High-Performance Data Storage for Large Solar Telescope Using Distributed File System

Yingbo Liu[1,2,3], Feng Wang[1,2,3,*], Hui Deng[1,2,3], Wei Dai[1,2,3] and Shoulin Wei[1,2,3]

[1]Yunnan Observatories, Chinese Academy of Sciences, Kunming 650011, China

[2]University of Chinese Academy of Sciences, Beijing, 100049, China

[3]Computer Technology Application Key Lab of Yunnan Province, Kunming University of Science and Technology, Chenggong, Kunming, 650500, China

Abstract: New Vacuum Solar Telescope (NVST), which is an altazimuth mounting telescope, is generating data more than 2 TB per day on fine observing conditions. These data impose a great pressure on the performance of real-time data transferring and scalability of the centralized data storage. Our investigation reveals that the combination of commodity servers and the distributed file system is a cost-effective alternative solution to these issues. In this paper, We study the distributed file system to handle the massive data generated by multi-channel data acquisition system under the solar observational environment. An aggregated FITS file based on the feature of the FITS format is employed to further improve the performance of distributed data storage. Experiments show that one single channel can provide the highest writing speed at 420 MB/s under 1 Gb network by using bonding technology. The real-time data transferring rate can supply data capture more than five times than the current in NVST. The works of this paper also provide a reference to high-performance data storage of other large telescopes.

Keywords: Distributed computer systems, Lustre, High performance storage systems, Optical telescopes.

1. INTRODUCTION

New Vacuum Solar Telescope (NVST), which is located at the northeast side of Fuxian Lake, is operated by Yunnan Astronomical Observatories, Chinese Academy of Science. NVST is an altazimuth mounting telescope with 1200 mm vacuum window and 980 mm pure aperture [1, 2]. The science tasks of NVST are mainly focused on the fine structure of solar magnetic field and its evolution. Astronomers are capturing huge photospheric and chromospheric solar images with a large field of view and high resolution, and more than 2 TB data are generated per day on fine observing conditions. The mass data volume is attributed to the high spatial and time resolution of scientific cameras used in NVST. For example, an observing channel in NVST employs NEO camera (http://www.andor.com) with an sCMOS Sensor 2560×2160, which can produce images at the maximum rate up to 30 fps at full resolution (about 330 megabytes per second), and the stable growing data challenge the acquisition and storage system, involving high performance and scalability of both I/O and storage capacity.

At present, data rates of observing wavelengths in NVST are shown in Table **1**. The combined rate of multi-channel is at 364 MB per second. In the future, with new wavelengths invited (e.g. 6503 Å, 8452 Å, 10803 Å, total 13 channels are planned to put into use), the combined rate of raw data from multi-channel will be at TB per second. Data at the rate will be too high to be handled due to limitations of the transport bandwidth in a single host. NVST proposes data storage requirements including: 1) high performance data storage, more than 364 MB/s; 2) high-scalable I/O and data capacity−more channels will be added; 3) data from instruments can be saved in parallel and multiple channels can share available storage capacity; 4) continuous data storage−1-8 hours a day. Therefore, NVST needs high performance and scalable data storage architecture to meet the requirements of the current and forthcoming data storage.

We have directly considered Direct Attached Storage (DAS), Network Attached Storage(NAS), Storage Area Network(SAN) as the storage architecture of NVST, but none of them can fully meet the requirements of NVST, due to the following concerns:

- DAS: It can meet the needs during a period of time while not support continuous data storage; it is not scalable, because a HBA (Host Bus Adapter) can only support a limited number of drives;
- NAS: It can be overwhelmed by many observing channels parallel writing the device at the same time with high-performance requirements, and NAS device could become a single point of failure;
- SAN: It is a complicated and expensive solution.

We study the distributed data storage to cope with the massive data generated by multi-channel data acquisition

Table 1. NVST facility instrument data rates on multi-channel high resolution imaging system.

