 있을 필요가 없으며 10GbE NIC, 메모리, RAID 컨트롤러 등 모든 디바이스가 PCIe(: PCI express) 버스에 연결되는 구조 하에서 데이터 전달 경로를 일원화시킬 수 있다면 통신 효율성 및 데이터 처리 성능을 개선할 수 있다.

III. 시스템 개발 목표

이에 따라 본 논문에서는 상기 문제점 극복을 위해 기존의 데이터 저장 시스템에서 일차적으로 HBA를 제거하고, 네트워크, 메모리, 하드디스크 등 디바이스 간 인터페이스를 PCIe 인터페이스 기반으로 구성하고자 한다. 이를 통해 네트워크로부터 유입된 패킷이 메모리를 경유하여 하드디스크 어레이에 최단 경로로 저장될 수 있음은 물론 JBO(: Just a Bunch Of Disks) 확장 시 시스템 성능과 용량을 효과적으로 향상시킬 수 있다. 그림 2는 이러한 PCIe 인터페이스 구성된 스토리지에서의 데이터 흐름의 예를 보여주고 있다[10].

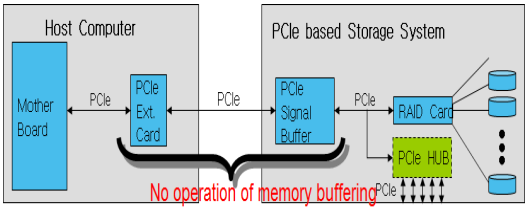


그림 2. 효율적인 데이터 전송 구현을 위한 PCIe 인터페이스 기반 스토리지의 시스템 구성
Fig. 2 System configuration of PCIe interface based storage for the efficient data transfer

또한 최근, 네트워크 상에서 초고속으로 입력되는 실시간 대용량 데이터를 효과적으로 수신 및 저장하기 위한 수단으로 멀티 프로세서, 쓰레드 기술 적용이 보편화되고 있다. 하지만 현실적으로 대부분의 스토리지가 5Gbps 미만의 패킷 처리에 만족하고 있으며 10Gbps 수준의 초고속 패킷에 대한 수신 및 저장은 찾아보기 힘든 실정이다. 이를 극복하기 위한 방안으로 본 논문에서는 멀티 프로세서, 멀티 쓰레드 환경에서 링 버퍼 적용을 통해 패킷 캡쳐 및 버퍼링 성능을 극대화할 것이다. 또한 스토리지 어레이 후반부까지 안정적인 데이터 기록이 이뤄질 수 있도록 성능 최적화를 진행하고자 한다. 링 버퍼에는 여러 종류가 있지만 오픈소스 형태로 신속한 패킷 캡쳐와 디바이스 드라이버 독립성을 제공하는 PF_RING이 최적의 수단으로 평가된다. 그림 3은 PF_RING의 동작 메커니즘을 도시한 것으로 이에 대한 시스템 적용을 통해 CPU 등 시스템 자원을 보존한 상태에서 효율적인 패킷 캡쳐를 구현할 수 있다[11].

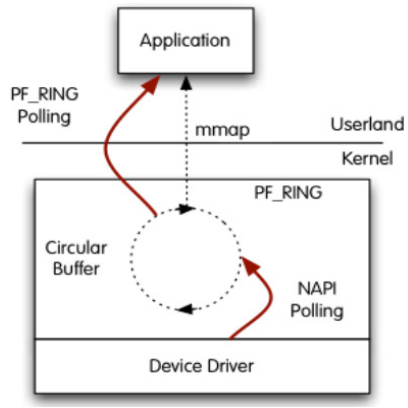


그림 3. PF_RING의 동작 메커니즘
Fig. 3 Operation Mechanism of PF_RING

PCIe 인터페이스 기반 데이터 저장을 위한 시스템 개발 완료 후 명확한 성능 평가를 위한 스토리지 구성 하드디스크 수는 16개로 지정하고자 한다. 이는 Mark6 시스템을 구성하는 하드디스크 개수와 동일한 개수이며, 데이터 기록 성능은 일단 8Gbps를 1차 달성 목표로 한다. 디바이스 사양으로는 먼저, 외부로부터 패킷을 수신하는 주 네트워크 인터페이스로 x520 칩셋이 탑재된 10GbE NIC를 선정하였고, 데이터 기록 후 즉각적인 사용이 가능하도록 RAID0 환경 하에서 리눅스 표준에 부합되는 XFS 파일 시스템을 적용하였다. 이러한 시스템 환경에서 데이터 기록 속도, 용량의 확장성을 지원하는 시스템을 설계하고자 하며 세부 논의는 다음 절에서 기술하기로 한다.

