Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Server Integration for Data Storage & Backup via Cloud Computing: A Review

Mayur S.Agrawal[1], Nilesh Vani[2]
[1]*Computer Engineering, GF's G.C.O.E, Jalgaon*
[2]*Computer Engineering, GF's G.C.O.E. Jalgaon*

Abstract—Cloud computing has emerged as a popular solution to provide for storage of data by server integration. Today some applications like database, media etc deals with the large amount of data having higher I/O data demands. In order to improve performance of these applications can use parallel file systems. PVFS2 is a free parallel file system developed by a multi-institution team of parallel I/O, networking and storage experts. In this paper we present design & implementation for the server integration for able to store and back up data through using remote servers that can be accessed through the Internet. The implementation aims to increase the availability of data and reduction in loss of information.

Keywords- Cloud, Storage, backup, File System, Cluster, PVFS2

I. INTRODUCTION

Cloud computing can be defined as that service (software, platform or infrastructure) located on the Internet and is accessed from a mobile device or desktop computer, giving users a wide variety of applications (databases, middle office software, storage, etc.). The different cloud deployment strategies public cloud services are characterized as being available to clients from a third party service provider via the Internet. The term "public" does not always mean free, even though it can be free or fairly inexpensive to use. A public cloud does not mean that a user's data is publicly visible; public cloud vendors typically provide an access control mechanism for their users. The difference between a private cloud and a public cloud is that in a private cloud-based service, data and processes are managed within the organization without the restrictions of network bandwidth, security exposures and legal requirements that using public cloud services might entail. In addition, private cloud services offer the provider and the user greater control of the cloud infrastructure improving security and resiliency because user access and the networks used are restricted and designated. A hybrid cloud is a combination of a public and private cloud that interoperates.

A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on Service Level Agreements (SLAs) established through negotiation between the service provider and consumers" [1] Major goal of this work is to develop such application that serves as storage and backup through remote servers via cloud computing, we selected as a file system PVFS2 for our work; to be free and open source, we have the freedom to use. It also offers availability, flexibility and overall great performance when writing to or reading from the I/O servers. Cloud storage is the data is backed up and stored by using an internet connection on remote servers.

II. RELATED WORK

Cloud Computing and cloud storage have become the preferred method for delivering information and online functionality. While some cloud services focus on providing consumers a wide range of services and functionalities. Others provide cloud storage to consumers for free or charge some type

of subscription-based fee.

A. **Dropbox**
B. **Google Drive**
C. **Windows Azure**
D. **Cloud File System Oracle**
E. **Panzura CloudFS file System**

Dropbox [2] gives users the capability of sharing entire folders with other Dropbox account users, which allows updates to be viewable by all collaborators. Users can download shared documents directly from Dropbox's web interface without having to install the Dropbox desktop client. Storing files in the Dropbox "Public" folder allows links to files to be sent to Dropbox and non-Dropbox users; however non-Dropbox link recipients must download the file to access/edit it, and any changes or revisions made to the file by the link-recipients will not be reflected in the Dropbox version of the file. Dropbox allows users to create a special folder on each of their computers, which Dropbox then synchronizes so that it appears to be the same folder (with the same contents) regardless of which computer is used to view it. In Another approach of Google drive [3] Users of Google Drive documents must have a Google Drive account. All updates and editing by collaborators will be synced to Google Drive. For documents that you have permission to access, you can receive notifications when changes are made. You can share files with people by sending them a link to your file [3]. Windows Azure [4], is an open cloud platform in a global network of data centers run by Microsoft. Let's compile applications in any language, tool or framework for integrate your public cloud applications. It is important to know how to actually manage the backup and storage of files within the input / output. Currently there are file systems for cloud environments [5,6]. Panzura CloudFS file System [5] is a file system developed to provide integration with cloud and NAS environments. It offers functionality transparent to users, as everyone can see the same file from any location. It also allows data sharing, without having to delete the original file. Cloud File System Oracle [6] is a file system for private cloud environments, designed to manage general purpose file store outside of an oracle database across multiple operative system platforms with one management interface. Too it´s tightly integrated with the automatic storage management features of the oracle database. The Parallel Virtual File System project is a multi-institution collaborative effort to design and implement a production parallel file system for HPC applications [7]

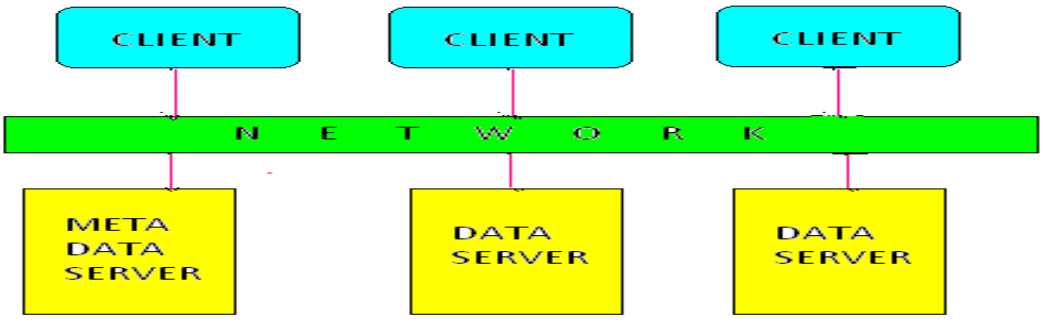

Figure 1: PVFS Structure

III. LITURATURE SURVEY

Parallel Virtual File System

The Parallel Virtual File System (PVFS) is an open source parallel file system. A parallel file system is a type of distributed file system that distributes file data across multiple servers and provides for concurrent access by multiple tasks of a parallel application. PVFS was designed for use in large