Channel	No. of Camera	Camera resolution (*pixels*)	Camera resolution (*pixels*)	Total Speed (*MBs*$^{-1}$)
TiO-band[1] (7058 Å)	1	2560×2160	10	80
G-band[1] (4300 Å)	1	2560×2160	10	80
H-alpah[2] (6562.8 Å)	1	4008×2672	10	204
Total			30	364

[1] employs Neo sCMOS camera of Andor.
[2] employs PCO4000 sCMOS camera of Pco AG.

system in the solar real-time observational environment based on a distributed file system-Lustre. In summary, the contributions of this paper are as follows:

- We seamlessly integrate a distributed file system and solar data acquisition system to facilitate simultaneous multi-channel and high-speed data collection for the first time, and to leverage the I/O scalability of Lustre for solar data storage.
- This paper utilizes the extensive feature of FITS format to combine several FITS files as a single storage unit to optimize I/O performance. The approach of assembling FITS file can further improve the generally recognized issue of low performance of small files on Lustre. We conduct an empirical performance evaluation of assembling FITS file on Lustre based on the layout of the FITS file.

The rest of the paper is organized as follows. In Section 2, we introduce the related works on astronomical and high-performance data storage. Section 3 reviews the preliminaries that are relevant to in this paper. A combination mode of Lustre and NVST data capture system is presented in Section 4. Performance evaluation is presented in Section 5. Discussion and a short summary are respectively provided in Section 6 and 7.

2. RELATED WORKS

In the new era of data intensive astronomy, data rates and volumes are too high for substantial human involvement [3]. Traditional single host local file systems (e.g. Ext3, Ext4) play the role of a back end infrastructure supporting data storage. They nether scale out nor perform to meet the performance and capacity of growing solar data volumes. NVST requires an alternate solution which can address the dramatic increases in I/O bandwidth and storage capacity under the solar data capture environment. Distributed file systems [4] is such a technique to survive massive data storage; it can provide high I/O performance and scalable bandwidth and it can be scaled out to fit petabyte level data storage, e.g., Lustre [5].

Astro-WISE[1] is a distributed system for astronomical data. It stores all data range from the raw data to the final science-ready product data [6]. A relational database and super massive databases are utilized to store astronomical data [7]. M. Stonebraker *et al*. use non-relational database (NoSQL) to store astronomical data [8]. SciDB [8, 9] as a kind of NoSQL is already applied in radio astronomy [10]. Distributed storage systems have already been widely used in astronomy [11], such as SDSS[2], ESO[3], ATST[4]. Gfarm, a distributed file system, has been used in astronomical data analysis and visual observatory [12]. LSST project utilizes Qserv to manage large data sets, which is based on MySQL and a distributed file system-xrootd [13].

From the perspective of both the data observing scheme and the performance requirement of I/O, NVST is similar to the ATST [14, 15]. The scalability of ATST's data storage is achieved by isolating instrument data flows on a per camera basis with a few shared resources. Different from the previous works, which mainly focus on data achieving and post-data storage, we propose to use a distributed file system to deal with the issue of massive real-time storage in the scene of NVST solar data acquisition environment. However, to our best knowledge in the literature few paper has discussed distributed file system-Lustre under the environment.

3. REVIEW

3.1. FITS

Flexible Image Transport System (FITS) [16] is the standard data format used in astronomy. It is a digital file format used for the transmission, analysis, and archival storage of scientific data sets and other images. A FITS image is commonly composed of one or more headers and data units (HDU). Each HDU consists of an ASCII formatted header unit followed by a data unit. Each header contains ASCII card images that carry keyword/value pairs, which are interleaved between data blocks. The keyword/value pairs provide information such as size, origin, coordinates, binary data format, history of the data, and anything else the creator desires. For FITS files, the first HDU, or primary header, contains no data. The primary header may be followed by one or more HDUs called extensions, which may take the form of images, binary tables, or ASCII text tables. The data type for each extension is recorded in the XTENSION header keyword [17].