Ⅳ. 초고속 자료 저장 시스템 설계 및 개발

4.1 하드웨어 시스템 설계

우리가 개발하고자 하는 시스템은 임의의 UDP(: User Datagram Protocol) 데이터를 초고속 캡쳐, 저장할 수 있는 스토리지로 리눅스 환경에서 실행되는 소프트웨어와 고성능의 하드웨어 기성품에 기반한다. 시스템 내부에서 8Gbps 패킷에 대한 안정적인 기록을 위해서는 그 이전에 외부 네트워크 상에서 데이터 유입 시, 안정적인 데이터그램 캡쳐 및 메모리 복사가 이뤄져야 한다. 보다 신속한 데이터 입출력이 가능하도록 본 논문에서는 RAID0 기반으로 16개의 1TB 디스크를 16TB 용량의 아카이브로 가상화켰으며 테스트베드로 사용된 시스템 형태 및 사양이 그림 4에 나타나있다.

Items	Description
CPU	2CPU / Intel Xeon E5-2623 v3 3.0GHz,10M Cache,8.00GT/s QPI,Turbo,HT,4C/8T (105W)
Memory	64GB (8ea * 8GB) RDIMM, 2133MT/s, Dual Rank, x8 Data Width
Storage Capacity	16TB (16ea * 1TB) 7.2K RPM SATA 6Gbps 3.5in Hot-plug Hard Drive,13G
NIC	Intel X520 DP 10Gb DA/SFP+ Server Adapter
Tranceiver	1ea * SFP+, SR, Optical Transceiver, Intel, 10Gb-1Gb

그림 4. 초고속 데이터 기록용 스토리지 서버 샤시(좌) 및 시스템 사양(우)
Fig. 4 Storage Server Chassis for the high speed data(Left) recording and its specification for CPU, Memory, NIC and etc.(Right)

초고속 데이터 기록이 구현되기 위해서는 안정된 성능의 하드웨어를 기반으로 소프트웨어가 가동되어야 한다. 이에 따라 상기 시스템이 초고속 스트림 기록에 적합한지 검증하기 위해 먼저 입출력 성능을 측

정하였다. 데이터 소스, 네트워크 등에서 발생 가능한 외부 변수를 최소화하고 시스템 자체의 성능을 정밀 진단하기 위해 본 논문에서는 메모리 버퍼에 로딩된 임의 크기의 데이터를 스토리지에 연속적으로 기록하고 그 과정에서 데이터 증가 추이를 모니터링하는 방법을 사용하였다. 해당 실험을 통해 얻어낸 결과를 도시하면 그림 5와 같다. 이를 통해 알 수 있듯이 전체 레이드 어레이 용량 중 12TB까지는 12Gbps 이상으로 일정하게 성능이 유지되다가 이후 하향되고 있음을 확인할 수 있다. 하지만 최저 성능은 8Gbps 이상으로 본 논문에서 구현하고자 하는 스트림 기록에는 문제가 없음을 알 수 있다.

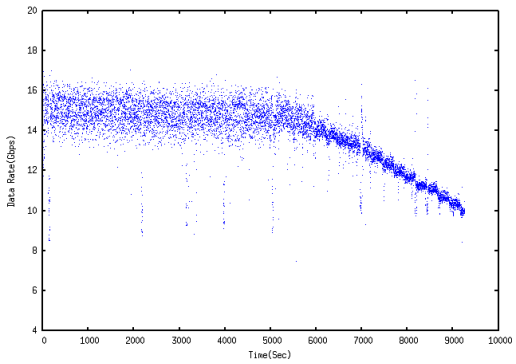


그림 5. 하드웨어 레이드 기반 스토리지 상의 기록 성능 측정

Fig. 5 Measurement of Writing performance from memory to hardware RAID based storage

