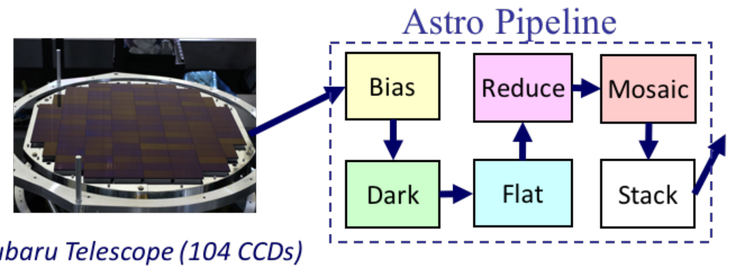
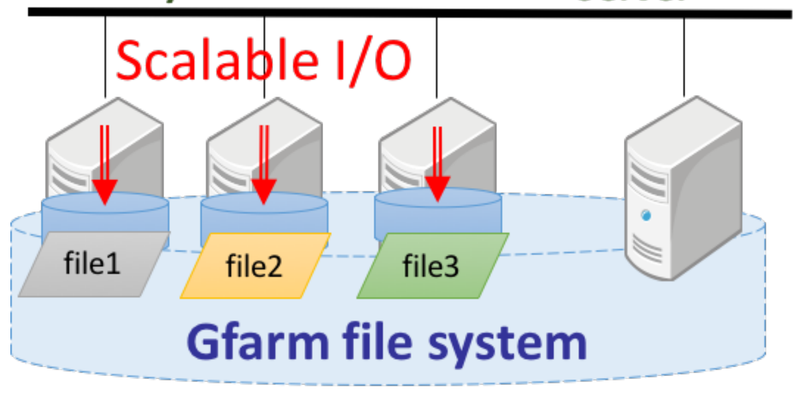
# Software Researches for Big Data and Extreme-Scale Computing

## Gfarm: a High Performance Distributed File System for Supercomputing [1,2]

Gfarm file system is an open source distributed file system. It is designed for both the cluster environment for high performance data analysis, and the geographically distributed environment for global data sharing and archive. Gfarm provides high performance by exploiting parallel I/O, and high availability by leveraging data replication service. Gfarm is used in a variety of scientific projects as the astronomical pipeline for Subaru telescope with 104 CCDs.

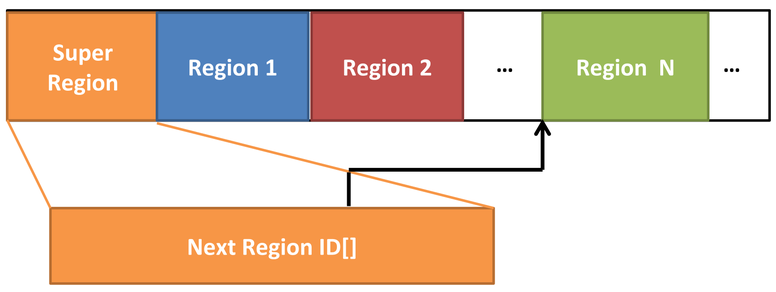


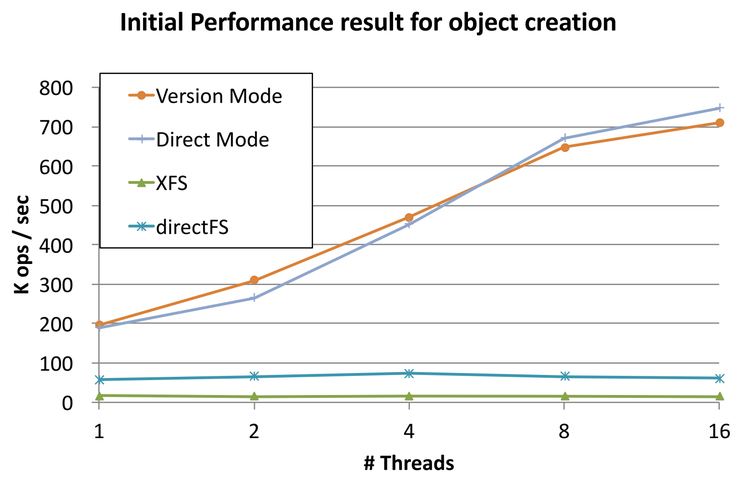

*Subaru Telescope (104 CCDs)*




## Object Storage using OpenNVM for High-performance Distributed File System [3]

This is a fast object storage for ioDrive that supports **virtual address space** and **atomic-write**. In our object storage, **regions** are located in fixed position in virtual address space in ioDrive. One region manages one object. All meta-data about the object store the region. We use the address of the first sector of region as object ID. Our object storage has two writing modes in region: Version Mode and Direct Mode.



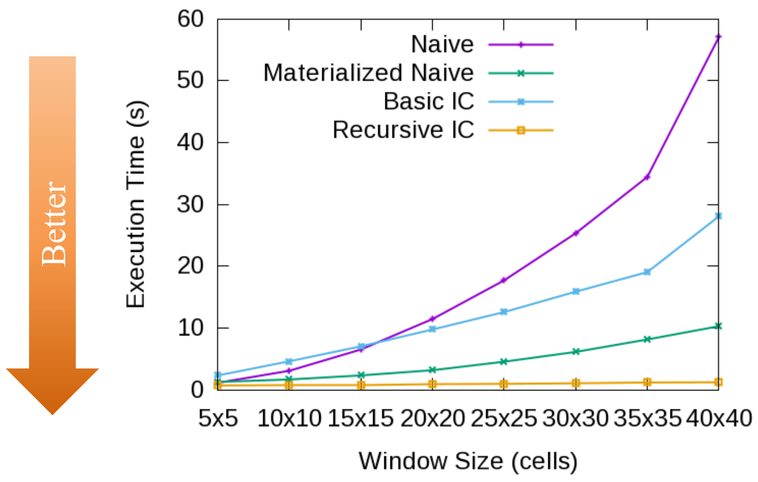




## Fast Window Aggregate on Array Database by Recursive Incremental Computation [4]

**Query: select max(v) from *arr* grouping by window (2,3)**

| | | | | |
|---|---|---|---|---|
| 4 | 7 | 3 | 1 | 8 |
| 5 | 2 | 6 | 2 | 2 |
| 3 | 9 | 3 | 2 | 4 |
| 7 | 7 | 8 | 2 | 6 |