[1] http://www.astro-wise.org/

[2] http://www.sdss.org
[3] http://www.eso.org
[4] http://atst.nso.edu

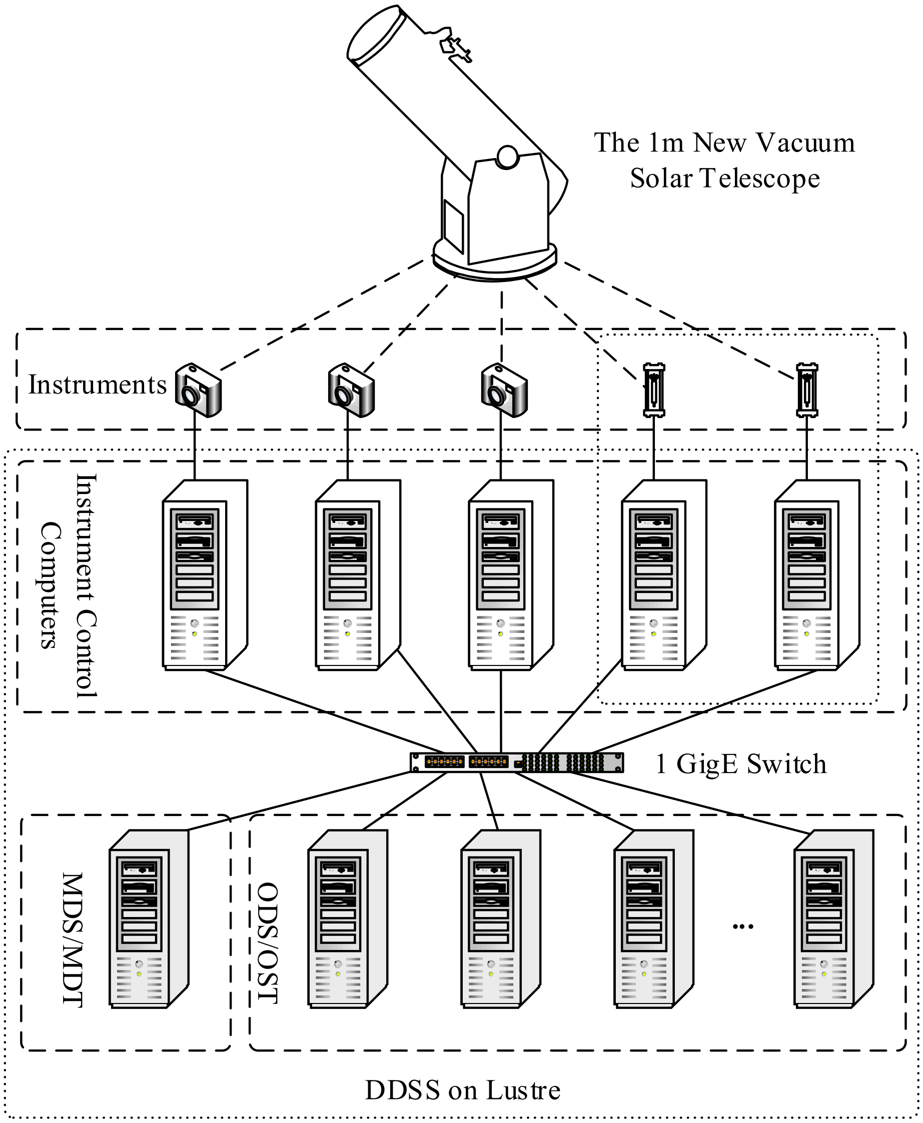
The 1m New Vacuum Solar Telescope
Instruments
Instrument Control Computers
1 GigE Switch
MDS/MDT
ODS/OST
...
DDSS on Lustre

Fig. (1). Integration of Lustre and the instrument control computer, which can support network collaboration with multi-channel solar data capture.

3.2. Lustre

Lustre [5, 18] is a GNU General Public licensed, open-source distributed file system developed and maintained by Sun Microsystems Inc. [19], and now it is long term supported by Intel Inc. Generally, it is used for large-scale parallel and capture environment, and it provides POSIX semantics and can effectively scale to support high performance computing systems with hundreds of gigabytes per second of aggregate I/O throughput.

3.3. Multi-Channel Data Acquisition

Multi-channel data acquisition system is a basic unit in solar data observing. As shown in Fig. (**1**), each channel of the system commonly has an instrument, and each instrument is connected to a computer through a special protocol, e.g. Ethernet, to transport images on the computer's hard drive. For the requirement of image quality, high performance and resolution scientific cameras are employed in NVST. With continuous data acquisition in a day, a great large volume of data could be obtained, and these data make a heavy pressure to the data storage system.

