Performance Improvement of Gfarm Using InfiniBand RDMA

Shin Sasaki† Ryo Matsumiya† Kazushi Takahashi†‡ Yoshihiro Oyama†‡
†The University of Electro-Communications, ‡JST, CREST
{sasashin, r.matsumiya, kazushi, oyama}@ol.inf.uec.ac.jp

In the field of HPC, distributed file systems are widely used to store input-output data processed by applications. The performance of distributed file systems have significant impact on applications, and their performance improvement is in great demand. In this on-going work, we focus on InfiniBand, an interconnect widely used in the field of HPC, and propose a method that achieves high-throughput data transfer between client nodes and I/O servers. In the proposed method, data are transferred into page caches of client nodes by using InfiniBand remote direct memory access (RDMA). Our target distributed file system is Gfarm [1].

Gfarm [1] is a distributed file system designed for data sharing among multiple nodes and data-intensive computing. Gfarm consists of a single meta data server that manages file meta data, and multiple I/O servers that store data in their local file systems. Gfarm library is provided to access to data on Gfarm. In general, we mount Gfarm in user space with *gfarm2fs*, a FUSE-based [2] utility program. Figure 1 (a) describes the path of data read/write when using gfarm2fs. While gfarm2fs is a user-friendly and sophisticated program, it can cause performance degradation due to an overhead of context switches caused by FUSE.

To reduce this overhead, we have developed a loadable kernel module (Gfarm LKM) that mounts Gfarm in kernel space. We load the LKM and mount Gfarm using `mount` command. Figure 1 (b) describes the path of data read/write when using Gfarm LKM. Although the LKM reduces an overhead related to FUSE, data are still copied from an I/O buffer of Gfarm library into page caches.

In our proposed method, in order to reduce this useless data copy, data are directly transferred from I/O servers into page caches of client nodes by using InfiniBand RDMA. Figure 2 describes a RDMA_WRITE-based implementation and Figure 3 describes a RDMA_READ-based one. When a client node reads data on Gfarm, page caches of the client node are mapped to DMA regions, and then date requests are transferred from an I/O server by using RDMA_WRITE (Figure 2) or RDMA_READ (Figure 3). In the client node, all the above operations are performed in kernel space.

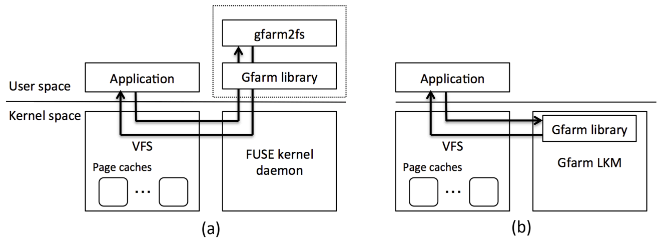

Fig. 1. The structures of (a) gfarm2fs and (b) Gfarm LKM

As a preliminary evaluation, we measured the throughput of sequential read under the IOR benchmark. In the evaluation, we used three nodes (a meta data server, an I/O server, and a client node) connected by InfiniBand QDR 4×. The specification of the three nodes are Intel Xeon CPU E5-2609, 64 GB main memory, SAS HDD 15,000 rpm × 2 (RAID 0), and CentOS 6.3 64bit.

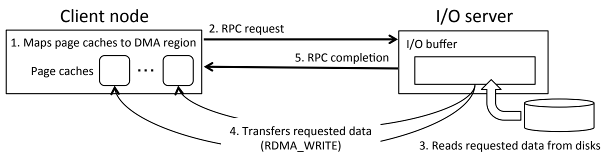
Client node
I/O server
1. Maps page caches to DMA region
Page caches
2. RPC request
5. RPC completion
I/O buffer
4. Transfers requested data (RDMA_WRITE)
3. Reads requested data from disks

Fig. 2. RDMA_WRITE-based implementation of the proposed method

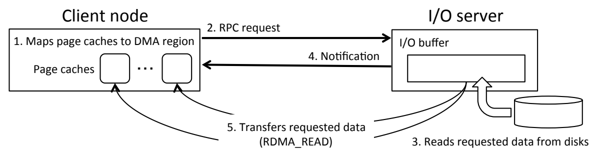
Client node
I/O server
1. Maps page caches to DMA region
Page caches
2. RPC request
4. Notification
I/O buffer
5. Transfers requested data (RDMA_READ)
3. Reads requested data from disks

Fig. 3. RDMA_READ-based implementation of the proposed method

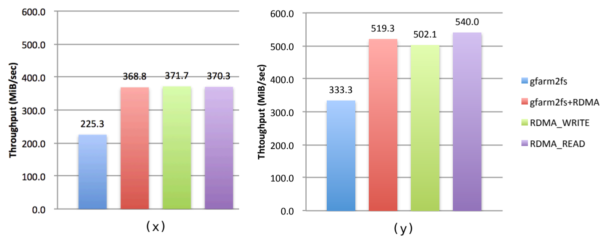

Fig. 4. Throughput of file read when files are not cached (x) and cached (y)

Figure 4 shows the throughput of reading a file (4GiB) sequentially. Figure 4 (x) is the result where an I/O server has no caches, and Figure 4 (y) is the result where an I/O server caches all requested data. gfarm2fs and gfarm2fs+RDMA represent original gfarm2fs and modified gfarm2fs that transfers data using InIfiniBand RDMA. RDMA_WRITE and RDMA_READ represent our RDMA_WRITE-based implementation (Figure 2) and RDMA_READ-based implementation (Figure 3) . As you can see in Figure 4 (y), the throughput of RDMA_READ slightly increases compared with gfam2fs+RDMA.

Our current implementations have not made full use of InfiniBand. However, we believe that the proposed method can achieve higher throughput. For future works, we plan to optimize the proposed method, and we are going to conduct further evaluation under real applications.

References

[1] O. Tatebe, K. Hiraga, and N. Soda, "Gfarm Grid File System," *New Generation Computing*, vol. 28, no. 3, pp. 257–275, 2010.
[2] E. Szeredi, "FUSE: Filesystem in userspace," http://fuse.sourceforge.net/.