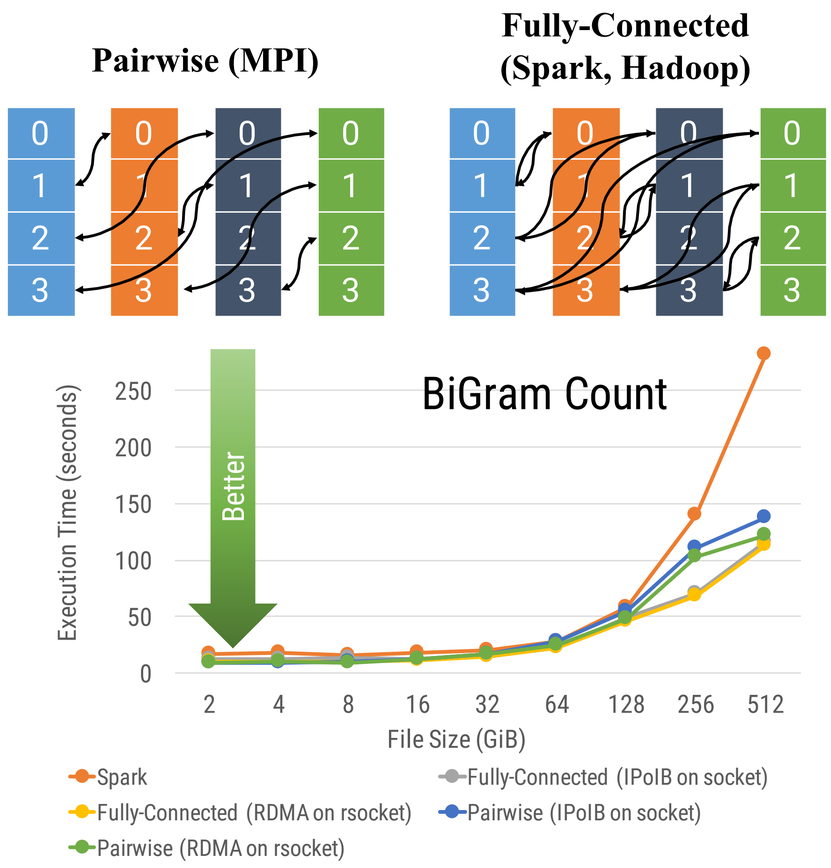
| | | | | |
|---|---|---|---|---|
| 7 | 7 | 8 | 8 | 8 |
| 9 | 9 | 6 | 4 | 4 |
| 9 | 9 | 8 | 6 | 6 |
| 7 | 7 | 8 | 6 | 6 |




We propose a method that exploits the scheme of incremental computation to accelerate the execution of window aggregates. Our proposed recursive incremental computation method completely eliminates all redundant computation, and it is fully implemented in SciDB. It improved performance by a factor of 10 on an earth science benchmark and by a factor of 64 on synthetic workloads with a certain data setting when compared with SciDB's built-in window operator.

**Acknowledgment**
This work is partially supported by JST CREST "System Software for Post Petascale Data Intensive Science", JST CREST "Extreme Big Data (EBD) Next Generation Big Data Infrastructure Technologies Towards Yottabyte/Year", JST CREST "Statistical Computational Cosmology with Big Astronomical Imaging Data", and KAKENHI #16K00150.

## On Exploring Efficient Shuffle Design for In-Memory MapReduce [5]




Shuffling, the inter-node data exchange phase of MapReduce, has been reported as the major bottleneck. We compared RDMA shuffling based on rsocket with the one based on IPoIB. We also compared our in-memory system with Apache Spark. Our system demonstrated performance improvement by a factor of 2.64 on BiGram Count as compared to Spark. We conclude that it is necessary to overlap map and shuffle phases to gain performance improvement.

**Reference**
[1] Osamu Tatebe, Kohei Hiraga, Noriyuki Soda, "Gfarm Grid File System," New Generation Computing, Ohmsha, Ltd. and Springer, Vol. 28, No. 3, pp.257-275, 2010.
[2] Gfarm File System, http://oss-tsukuba.org/en/software/gfarm
[3] Fuyumasa Takatsu, Kohei Hiraga, Osamu Tatebe, "Design of Object Storage Using OpenNVM for High-performance Distributed File System," Journal of Information Processing, Vol. 24, No. 5, pp. 824-833, 2016.
[4] Li Jiang, Hideyuki Kawashima, Osamu Tatebe, "Fast Window Aggregate on Array Database by Recursive Incremental Computation," The IEEE 12th International Conference on eScience, accepted.
[5] Harunobu Daikoku, Hideyuki Kawashima, Osamu Tatebe, "On Exploring Efficient Shuffle Design for In-Memory MapReduce," BeyondMR workshop, Article 6, 2016.