4. STORAGE SYSTEM IMPLEMENTATION

4.1. Application Diagram

This section compiles Lustre with the data acquisition system of NVST to support network collaboration with multi-channel solar data collection. The combination of a computer and Lustre is the cost-effective alternative to massive solar data storage. The integrated diagram is depicted in Fig. (**1**). The instrument control computer, which is a critical component which binds telescope and data storage system, is part of the data acquisition system, and it plays the client role in the Lustre file system.

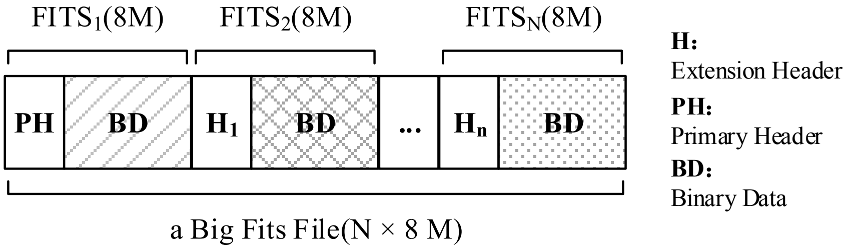

Fig. (2). A FITS image commonly consists of a primary ASCII header unit and followed by a binary data unit. This example assembles N FITS images, where N is the number of files (TiO-band or G-band each with 11 megabytes) of NVST into a larger one using the extendable HDU feature of the FITS.

The data flow of NVST (from telescope observing system to a data storage system) can be briefly described as follows: sCMOS cameras or other instruments detect raw data from the telescope, and submit these data to an instrument control computer on which the header unit of FITS files are filled. Secondly, header units and raw image data are aggregated into an integral astronomical FITS image. Finally, generated FITS images are dispatched to storage servers.

The bandwidth is a critical point that can affect the performance of this hybrid system. It is generally limited by the transmission ability between data capture instrument control computers and Lustre network. Under 1 Gb network environment, performance of clients mainly determined by NIC in a network-based storage system, For example, if the data capture server attached a 1 Gb TCP/IP NIC, actually, it can only provide less than the bandwidth of 125 MB/s, so it cannot satisfy the current I/O speed of high resolution image rebuilding in NVST. In order to provide sufficient bandwidth for data storage, We utilize NIC bonding, which provides a method for aggregating multiple network interfaces into a single logical bonded interface, to extend the bandwidth.

4.2. Aggregated FITS

Lustre has well performance on larger files, because a larger file could reduce the costs of communication to a metadata server of a distributed parallel data storage system. But the data size of NVST is relatively smaller, and small files would bring more communication between servers, the high performance of Lustre cannot fully be leveraged. It is straightforward to modify a proper FITS file size to improve I/O performance on distributed parallel data storage.

Fig. (**2**) shows an aggregated file with many sub-FITS files. A briefly cost analysis of aggregated file is as following: If an aggregated file is composed of many files, and each file with the same size S_f, the aggregated file size S_F can be described in Eq. 1,

$$S_F = N \times S_f \tag{1}$$

If an aggregated file is composed of many different file sizes ($S_{f1}, S_{f2}, ..., S_{fN}$), the total file size is as shown in Eq. 2,

$$S_F = S_{f1} + S_{f2} + ... + S_{fN} \tag{2}$$

We suppose the cost of a single direction communication between client and server is c. When a file is stored on Lustre, a client has to go through the MDS for synchronization, so a file saved to Lustre system at least has c communication cost. An assembly file is composed of N FITS sub-images. If N files are needed to be stored, an aggregated FITS file F can reduce $(N-1)c$ communication cost. Under the multi-channel observation, m is numbers of channels. Each channel works at high-speed rate. In order to simply the issue, we suppose the data capture rates of the channels are same, and each channel has to store N files. We can infer that total cost of $m(N-1)c$ can be saved. This kind of aggregation fully leverage the performance of larger file data storage, especially in the scene of real-time high-performance data storage.