4.2 소프트웨어 설계

우리가 설계 및 개발하고자 하는 데이터 기록 시스템은 네트워크에 연결된 형태로서 표면적으로는 NAS(: Network Attached Server)를 지향한다. 외부 네트워크로부터 UDP 데이터그램 형태로 유입되는 패킷을 신속하고 안정적으로 처리함에 있어 메모리 상의 캡쳐 및 버퍼링은 핵심적인 비중을 차지한다. 이를 구현하기 위한 라이브러리로 본 논문에서는 해당 분야에서 널리 사용되고 있는 libpcap과 pf_ring을 선택하였다[11-12]. 프로그램은 기능 및 역할에 따라 사용자 프로그램과 데이터 프로그램으로 세분화 가능한데 시스템 구성을 블록 다이어그램으로 나타내면 다음과 같다.

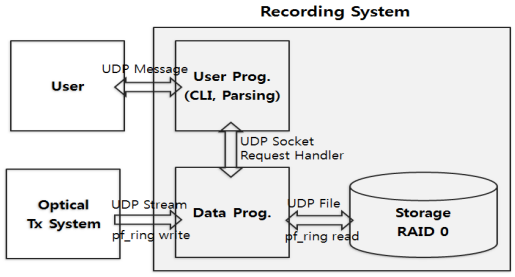


그림 6. 레이드 0 기반의 스토리지 시스템 및 사용자, 데이터 프로그램 설계

· Fig. 6 Block diagram of RAID0 based storage, which is consists of user program and data program

사용자 프로그램은 사용자로부터 명령에 해당하는 임의의 문자열을 입력받아 파싱 처리 후 데이터 프로그램으로 전달하는 역할을 한다. 데이터 프로그램은 파싱된 커맨드 및 파라미터에 준하여 시스템 상에서 실제적인 데이터 캡쳐 및 기록을 수행한다. 그에 대한 데이터 처리 내역은 시스템 상태와 더불어 다시 사용자 프로그램으로 반환되며 이를 위한 두 프로그램 간 통신은 UDP 기반 소켓 통신으로 구현하였다. 사용자 프로그램은 사용자와 데이터 프로그램 사이에서 문자열을 기반의 입출력을 수행하기에 결코 높은 성능이 필요치 않다. 다만, 효과적인 사용자 입력 전달과 더불어 에러 및 시스템 상태 관리를 위한 문자열 처리가 중요함에 따라 해당 특성을 감안하여 사용자 프로그램은 파이썬으로 개발하였다.

데이터 프로그램은 초고속 데이터 흐름에 관련된 일련의 작업을 수행하도록 하였다. 이에 따라 10GbE NIC에 수신된 데이터는 대용량 RAM 버퍼에 저장되니 후, 최종적으로 다수의 디스크로 구성된 스토리지 상에 기록 및 저장된다. 이에 대한 구현을 위해 기록 대상에 해당하는 데이터의 시작과 끝은 VLBI에서 타임 스탬프 검사를 통해 정밀하게 제어될 필요가 있다. 고속의 패킷 캡쳐 및 버퍼링을 구현하는 것은 대용량 메모리 관리는 물론 NIC, 스토리지에 대한 제어를 수반한다. 그리고 그 과정에서 상당한 시스템 자원이 요구된다. 이에 따라 데이터 프로그램은 직접적으로 메모리를 다루고 시스템 성능에 최적화되어 있는 C로 구현하였고 그에 대한 세부 구성 및 동작 메커니즘을 블록다이어그램으로 도시하면 그림 7과 같다.

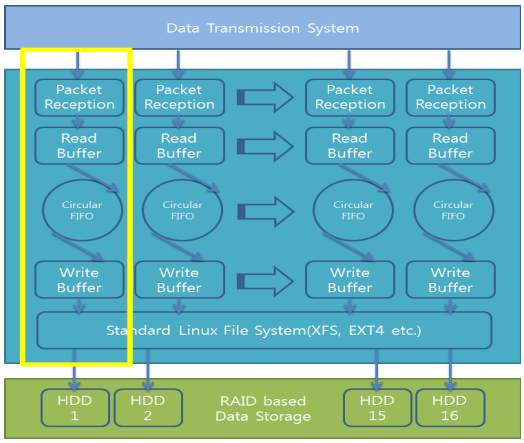


그림 7. 패킷 수신, 메모리 버퍼링 및 스토리지 기록을 위한 작동 메커니즘. UDP 기반의 각 데이터 그램은 10GbE 개별 포트로 전송됨
Fig. 7 Operation mechanism for the packet reception, memory buffering and storage writing. Each UDP based datagram is transferred to one of 10GbE port

