PROCEEDINGS OF SCIENCE

Explore New Computing Environment for LHAASO offline data analysis

Qiulan Huang[1]
Institute of High Energy Physics, Chinese Academy of Sciences
19B Yuquan Road, Shijingshan District, Beijing,China
E-mail: huangql@ihep.ac.cn

Gongxing Sun
Institute of High Energy Physics, Chinese Academy of Sciences
E-mail: sungx@ihep.ac.cn

Qiao Yin
Institute of High Energy Physics, Chinese Academy of Sciences
E-mail: yinqiao@ihep.ac.cn

Zhanchen Wei
Institute of High Energy Physics, Chinese Academy of Sciences
E-mail: weizc@ihep.ac.cn

Qiang Li
Institute of High Energy Physics, Chinese Academy of Sciences
E-mail: liqiang88@ihep.ac.cn

This paper explores a way to build a new computing environment based on Hadoop to make the Large High Altitude Air Shower Observatory(LHAASO) jobs run on it transparently. Particularly, we discuss a new mechanism to support LHAASO software to random access data in HDFS. This new feature allows the Map/Reduce tasks to random read/write data on the local file system instead of using Hadoop data streaming interface. This makes HEP jobs run on Hadoop possible. We also develop MapReduce patterns for LHAASO jobs such as Corsika simulation, ARGO detector simulation (Geant4), KM2A simulation and Medea++ reconstruction. A user-friendly interface is provided. In addition, we provide the real-time cluster monitoring in terms of cluster healthy, number of running jobs, finished jobs and killed jobs. Also the accounting system is included. This work has been in production for LHAASO offline data analysis to consume CPU about 20,000 hours per month since September, 2016. The results show the efficiency of IO intensive job can be improved by about 46%. Finally, we describe our ongoing work of data migration tool to serve the data move between HDFS and other storage systems.

International Symposium on Grids and Clouds (ISGC) 2018 in conjunction with Frontiers in Computational Drug Discovery
16-23 March 2018
Academia Sinica, Taipei, Taiwan

[1]Speaker

PoS(ISGC 2018 & FCDD)021

1. Introduction

The exploitation of a new computing environment is necessarry to overcome a series of challenges with the development of the new generation of High Energy Physics(HEP) experiment. The Large High Altitude Air Shower Observatory(LHAASO)[1] is a project to build a multi-component air shower detector, located at the Daocheng site(Sichuan province, P.R.China) at the altitude of 4410m, with the expectation of the most sensitive project to study Gamma Ray Astronomy in the energy range of ~2×10^{11}-10^{15} eV and Cosmic Ray studies at the energy range of ~10^{12}-10^{18} eV. LHAASO is expected to take data in 2018 and to generate 2PB raw data per year , which requires massive storage and large scale computing power. Therefore, the new solution has to meet the scalability and performance requirements.

With the growing HEP data, computing scientists have made a lot of efforts to learn new technologies to cope with the challenges brought by the mass data. Recently, Big Data and Artificial Intelligence technologies have gradually been used in HEP offline data processing. Scientific community lauched some projects to bring Hadoop[2] into scientific computing. In 2009, Brian Bockelman introduced Hadoop as a grid storage element for one LHC experiment, the Compact Muon Solenoid(CMS)[3]. After this, serval OSG sites started to use HDFS as the Storage Element in OSG Grid[4]. Also CERN proposed to use HBASE to manage experimental data in 2012[5]. As showned in Figure 1, A new computing platform for some HEP offline data processing is proposed. It adopts the efficient parallel framwork of Hadoop to improve the parallism of data processing and makes the computing mode switched from “Move data to computation” to “Move computation to data”. The new computing architecture extends the HDFS data access mechanism to make data access localized to achieve high I/O performance. And the expensive facilities like powerful network switches and disk arrays are eliminated in the new computing system, which means that less money is devoted to the computing farm while higher performance is obtained.