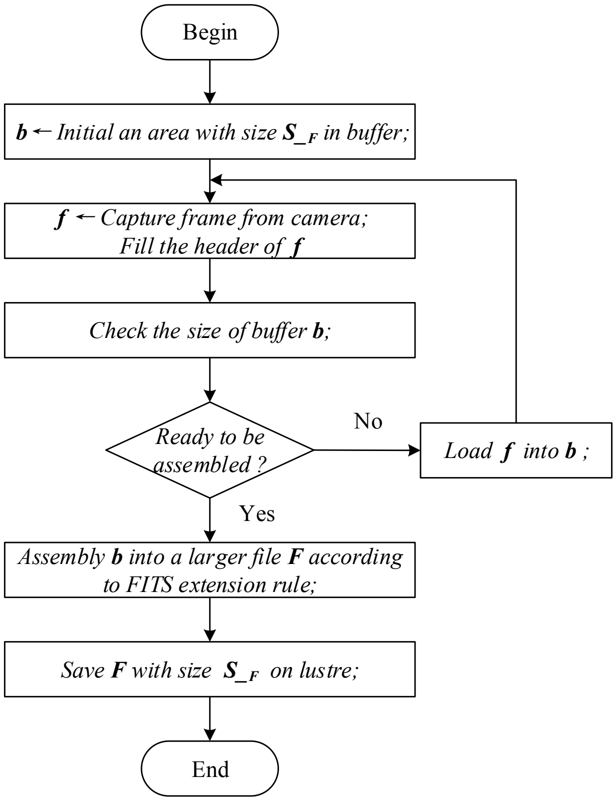
Begin
b ← Initial an area with size S_F in buffer;
f ← Capture frame from camera; Fill the header of f
Check the size of buffer b;
Ready to be assembled ?
No
Load f into b ;
Yes
Assembly b into a larger file F according to FITS extension rule;
Save F with size S_F on lustre;
End

Fig. (3). A FITS assembly flow. An aggregated FITS file conformed to the standard of multi-extension fits file format can be generated in the processing of producing FITS file.

By comparison to other aggregated approach (e.g. compression techniques), aggregated file is more convenient for researchers to use directly (e.g. An aggregated file with several independent FITS files can be recognized by astronomi-

Table 2. Sub-files contained in an aggregated fits file.

Channel	Total Speed v (MBs^{-1})	Time for Assembling a File S_F/v (sec.)	No. of sub- files S_F/S_f
TiO-band (7058 Å)	80	3.2	24
G-band (4300 Å)	80	3.2	24
H-alpah (6562.8 Å)	204	1.3	12

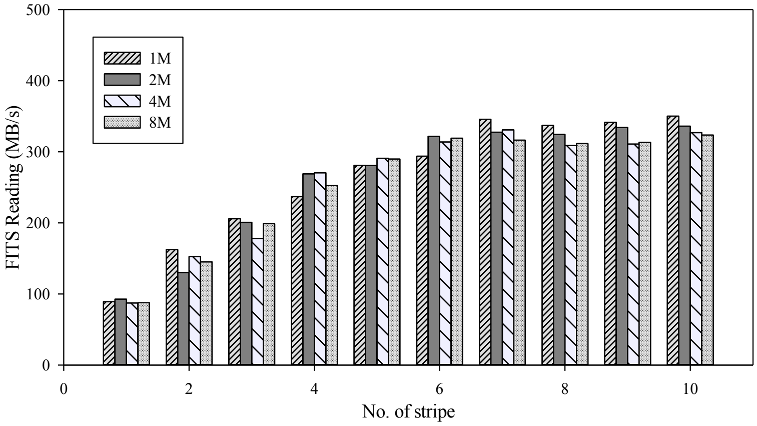

Fig. (4). Performance for writing aggregated FITS file with the size of 250 MB.

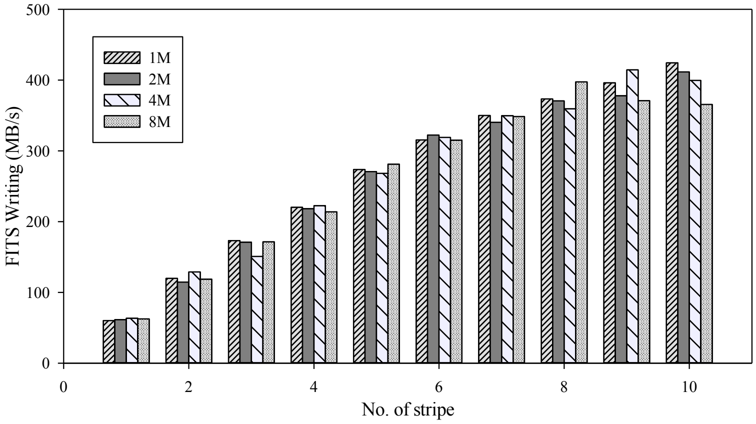

Fig. (5). Performance for reading aggregated FITS file with the size of 250 MB.