위 블록 다이어그램에서 알 수 있듯이 패킷 처리는 실질적으로 링 버퍼를 기준으로 네트워크 단과 스토리지 단으로 구분된다. 네트워크 단은 패킷이 유입되는 소스로서 역할을 하며 스토리지 단은 입력된 패킷을 스토리지에 출력하는 목적지에 해당한다. 데이터 흐름에 따라 상기 블록 다이어그램에 기반한 데이터 처리 동작 메커니즘을 세부적으로 기술하면 다음과 같다. 먼저, 시스템에 유입되는 패킷은 libpcap을 통해 수신되고 이후 메모리를 이용한 대용량 환형 FIFO(: First In First Out) 버퍼에 입력되어 버퍼링된다. 메모리 버퍼의 용량이 클수록 고속으로 유입되는 패킷을 보다 효과적으로 처리할 수 있다. 이에 따라 시스템이 보유한 메모리 용량의 상당부분을 패킷 버퍼링을 위해 정적으로 할당하였고 타 프로그램에 의해 점유되어 사용되는 일이 없도록 하였다. 입력 패킷에 대한 버퍼링과 함께 메모리 버퍼는 스토리지에 대한 데이터 출력도 병행하여야 한다. 이 두 가지 작업을 보다 효율적이고 안정적으로 처리하기 위해 본 논문에서는 입출력 모듈 각각에 별도의 프로세서를 할당하였다. 나아가 스토리지에 대한 데이터 기록 과정에서 데이터 프로그램 내부에서 패킷 입출력이 동시에 수행되어야 하는 점을 감안하여 쓰레드 기법을 적용하였다. 대용량 패킷에 대한 보다 효과적인 입출력 버퍼링 처리를 위해 링 버퍼 기술을 적용하였고 이를 위한 API로 pf_ring 라이브러리를 호출하였다. 관련해서 데이터 프로그램 구현을 위한 주요 함수로 pfring_open, pfring_config, pfring_recv, pfring_close 등이 사용되었다[13].

일반적으로 스토리지 시스템에는 2개 이상의 10G NIC가 장착되어 있고 각 NIC는 2포트를 지원한다. 따라서 시스템에 입력 가능한 스트림 개수는 10GbE 네트워크 포트 수에 가변적이다. 이를 감안하여 그림 7에서는 패킷 수신, 입출력 버퍼를 병렬 형태로 도시하였다. 하지만 본 연구에서는 8Gbps 속도로 입력되는 하나의 스트림에 대한 메모리 버퍼링 후 기록을 목표로 하고 있으며, 데이터 소스에 해당하는 전단부의 전송 시스템과 연결되는 광 패치 케이블이 1회선인 점을 감안하여 입력 스트림은 1개로 제한된다. 이에 따라 데이터 캡쳐 및 기록을 구현하는데 있어 10GbE 포트 하나로도 충분하고 링 버퍼를 여러 개로 분할할 필요가 없다. 그림 7의 노란색 박스는 이에 대한 부분을 보여주고 있다.

4.3 스토리지 성능 최적화

이와 더불어 스토리지 상에서 안정적으로 데이터를 저장하기 위한 기술의 일환으로 일정한 속도를 갖는 하드디스크 기록 제어 방법에 대한 연구를 진행하였다. 데이터 전송 시스템으로부터 수신되는 스트림을 안정적으로 기록하기 위해서는 스토리지의 성능 변화가 최소화되어야 한다. 하지만 현재 물리적으로 다수의 하드디스크로 구성되는 스토리지의 경우 데이터 기록 전반과 후반부에 현저한 성능 차이가 발생하며 이는 스토리지의 기록 성능을 분석한 그림 5를 통해서도 확인할 수 있다. HDD(: Hard Disk Drive) 기반 스토리지 하에서 성능이 불안정할 수 밖에 없는 것은 HDD의 구조와 연관이 있다. 즉, 하드디스크는 플래터라는 커다란 원판 위에 데이터를 기록하고, 정보 입출력 과정에서 원판을 빠른 속도로 회전시키게 되는데 하드디스크의 바깥쪽과 안쪽 사이에 선속도의 차이가 존재하기 때문에 성능 격차가 발생하는 것이다. 이는 일반 사용자에게는 큰 문제가 되지 않을 수 있지만 고속 데이터 저장을 원하는 사용자의 경우 그동안 초기 일정 용량 사용 이후 더 이상 하드디스크를 사용하지