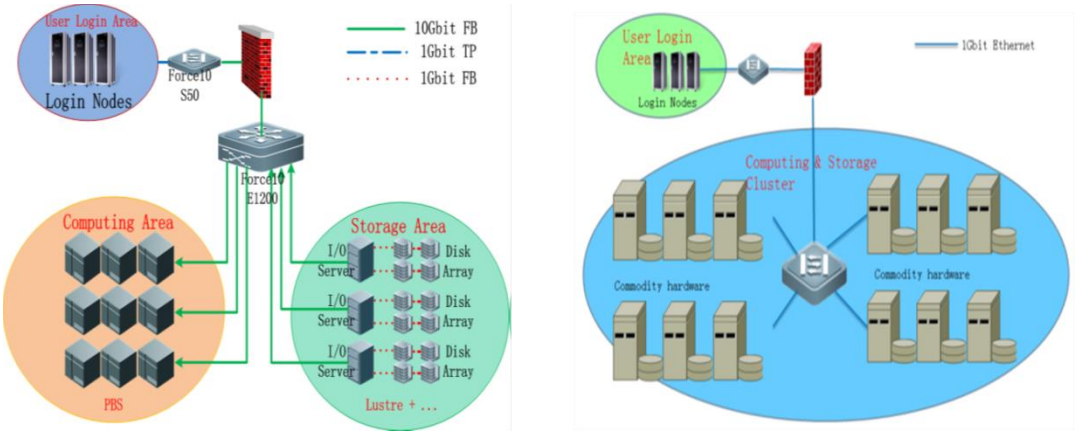

Figure 1. Traditional computing architecture and new computing architecture

Hadoop is an open-source data processing framework that includes a scalable, fault-tolerant distributed file system, HDFS[6] and parallel programming MapReduce[7], developed by Apache. Although Hadoop gained a lot of attention from industry for its scalability and parallel computing framework for large data sets, it is still difficult to run LHAASO data processing tasks directly on Hadoop. In this paper, we will explore ways to build a new computing environment using Hadoop to make LHAASO jobs run transparently. Particularly, we will discuss a new data access mechanism to support LHAASO software to randomly access

data in HDFS. Because HDFS streams data only supporting sequential write and append. It cannot satisfy LHAASO jobs access data randomly. This new feature allows Map/Reduce tasks to randomly read/write on the local file system of data nodes instead of using Hadoop data streaming interface, which makes it possible to run HEP jobs on Hadoop. We also developed diverse MapReduce patterns for LHAASO jobs such as Corsika simulation, ARGO detector simulation (Geant4), KM2A simulation and Medea++ reconstruction. And we wrapped the patterns to make them transparent to users. In addition, we provide a real-time cluster monitoring system in terms of cluster load, number of running jobs, number of finished jobs and number of killed jobs. Also the accounting system is included. This work has been in production for LHAASO offline data analysis to consume CPU about 20,000 hours per month since September, 2016. Results show that the efficiency of I/O intensive jobs can be improved by about 46%. Finally, we describe our ongoing work of data migration tool to serve the data movement between HDFS and other storage systems.

2. HDFS data access extension

HDFS is a scalable, fault-tolerant Distributed File System. It provides high throughput acess to application data and is suitable for cases that handle large data sets. It supports a few POSIX requirements to enable streaming access to data. As a result, it cannot support random read and write operations. Howerver, the I/O pattern of HEP software is random access, and physical data is stored with ROOT format[8]. In order to make LHAASO jobs run on the new computing environment, it's urgent to extend the HDFS data access methods to support random read and write.

2.1 Design and Implementation

There are one NameNode and several DataNodes inside of HDFS. The NameNode contains all the information regarding which block is stored on which particular DataNode in HDFS, hence client needs to interact with the NameNode to get the address of the specific DataNode where the requested blocks are actually stored. Whatever clients read or write in HDFS, clients need to communicate with the NameNode to get block path firstly, then clients can access the blocks by calling HDFS streaming data access API. Additionally, NameNode is responsible to check whether clients are authorized to access that block. So it gives a security token to clients which they need to communicate with the DataNode for authentication. After getting the address of DataNode that contains the specific blocks, clients will directily connect to the DataNode to read/write blocks with FSDataInputStream/FSDataOutputStream interfaces.

We designed and implemented the new data access by extending the data access methods of HDFS. This new data access allows the HEP software to read/write data directly in the local File System with the read/write interfaces of local File System instead of calling FSDataInputStream/FSDataOutputStream interfaces. We modified the source code of HDFS to enable the client to access data in the local File System. Figure 2 shows the data read flow.