cal tools, such as FITSview, MaxIm DL, while no need to uncompressed and restore files). An aggregated FITS file can be generated in the processing of producing FITS. It is intrinsically to compile multi-independent FITS file into a single one. The assembly flow is can be described as showed in Fig. (**3**). This method leads to a straightforward data transfer as there is a contiguous data layout in memory. It is simpler and easier to build a larger FITS file when the data are transmitted from cameras and other instruments.

5. PRELIMINARY PERFORMANCE TEST

To complete initial validation of application, we set up test beds. In the test beds, Lustre 2.4 is installed on 1 MDS and 10 OSSs. The MDS has 2 AMD Opteron 2.40 G CPUs, 16 GB of memory. Three MBF2300RC drives are configured as RAID5 by utilizing LSI MegaRAID SAS 9260-8i controller. Each OSS has 2 AMD Opteron 1.05 G CPUs, 4 GB of memory, and 1 TB 7200 RPM hard disk. All machines are running CentOS 6.4 and using LVM to manage disks. A 400 GB logical volume is created on the MDS and used for metadata storage. A 400 GB logical volume is created on each OSS and used as an OST, leading to an aggregate storage space of 4 TB. A client of Lustre is also installed on a CentOS 6.4 with a data acquisition software. The client has 12 AMD Opteron 4.18 G CPUs, 32 GB of memory and 500 GB 7200 RPM ATA hard disk. Three dual ports NIC cards are attached to this server with bonding technology. It can provide the bandwidth of 750 MB/s. All nodes are connected to a 1 Gb Ethernet LAN.

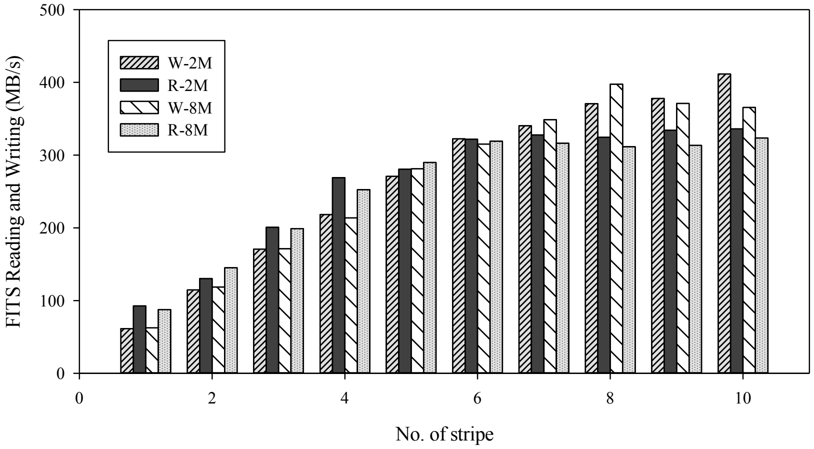

Fig. (6). Performance for writing/reading aggregated FITS file with the size of 250 MB.

The input size of workload used for our test is an aggregated FITS file of 256 MB (S_F). Each workload of NVST in Table **2** consists of several FITS files listed in Table **1**, and the time to construct an aggregated file is also showed in Table **2**. Stripe size of Lustre is default 1 MB. We mainly focus on the performance of a different number of stripes against different I/O requests (1 MB - 8 MB). The performance of the aggregated file is shown in Fig. (**4**, **5** and **6**).

Fig. (**4**, **5** and **6**) show the performance of an aggregated FITS file of 250 MB over the strip number from 1 to 10. The horizontal axis of Fig. (**4**, **5** and **6**) are the stripe sizes of Lustre, and the vertical axis is the I/O performance. As it can be seen from the Fig. (**4**), writing performance increases according to different I/O requests while the stripe is less than 8. When data cross more than 8 servers, writing performance with request buffer of 4 MB and 8 MB is slowly decreased. Meanwhile, the performance of 1 M and 2 M requests is on a steady rise. It reveals that data split into multiple segments would bring more cost for the default stripe size of 1 MB. From the Fig. (**5**), the reading performance over the designated stripe range (1-7) reveals a general trend of rising. With the size 8 M of striping used, read performance is slightly different. When aggregated I/O speed is up to 320 MB/s shown in Fig. (**6**), writing performance is higher than the reading performance. This is conformed to the actual characteristics of data storage of in NVST, which is more writing intensive than reading under real-time observation.