못하는 결과를 야기시켰다. 스토리지 후반부의 낮은 성능으로 사용하지 못하는 이러한 상황은 스토리지 활용에 있어서도 결코 간과할 수 없는 부분에 해당하는 것으로 개선 방안이 마련되어져야 할 것이다.

상기 부작용을 극복하기 위해서는 먼저 스토리지의 성능 편차가 최소화되어야 한다. 이를 구현하기 위한 방안으로 본 논문에서는 소프트웨어 레이드 기반으로 스토리지 재구성하는 접근법으로 안정화를 구현하고자 하며 이를 위해서는 한 가지 전제가 만족되어져야 한다. 그것은 소프트웨어 기반 레이드 어레이의 성능이 하드웨어 기반 레이드 어레이에 필적해야 한다는 것으로서 이는 본 논문에서 제안하는 방법이 스토리지 상의 데이터 기록을 개시를 일정 간격으로 제어하는 것에 착안하고, 이를 위한 수단으로 분할된 파티션을 이용하기 때문이다. 관련해서 mdadm 기반으로 구성된 소프트웨어 레이드 기반 스토리지와 하드웨어 레이드 기반 스토리지의 성능 비교 결과를 도시하면 다음과 같다.

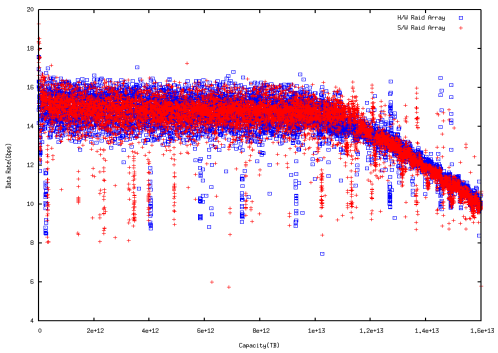


그림 8. 하드웨어 및 소프트웨어 레이드 기반 스토리지의 성능 비교
Fig. 8 Performance comparison between hardware and software RAID based storage

그림 8에서 파란색 로 표시된 점들은 앞서 그림 5와 동일하게 하드웨어 레이드 기반 스토리지의 성능을 나타낸다. 반면, 빨간색 +로 표시된 점들은 소프트웨어 레이드 기반으로 재구성된 스토리지의 성능을 보여주고 있다. 이 결과를 통해 마더보드와 HDD 간 인터페이스의 성능 한계로 입출력 속도가 15Gbps 내외로 제한되는 전반부를 포함한 스토리지 전 구간에서 하드웨어 및 소프트웨어 레이드 어레이의 성능이 일치함을 확인할 수 있다. 이는 소프트웨어 레이드를 구현하는 mdadm이 그만큼 시스템에 최적화되어 최상의 성능을 이끌어내기 때문인 것으로 풀이된다. 소프트웨어 레이드 기반으로 스토리지를 재구성하는 것은 성능 편차 해소를 통해 스토리지 후반부의 낮은 성능을 조금이라도 개선하기 위함이다. 본 논문에서는 이를 위한 접근법으로 스토리지를 구성하는 각 디스크의 파티션에 대한 분할 및 재조합을 시도하였으며, 해당 절차에 대해 기술하면 다음과 같다. 먼저 parted를 이용해 스토리지를 구성하는 n개의 하드디스크 각각에 대해 n개의 파티션을 분할하는 것이 선행되어야 한다. 이후 각 디스크에서 분할된 파티션을 앞에서부터 순차적으로 추출하여 가상의 물리 볼륨을 n개 생성한다. 가상의 물리 볼륨은 각 원본 하드 디스크에서 순차적으로 발췌된 파티션을 기반으로 구현되었기에 모두 동일한 입출력 성능을 갖는다. 이후, 마지막 단계로 LVM(: Logical Volume Manager) 적용을 통해 n개의 물리 볼륨을 거대한 하나의 볼륨 그룹으로 통합하고 해당 볼륨에 파일 시스템을 구축함으로써 최종적으로 안정적인 입출력 성능을 지원하는 거대 스토리지를 구축할 수 있다.