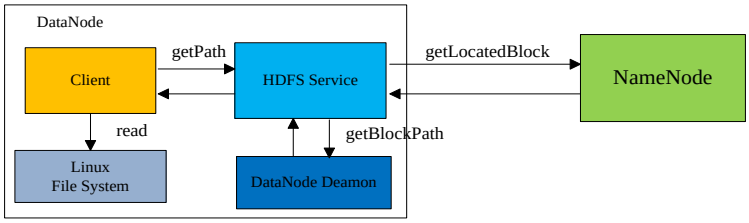

Figure 2. Read data flow

As shown in Figure 2, it is known that when clients read data in HDFS, clients should connect to the NameNode to request the location of blocks. After getting the block path, clients can access data locally. To make the new data access work, we have to schedule the map/reduce tasks to the DataNode where the data is actually stored.

Write operations can be a little more complicated, as shown in figure 3. The client sends a write data request to create a file on HDFS. Then the HDFS service makes an RPC call to the NameNode to create a new file in the File System's namespace. The NameNode performs various checks to make sure the file doesn't exist and the client has the permission to create the file. If all these checks passed, NameNode will return back to client with the block path to figure out the address of the DataNode and the physical data path. Otherwise, an IOException will be thrown to the client. Receiving the location where the block will be stored, the client starts to write data in the local File System. When the write operation is done , it calls close() method, then notifies the NameNode to change the file' status as completed.

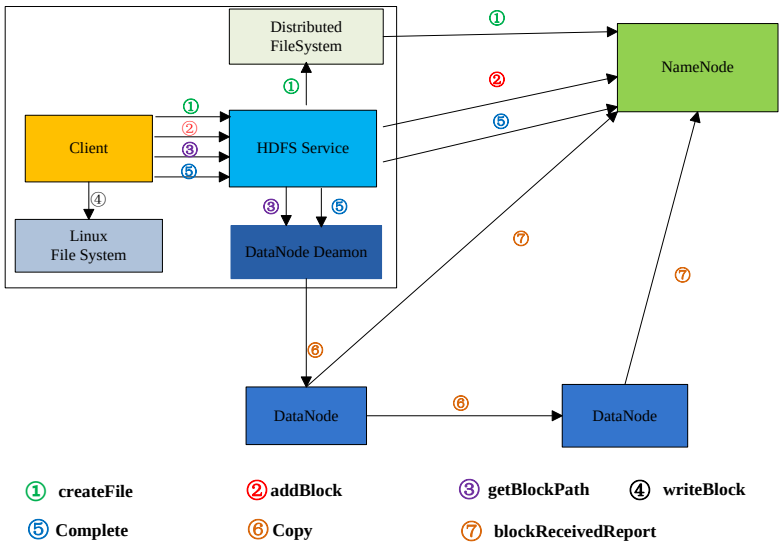

Figure 3. Write data flow

2.2 Evaluation

The new computing environment is implemented with Hadoop(MapReduce 2.0 and modified HDFS 2.6.0). We evaluated the performance of the data access extension in HDFS by analysing the LHAASO offline data processing jobs running in the new computing environment. The jobs include Cosmic Ray simulation jobs(corsika), detector simulation jobs(Geant4) and ARGO reconstruction jobs(medea++) . Also, the I/O performance of HDFS with the new data access is tested by the ROOT tool. The testbed is consisted of six nodes, one is NameNode and the other five are DataNodes(6*6TB disks each). The network is 1Gb Ethernet connection.

In order to evaluate the I/O performance of HDFS, we performed a comparison of HDFS with Lustre using the ROOT tool to get read and write performances. The test commands are listed as the following:

1)Root Write: $ROOTSYS/test/Event EventNumber 0 1 1

2)Root Read: $ROOTSYS/test/Event EventNumber 0 1 20

Test results are shown in Figure 4. Compared wtih Lustre, write event performance of HDFS is improved by 10% and read performance is increased 2~3 times.