These results also show that we use several storage servers to meet the storage requirements of high performance writing under 1 Gb TCP/IP network. The maximum of data writing in Fig. (**4**) is about 420 MB/s which implies that we can run TiO-band (7058 Å), G-band (4300 Å) and H-alpah (6562.8 Å) simultaneously each with 10 fps, and we can independently run H-alpah (6562.8 Å) with more than 20 fps or TiO-band (7058 Å)/G-band (4300 Å) with more than 30 fps respectively. This real-time data transferring rate can supply 5.25 (420/80) times maximum data capture rates than the current in NVST with a single channel (7058 Å or 4300 Å). The utilization of network bandwidth is about 60% (total bonding bandwidth is 6×125 = 750 MB/s). This shows the potential for improving global I/O rates of distributed parallel storage using Lustre.

6. DISCUSSION

The advantages of the application mode is simple and easy to deploy, but this mode also raises several questions concerning: 1) data capture server bears heavy workloads: receiving sCMOS data, filling header unit, assembling FITS file and saving data; 2) parallel ability is constrained in the process of a single host, due to the limited resources; 3) the performance of storage is limited by network bandwidth. We can see that proper larger file can improve the I/O performance. We consider that assembling-FITS file can be a simple way to improve I/O performance for solar data capture environment. It is suitable for storing the streaming data of NVST, but it is dangerous to use an excessive larger file, because it will bring the risk of data lost due to the strategy of Lustre stripe when a data storage system encounters an internal or external failure.

The performance of Lustre has not been optimized in our experiment. There are some factors can be tuned for high performance of small I/O, so we could infer that higher performance than the results of the simulation could be improved. The experiments use a single client to simulate a single channel data observing, but it can go beyond one client, and multi-client of Lustre can fit for multi-channel data storage. Bonding technique can break through the bottleneck of 1 Gb network (One NIC with one port attached to the 1 Gb network can only provide less than the bandwidth of 125 MB/s, while we can use bonding technique to extend the bandwidth to 750 MB/s by using three NIC each with two ports), and we can expand on this idea in the 10 Gb network to further support high performance data storage in solar data capture environment. We also suggest using high performance networks to serve the astronomical data capture environment, such as InfiniBand.

CONCLUSION

There are several data storage issues of NVST that have been discussed in this paper, including surviving massive observing data-adaptive storage by using a distributed parallel file system, applying Lustre to real-time solar data storage environment, combining Lustre with a data acquisition system and aggregating small files to a larger one to improve

I/O throughput and conducting initial evaluation of the application. The application model discussed in this paper is practical and realistic, because the clients of Lustre integrated in the schemes can use POSIX call in the way of mounting. It is straightforward for astronomers to access data transparently.

This work demonstrates the potential for improving I/O rates in solar data storage by using distributed file system-Lustre. We believe this approach of data storage is valuable to solve various challenges to current issues of astronomical data storage, and we look forward to combining them with acquisitions systems in our future efforts for facing the challenges. The infrastructure of a file system discussed in this paper could be extended to another file system far beyond Lustre, including clustered file system such as GPFS, PVFS, and state-of-art system, such as Ceph, Moosefs, ClustreFS, and the works of this paper also provide a reference to other large telescope' high performance data storage.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China (No. U1231205, 11263004, 11203011, 11163004 and 11103005) and Natural Science Foundation of Yunnan Province (2013FA013, 2013FA032, 2013FZ018). The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES

[1] Z. Liu and J. Xu, "1-meter near-infrared solar telescope," In *First Asia-Pacific Solar Physics Meeting ASI Conference Series*, vol. 2, 2011, pp. 9–17.

