Autonomous distributed filesystem for heterogeneous computing environment

Koji Watanabe (310125)

School of School of Information Science,
Japan Advanced Institute of Science and Technology

Feb.10, 2005

Keywords: Grid GridFTP multiple-parity distributed-storage reliability.

1 Purpose and Background

In recent year, Grid computing has become a key technology in areas such as high energy physics, human genome analysis. Many computers are connected to grid, so fault tolerant and allocation of efficient resources are indispensable technologies. Grid environment is constructed with various kinds of computers. Because of difference of these computers' reliability, we have to calculate reliability of grid in detail. Grid technology will play an important part in construct wide-area distributed storage environment for Petabyte scale data. In recent research for data storage system via Grid, prime object is placed in the preservation and fast transfer method of the data. Technology in order to guarantee reliability is not discussed in those researches. The only technology for fault tolerant is data replication. Data replication is good at low overhead and load balancing, but the total amount of data is tend to become large.

In this paper, we focus on the distributed file system which includes fault tolerant technology. Grid system in past research consists of many computers whose reliability is high, but ours is not guarantee high reliability. We are planning to use normal PC or WorkStation, so each of reliability of these computers is unstable. In order to use unstable computers, it is necessary to calculate detailed system reliability. We calculate reliability of each node by MTBF and MTTR. The master server, which is manages the grid, observes the uptime and the downtime of each nodes, and calculate MTBR and MTTR. We use Reed-Solomon code to avoid data loss from multiple disk failures. The master server determines the ability to tolerate how many failures depend on system reliability.

Using these methods, we pay attention to efficient disk usage and system reliability, we have developed distributed file system and evaluate of the systems' performance.

2 Proposed System

System has three kinds of computers. Those are "Master server", "Resource management server", "Storage node". Each computers are installed Globus Toolkit 2.2. Master server is a computer which splits data into some blocks, generates Reed-Solomon code, calculation reliability of system and rebuild of data. It is implemented by C and Perl. Resource management server is a computer which manages resources of grid such as free disk space or free memory size. This server runs MDS (Globus Metacomputing Directory Service) which is a part of globus services. Storage nodes are computers that store the data and redundancy data. It run GridFTP server program.

3 System Reliability

In order to guarantee reliability of system, master server generates redundant data which is called ”Multiple parity”. When storage is distributed among n nodes, ”n+1-parity”can recover from one device false. Also, ”n+m-parity”can recover from m devices false. Master server determines the number of m depend on system reliability. Our failure model is expressed by x-out-of-y parallel system. We classify storage nodes into three classes depend on its reliability. As for each class, representative’s value is allocated, and this value is used when system reliability is calculated. Because the calculation of the reliability of the parallel system that consists of computers that reliability is different is very complex, and explosion of calculation is occurred.

4 Performance of System

In this section, we evaluate two processing mode. One is ”own process mode” which is a mode to process master server job by itself. Another is ”External mode” which is a mode for load balancing. We construct the grid with 8 Sun Blade1500s and 1 Opteron Workstation. Each of computers is connected by Gigabit Ethernet. As a preliminary examination, we have tested improvement of IO performance by using striping data transfer; have measured overhead by handling Reed-Solomon codes, and bottleneck of nfs file system. The maximum parallel transfer rate was about 740Mbps. This result is reasonable. It is 800Mbps even if the maximum data transfer rate from the buffer to the buffer. In test of decoder, it took 58 seconds to recover from multiple failures.

Next, we measure performance to write and read data stored to grid system in own process mode. When system writes a data to grid, the system have three phases. At first, the master server determines which storage node will be used, and calculates the reliability of the system.

Next, the master server splits original data, and encodes multiple parities. At last, the master server transfer files to storage nodes. We examined 5 kinds of original data. These are 125MB, 250MB, 500MB, 1000MB and 2000MB. It was too short to detect termination of the process to operate the original file. There are two case when read performance is measured. One is normal read. The original data was recovered from distributed files correctly. Another is read data with some disk failures. It took more time to recover and decode files than without failures. But, the original data was recovered from distributed files correctly, too. It was verified by comparing checksum of the recovered file with the checksum of the original data.

In addition, in order to keep data safe, we have implemented a program to check healthiness of data. It is vivificated that data with some disk failure are recovered correctly. If any troubles are found, the program goes into recovery mode. The trouble said to here, includes the troubles of network, the down of the host and the erasure of distributed files. It is simple to rebuild the original data. At first, program checks status of distributed files. If there is any problem, the checking program runs the program to decode, and create new temporal file. These temporal files are dividing and encode to new parity, and transfer to grid.

5 Results

In this research, when we construct distributed storage system into grid, it is good method to keep high reliability and efficiency disk usage than replica system by using multiple parity and calculation system reliability by MTBF and MTTR of each node. For example, storage is distributed among 15nodes which includes 2 parity nodes, proposed system can save disk space 37% than the replica system. In external mode, the ability of file transfer is not so high, because of low performance of nfs. In current version of the system, we can only calculate the availability of the system. The availability is the probability that the system can not accessible. Another indicator of reliability is Mean Time To Data Loss (MTTDL). It is time until the data could not be read completely. In the next version of the system, we will implement the function to calculate MTTDL in order to keep the system higher reliability.