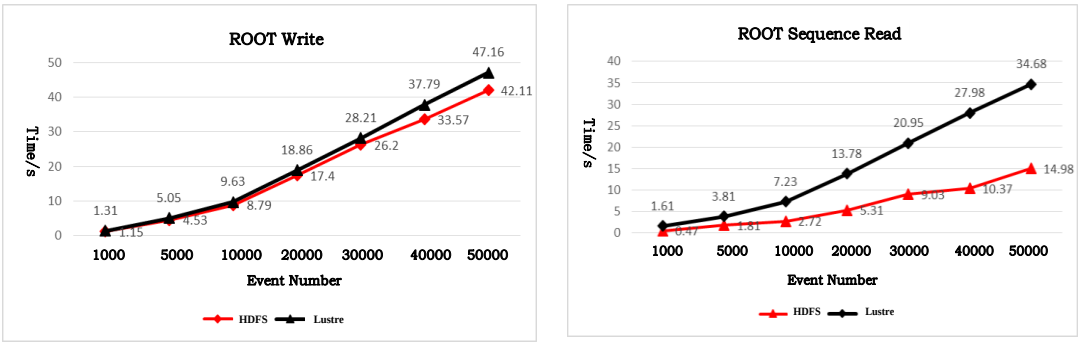

Figure 4. IO performance comparison between HDFS and Lustre

We also submitted real jobs to test the CPU efficiency of the LHAASO simulation and reconstruction jobs.. As shown in Figure 5 and Figure 6, it is found that the CPU utilization ratio of CPU intensive jobs (corsika and Geant4) is up to 100%. It illustrates that the performance of HDFS and Lustre is comparable. And the CPU utilization ratio of I/O intensive job(medea++) is 100% on HDFS, while 67% on Lustre. As the I/O intensive job requires large I/O over network as well as the Lustre cient service consumes additional system overhead. Both of them affect the job operation efficiency.

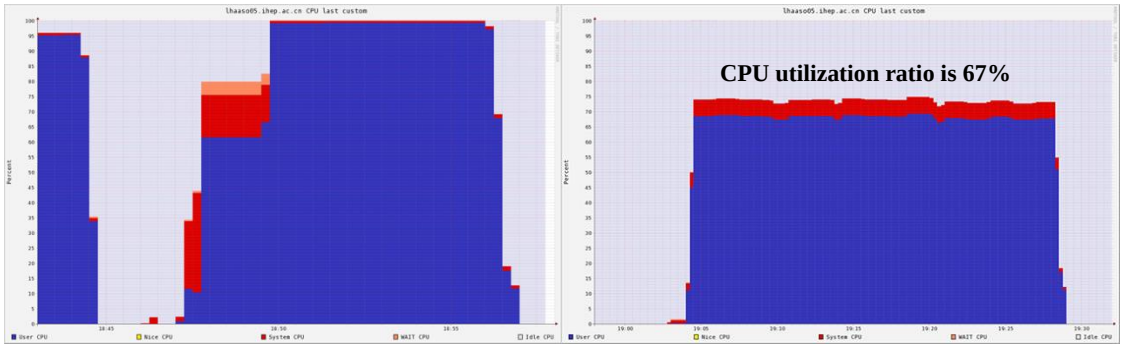

Figure 5. CPU utilization ratio in HDFS

Figure 6. CPU utilization ratio in Lustre

3. Data migration

The data migration tool is a good supplement of the Hadoop computing cluster in HEP. It provides input/output mode to move data between HDFS and other storage systems. Introducing the MapReduce parallel programing framework, this tool achieves high parallelism in data transfer. In our environment, most of the experimental data is stored in Lustre[9] and EOS[10]. So we mainly serve the data migration between HDFS and Lustre/EOS. When users want to do data analysis, the processing data should be migrated into HDFS in advance. Figure 7 shows the architecture of the data migration system.

System front-ends: It is the interface for users, we provide both Client command line and Web Portal.

Task Managerment layer: Classify the data migration requests and dispatch them to the Hadoop cluster. In this layer, dynamic priority scheduling strategy is proposed. The scheduling strategy considers the request attributions (such as users' priority, the urgent of data transfer), load of cluster, queue status and the data set size to be moved. According to those factors, the schedular calculates the the priority of each user's request. With the priority order, data transfer requests will be processed in the order of priority.

Migration service layer: It is the key service in the system, which is responsible to monitor and collect the requests from users, parse them to identity data movement mode, and then wrap the requests as map/reduce pattern.

Data Migration layer: Split the request jobs into a reasonable number of tasks according to the load of cluster, then schedule the map tasks to the DataNodes in the cluster. The task will call the GridFTP[11] API to transfer data between different sources.