[2] Z. Liu, J. Xu, B.Z. Gu, S. Wang, J. Q. You, L. X. Shen, R. W. Lu, Z. Y. Jin, L. F. Chen, K. Lou, Z. Li, G. Q. Liu, Z. Xu, C. H. Rao, Q. Q. Hu, R. F. Li, H. W. Fu, F. Wang, M. X. Bao, M. C. Wu and B. R. Zhang, "New vacuum solar telescope and observations with high resolution," *Research in Astronomy and Astrophysics*, vol. 14, no. 6, p. 705, 2014.

[3] M. J. Graham, S. G. Djorgovski, A. Mahabal, C. Donalek, A. Drake, and G. Longo, "Data challenges of time domain astronomy," *Distributed and Parallel Databases*, vol. 30, no. 5-6, pp. 371–384, 2012.

[4] B. Tierney, J. Lee, L. T. Chen, H. Herzog, G. Hoo, G. Jin, and W. E. Johnston, "Distributed parallel data storage systems: A scalable approach to high speed image servers," In *Proceedings of the second ACM international conference on Multimedia*. ACM, 1994, pp. 399–405.

[5] P. J. Braam and R. Zahir, "*Lustre: a Scalable, High Performance File System,*" Cluster File Systems, Inc, 2002.

[6] J. Mwebaze, J. McFarland, D. Booxhorn, and E. Valentijn, "A data lineage model for distributed sub-image processing," In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. ACM, 2010, pp. 209–219.

[7] J. Becla, A. Hanushevsky, S. Nikolaev, G. Abdulla, A. Szalay, M. Nieto-Santisteban, A. Thakar, and J. Gray, "Designing a multi-petabyte database for lsst," In *Astronomical Telescopes and Instrumentation*. *International Society for Optics and Photonics*, 2006, pp. 62700R–62700R.

[8] M. Stonebraker, P. Brown, A. Poliakov, and S. Raman, "The architecture of scidb," In: *Scientific and Statistical Database Management*. Springer, 2011, pp. 1–16.

[9] P. Cudré-Mauroux, H. Kimura, K.-T. Lim, J. Rogers, R. Simakov, E. Soroush, P. Velikhov, D. L. Wang, M. Balazinska, J. Becla, D. DeWitt, B. Heath, D. Maier, S. Madden, J. Patel, M. Stonebraker, and S. Zdonik, "A demonstration of scidb: A science-oriented dbms," In *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1534–1537, 2009.

[10] G. van Diepen, "*Scidb radio astronomy science case*," 2013. [Online]. Available: http://www.scidb.org/UseCases/radioAstronomy/usecase.pdf

[11] J. Withington, "Distributed data storage," *Astronomical Data Analysis Software and Systems XIII*, vol. 314, p. 66, 2004.

[12] O. Tatebe, N. Soda, Y. Morita, S. Matsuoka, and S. Sekiguchi, "Gfarm v2: A grid file system that supports high-performance distributed and parallel data computing," In: *Proceedings of the 2004 Computing in High Energy and Nuclear Physics*, 2004.

[13] D. L. Wang, S. M. Monkewitz, K.-T. Lim, and J. Becla, "Qserv: A distributed shared-nothing database for the lsst catalog," In *State of the Practice Reports*. ACM, 2011, p. 12.

[14] B. Cowan and S. Wampler, "Technologies for high speed data handling in the atst," *Astronomical Data Analysis Software and Systems XX*, vol. 442, p. 297, 2011.

[15] S. Wampler and B. Goodrich, "A scalable data handling system for atst," *Astronomical Data Analysis Software and Systems XVIII*, vol. 411, p. 527, 2009.

[16] W. D. Pence, L. Chiappetti, C. G. Page, R. Shaw, and E. Stobie, "*Definition of the flexible image transport system (fits)", version 3.0*," A&A, vol. 524, p. A42, 2010.

[17] E. Smith. (2011) *Multi-extension fits file format*. [Online]. Available: http://www.stsci.edu/hst/HST/overview/documents/datahandbook/intro/ch23.html

[18] P. J. Braam, "*The lustre Storage Architecture*," Inc. http://www.clusterfs.com, vol. 8, p. 29, 2003.

[19] W. Feiyi, S. Oral and G. Shipman, "*Understanding Lustre Filesystem Internals*," ORNL/TM-2009/117, 2009.

Received: November 26, 2014 Revised: January 08, 2015 Accepted: January 21, 2015