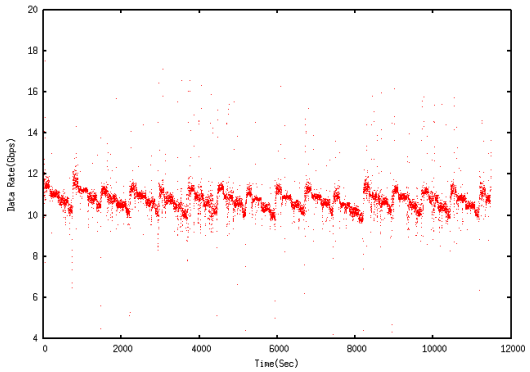


그림 9. 성능 편차 감소를 통한 소프트웨어 레이드 기반 스토리지 안정성 재고
Fig. 9 Stability of software RAID based storage is improved by making decrease variation of performance

그림 9는 이를 4.1에 기술된 스토리지 시스템에 적용하여 개선된 입출력 특성을 나타내는 것으로 이전과 비교 시 편차가 개선되고 스토리지 성능이 다소

안정화된 것을 확인할 수 있다. 이를 통해 사용자는 스토리지 전반에 걸쳐 안정적으로 데이터를 기록하는 것이 가능하다.

Ⅴ. 테스트 베드 구축 및 성능 평가

지금까지 초고속 데이터 입출력 구현을 위한 데이터 저장 시스템 설계 및 최적화 방안에 대해 기술하였다. 해당 시스템의 유효성 및 안정성이 검증하기 위한 성능 지표로 8Gbps 패킷에 대한 안정적인 저장을 설정하였다. 세부적으로는 네트워크를 통해 입력되는 데이터를 고속으로 기록하는 과정에서 패킷 손실율은 0.1% 이하로 유지되어야 하며 이를 통해 데이터의 자기상관 처리 이후 프린지가 검출된다면 시스템의 성능을 효과적으로 검증할 수 있다. 실제 천체 관측이 이뤄지는 각 전파천문대의 경우 전파망원경과 수신기를 통해 입력되는 전파 신호는 이후 BBC(: Base Band Converter), 샘플러, 데이터 획득 시스템을 거쳐 적절한 포맷의 디지털 데이터로 변환된다[14]. 이에 따라 본 논문에서는 실전과 유사한 신호 생성 및 전달 체계를 실험실 내에 재현하였고 개발된 시스템의 유효성 검증 및 정량적 성능 평가를 위해 테스트베드를 아래와 같이 구축하였다.

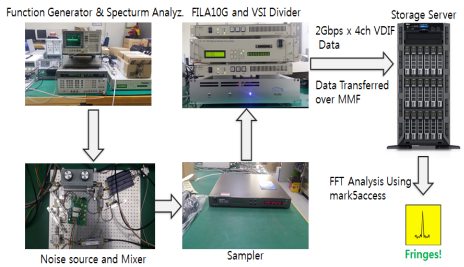


그림 10. 개발된 스토리지의 성능 평가를 위한 실험 환경 및 도출 결과
Fig. 10 Lab testbed for the performance evaluation of our developed storage and data flow graph

좌측 상단에 위치한 신호 발생기에 의해 생성된 200MHz 주파수 신호는 노이즈 신호와 합성되어 샘플러로 전송된다. 샘플러에서 아날로그 타입의 데이터는 양자화되어 디지털 신호로 변환되고, 이후 데이터 포맷 변환장치인 FILA10G를 통해 최종적으로 VDIF 기반의 UDP 데이터그램이 스토리지로 송신된다. 본 논문에서 서술한 스토리지가 정상적으로 작동하고 데이터 처리에 문제가 없다면 데이터 기록 후 그에 대한 FFT(: Fast Fourier Transform) 분석 과정에서 신호 발생기에서 생성된 것과 동일한 주파수 신호가 검출되어야 한다. 해당 주파수 크기는 200MHz로서 실제 기록된 데이터를 mark5access를 통해 분석한 결과를 도시하면 그림 11과 같다. 출력된 결과를 통해 512MHz 대역폭을 갖는 주파수 신호의 경우 스토리지 상에 2Gbps 속도로 정상적으로 기록되었음을 알 수 있다.