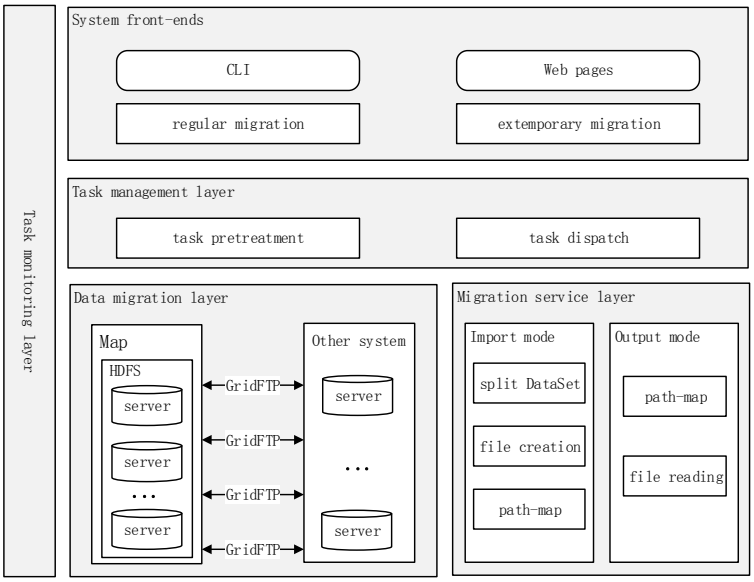

Figure 7. The architecture of data migration

Thanks to the the dynamic priority request scheduling and the combination of MapReduce and GridFTP, this tool achieves good performance in data migration. The data transfer is relatively smooth and the peformance is up to 115MB/s of each node under the 1Gb Ethernet network, shown as Figure 8.

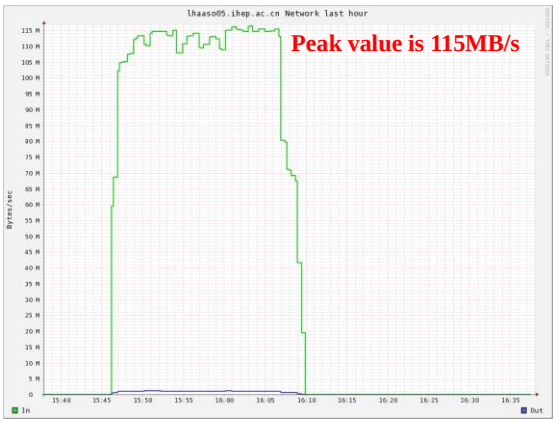

Figure8. The performance of one node

4. Status in LHAASO

We introduced a new compution architecturce into the LHAASO experiment in IHEP. Since 2016, the new computing environment has been put into operation, aiming to provide distributed computing services for Cosmic Ray simulation, ARGO dector simulation and Medea++ construction. Figure 9 shows the topology of the new computing cluster. It consists six nodes, one is NameNodea and the other are five DataNodes. There are 120 CPU cores and 140TB storage in the cluster.

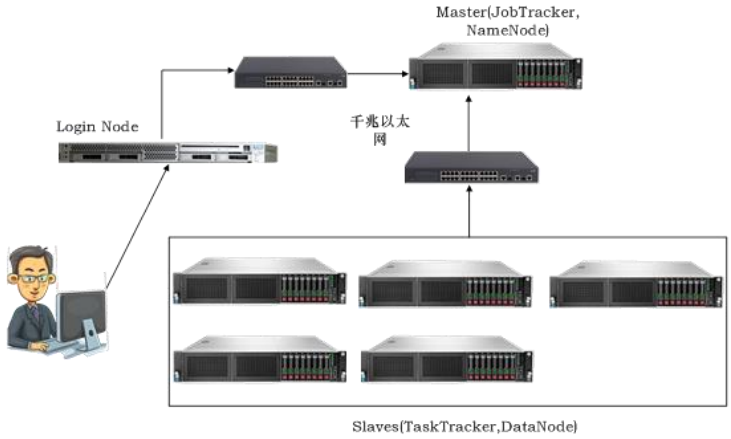

Figure 9. The architecture of the new computing cluster

The platform is public to all the users devoted to the LHAASO experiment. In 2017, the amount of finished jobs is 20,225(502,341 tasks), which cost about 212,730 CPU hours. And 119TB capacity is used.

We deployed the real-time monitoring system with Ganglia to monitor the cluster status. By exposing the metrics to Ganglia instance so that Ganglia can detect potential problems of the cluster, we can easily find the issues and figure out them to ensure the availabilty of the cluster. Except the default metrics like CPU, disk, load, memory, network, and process, custom metrics are added to monitor the number of running jobs, finished jobs, pending jobs and killed jobs. Also accounting system was developed to store all the finished jobs in database. The information of one item in database includes job identification ID, job owner, start time, end time, physical memory, virtual memory, execution node,Wall time and CPU time. As shown in Figure10, with the accounting data, it is easy to do job statistics.

ExecHost	All Job Walltime(h)	All Job CPUTime(h)	Efficiency	All Job Sum		Failed Job WallTime(h)	Failed Job CpuTime(h)	Failed Job	Fail Rate	
ybjslc05.ihep.ac.cn	114064.083	212730.631	1.865	20225	Detail	0.000	0.000	3318	0.164	Detail
Tot/Ave	114064.083	212730.631		20225		0.000	0.000	3318	0.164	

Figure 10. Job statistics

5. Conclusion

It is presented in this paper the exploitation of adopting Hadoop to build a new computing infrastructure for LHAASO. This work has been put in production for the LHAASO offline data analysis since September, 2016. It is a successful story in LHAASO and reduces the cost of facilities.

To make LHAASO jobs run on the new computing environment, some activities are necessary to be taken. Particularly, HDFS data access extension is implemented to support random read/write operations and data modification, reinforcing the data access of HDFS, not

limited to streaming access. We moved the LHAASO simulation and analysis jobs towards the new solutions. The obtained results demonstrate that the job efficiency is greatly improved with the new solution, especially for the I/O intensive jobs. The performance is increased by about 46%. At the same time, data migration tool is designed and implemented to serve data migration between HDFS and other storage systems like Lustre and EOS in IHEP, which switches from the data transfer requests to MapReduce jobs running on the Hadoop cluster and adopting GridFTP API to move data. It is proved that the tool provides high data transfer perfomance, which is a good supplement of Hadoop ecosystem. Also, the user-friendly interface, monitoring and accounting systems are provided to make the new computing environment ease of use and stable.

The project keeps scaling up, we prefer to extend the new solution to other HEP experiments like Ali CMB project[11]. And in order to improve the efficiency of more complicated statistic jobs, we are working on memory computing at present .

6. Acknowledgment

This project is supported by the Chinese NSF grants "Study on Key Technologies of Memory Computing for High Energy Physics Partial Wave Analysis (No.11875283, No. 11775249)".

References

[1] Sciascio, et al. The LHAASO experiment: from Gamma-Ray Astronomy to Cosmic Rays. CRIS 2015 Conference[hep-ex].

[2] *Welcome to Apache Hadoop*. http://hadoop.apache.org/.

[3] Bockelman B. *Using Hadoop as a grid storage element*[C]Journal of physics: Conference series. IOP Publishing, 2009, 180(1): 012047.

[4] Bradley D, Dasu S, Maier W, et al. *A Highly Distributed, Petascale Migration from dCache to HDFS*[C] HEPiX Fall 2011 Workshop. Vancouver, USA, 2011: 1-24.

[5] Lassnig M, Garonne V, Dimitrov G, et al. *ATLAS Data Management Accounting with Hadoop Pig and HBase*[C] Journal of Physics: Conference Series. IOP Publishing, 2012, 396(5): 052044.

[6] *HDFS architecture*. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

[7] DEAN J, GHEMAWAT S. *MapReduce: simplified data processing on large clusters* [J]. Communications of the ACM, 2008, 51(1): 107-13.

[8] BRUN R, RADEMAKERS F. *ROOT—an object oriented data analysis framework* [J]. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 1997, 389(1): 81-6.

[9] KOUTOUPIS P. *The lustre distributed filesystem* [J]. Linux Journal, 2011, 2011(210): 3.

[10] AJ Peters, EA Sindrilaru et al. *EOS as the present and future solution for data storage at CERN*. Journal of Physics: Conference Series. IOP Publishing, 2015, 664(2015):042042.

[11] Hong Li, Si-Yu Li, Yang Liu, Yong-Ping Li et al. Probing Primordial Gravitational Waves: Ali CMB Polarization Telescope. Published on line by National Science Review, 2018 arXiv:1710.03047.