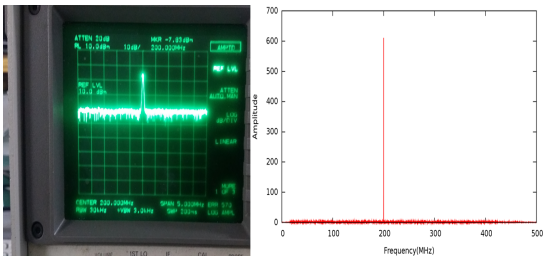


그림 11. 스펙트럼 분석기를 통해 확인한 신호 발생기 주파수 출력 200MHz와 노이즈(좌) 및 해당 합성 신호에 대한 FFT 분석 결과(우)
Fig. 11 200MHz frequency from synthesized signal generator on the display of spectrum analyzer(Left) and the result of FFT processing for the signal and noise(Right)

그림 11에 출력된 결과는 512MHz 대역폭 1채널에 해당하는 것으로 이를 동일하게 4중 복제할 경우 총 데이터 양은 4배가 되며 전송 속도 역시 이에 비례하여 8Gbps로 증가한다. 본 논문에서 개발한 시스템은 최대 10Gbps 이상의 기록 성능을 지원하는 스토리지로서 정상적으로 동작할 경우 이 신호는 완벽히 재현되어야 한다. 이를 위해 VSI(: VLBI Standard Interface) 디바이더를 통해 200MHz 주파수 및 노이즈 신호를 동일하게 4개로 복제하여 8Gbps 신호를 생성하였고 스토리지 상에서 이를 기록하였다. 하단의 그래프는 해당 데이터를 FFT 분석한 결과로서 4개의 파형이 동일한 노이즈 패턴을 보이고 있고, 동일한 위치에서 200MHz 주파수 신호가 반복되어 출력되는 것

을 확인할 수 있다. 이 결과를 통해 개발된 스토리지에 8Gbps 신호가 정상적으로 기록되었음은 물론 시스템 내부에서 데이터 캡쳐 및 버퍼링이 완벽하게 구현되었음을 검증하였다.

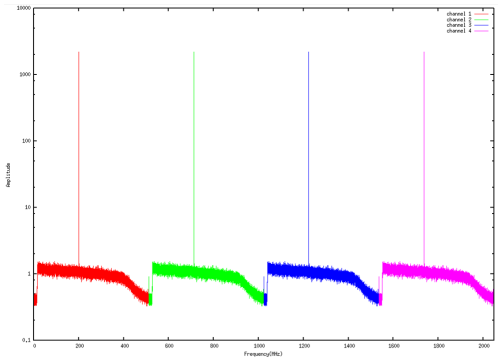


그림 12. 512MHz 대역 4채널로 구성된 8Gbps 기록 데이터에 대한 FFT 분석 결과
Fig. 12 Result of FFT processing for the 8Gbps recorded data(512MHz bandwidth X 4 channel)

Ⅵ. 결 론

10GbE, 광 모듈 도입 등 네트워크 인프라의 비약적인 발전에 따라 대용량 데이터 전송 및 저장에 대한 요구가 더욱 증대되고 있다. 교육, 의료, 방송 등의 분야 뿐 아니라 기초과학의 한 분야인 VLBI에서 고성능 스토리지의 입출력 성능은 광대역 관측 수행을 가능케 하는 핵심 요소이며 그 중요성은 더욱 부각되고 있다. 하지만 클라우드 기반의 기존 스토리지 하에서는 빅 스트림 과학 데이터에 대한 기록이 불가한 실정이고, 이에 따라 그동안 별도의 시스템을 독자적으로 구축해 사용하였다. 이에 따라 본 논문에서는 네트워크 상에서 10Gbps 레벨로 입력되는 초고속 데이터의 효율적 저장 및 관리를 위한 시스템을 설계 및 개발하였다. 이를 위해 하드웨어적 구성면에서 PCIe 인터페이스 기반으로 연결된 하드디스크 어레이를 사용하였고 libpcap, pf_ring 라이브러리 호출을 통해 입력 패킷을 효과적으로 처리할 수 있는 스토리지를 구축하였다.

본 논문에서 개발한 시스템은 몇 가지 점에서 차별화된 특성을 갖는다. 이를 요약하면 네트워크 기반의 스토리지로서 확장성을 염두에 두어 하드웨어 기반이 아닌 소프트웨어 기반의 스토리지를 설계하였고 시스템 제어를 위한 파일 시스템으로 리눅스 표준에 해당하는 XFS를 적용하였다는 것을 들 수 있다. 개발 완료된 시스템 상에서 패킷 캡쳐, 버퍼링이 정상적으로 이뤄지고 스토리지에 최종 기록된 데이터에 이상이 없는지 검증을 위해 로컬 상에 신호발생기를 필두로 임의의 주파수 신호를 생성하여 디지털 변환, UDP 데이터그램 형태의 패킷으로 전송하는 테스트베드를 구축하였고 기록된 데이터에 대한 FFT 분석을 통해 스토리지가 안정적이고 원활히 데이터를 처리하였음을 확인하였다.

현재 개발된 스토리지의 성능은 10Gbps 수준이지만 향후 30Gbps 이상을 목표로 하고 있으며 이를 위한 시스템 성능 고도화를 계획하고 있다. 나아가 Lustre, FreeNAS 등 불특정 다수의 비정형 데이터에 강점을 갖는 클라우드 기반 스토리지를 기초과학 분야의 대용량 스트림 처리에 사용 가능하도록 연구 영역을 확장할 예정이다.

References

[1] R. Spencer, R. Jones, A. Mathews, and S. O'Toole, "Packet Loss in High Data Rate Internet Data Transfer for eVLBI," In *Proc. 7th European VLBI Network Symp.*, Toledo, Spain, Oct. 2004.

[2] P. Avery, "Grid Computing in High Energy Physics," *Beauty 2003 Conf.*, Carnegie Mellon University, USA, Oct. 2003, pp. 11-15.

[3] 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.

[4] M. Lee, "A study on the Throughput Guarantee with TCP Traffic Control," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 11, no. 3, 2016, pp. 303-308.

[5] 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.
[6] M. Chen, S. Mao, Y. Zhang, and V. C. Leung, *Big Data Related Technologies, Challenges and Future Prospects.* Heidelberg: Springer, 2014.
[7] A. Rhitney and D. Lapsley, "Mark6 Next-Generation VLBI Data System," In *Proc. IVS General Meeting,* Madrid, Spain, 2012, pp. 86-90.
[8] R. Cappallo, C. Ruszczyk, and A. Whitney, "Mark6: Design and Status," In *Proc. 21st Meeting of the European VLBI Group for Geodesy and Astronomy*, Espoo, Finland, Mar. 2013, pp. 9-12.
[9] F. Takahashi, T. Kondo, Y. Takahashi, and Y. Koyama, *Very Long Baseline INterferometer.* Tokyo: Ohamsha Press, 1997.
[10] D. Yoon, *Introduction to PCI Express Interbased based High Performance Storage System.* Seoul: Verifian, 2014.
[11] L. Deri, *PF_RING High-speed packet capture, filtering and analysis.* Pisa: ntop, 2016.
[12] D. Central, *Packet Capture With libpcap and other Low Level Network Tricks.* New York: NAU's Computer Systems Engineering, 2008.
[13] L. Deri, *PF_RING API.* Pisa: ntop, 2016.
[14] A. Rhompson, J. Moran, and G. Wwenson, *Global Positioning Systems, Interferometry and Synthesis in Radio Astronomy.* New York: Wiley, 2004.

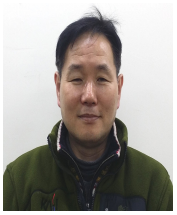
저자 소개

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