scale cluster computing. PVFS focuses on high performance access to large data sets. It consists of a server process and a client library, both of which are written entirely of user-level code. A Linux kernel module and pvfs-client process allow the file system to be mounted and used with standard utilities. The client library provides for high performance access via the message passing interface (MPI). The second PVFS version, PVFS2, is an extension of the first one that improves modularity and flexibility among modules, and provides a strong integration with MPI-IO. The components of a distributed file in PVFS are: N chunks of file data, one metafile with file attributes, and one directory entry. PVFS stripes a single file across the I/O or data servers. Each file will have N datafiles, one on each data server, with a chunk (several stripes) of the data in the file. The 64 bits descriptor used to refer a datafile is a datahandle. The list of all the datahandle of a file and its attributes are maintained in a metafile on a metadata server. Metafile has also a metahandle that represents it. The parent directory of the file can be on another metadata server. PVFS2 consisting of three different components as it is shown in Fig. 1

1. Clients
2. Data servers
3. Meta Data servers

The PVFS server runs as a process on a node designated as an I/O node. I/O nodes are often dedicated nodes but can be regular nodes that run application tasks as well. The PVFS server usually runs as root, but can be run as a user if preferred. Each server can manage multiple distinct file systems and is designated to run as a metadata server, data server, or both. All configuration is controlled by a configuration file specified on the command line, and all servers managing a given file system use the same configuration file. The server receives requests over the network, carries out the request which may involve disk I/O and responds back to the original requester. Requests normally come from client nodes running application tasks but can come from other servers. The server is composed of the request processor, the job layer, Trove, BMI, and flow layers. In a cluster using PVFS nodes are designated as one or more of: client, data server, metadata server. Data servers hold file data, metadata servers hold metadata include stat-info, attributes, and datafile-handles as well as directory-entries. Clients run applications that utilize the file system by sending requests to the servers over the network. PVFS has an object based design, which is to say all PVFS server requests involved objects called dataspaces. A dataspace can be used to hold file data, file metadata, directory metadata, directory entries, or symbolic links. Every dataspace in a file system has a unique handle. Any client or server can look up which server holds the dataspace based on the handle. A dataspace has two components: a bytestream and a set of key/value pairs. The bytestream is an ordered sequence of bytes, typically used to hold file data, and the key/value pairs are typically used to hold metadata. PVFS is designed so that a client can access a server for metadata once, and then can access the data servers without further interaction with the metadata servers. This removes a critical bottleneck from the system and allows much greater performance. PVFS uses a networking layer named BMI which provides a non-blocking message interface designed specifically for file systems. BMI has multiple implementation modules for a number of different networks used in high performance computing including TCP/IP, Myrinet, Infiniband, and Portals. [8] PVFS clients and servers run at user level. Kernel modifications are not needed. There is an optional kernel module that allows a PVFS file system to be mounted like any other file system, or programs can link directly to a user interface such as MPI-IO or a Posix-like interface. This features makes PVFS easy to install and less prone to causing system crashes

Sever Integration for storage and backup-

We integrate server using private cloud model for backup and data storage will develop and application is implemented in such way that automatically synchronizes all information backed up or stored by the user in the virtual folder to the cloud. In the cloud environment, physical folders were located on a mounted disk space PVFS2 servers .PVFS2 stripes files over the multiple data servers.

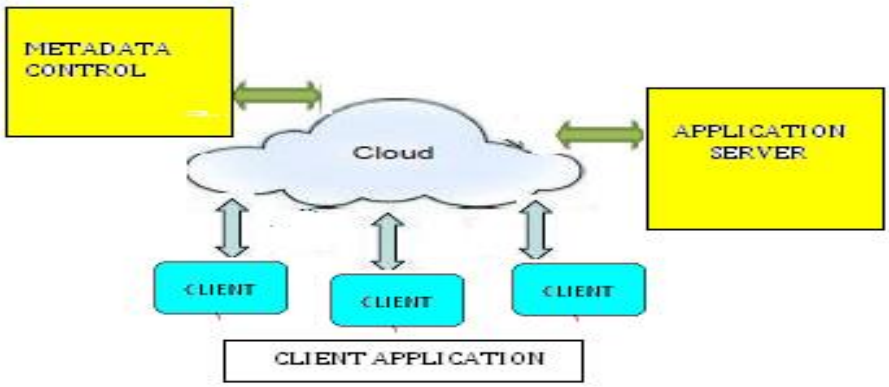

Figure 2: Client Server Application

A client application is made in JAVA Communication and information sent between applications will be made through Sockets using the classes in the java.net package using TCP. files will be serialized and sent as strings of bytes, tentatively such byte strings encrypted with specific algorithm. The server application will contain one thread per user, provided by the Thread class in Java, which through Sockets application will contact the client with whom share data and metadata, depending on the rules described in the previous section. The application server will receive data and metadata as strings of bytes encrypted the decrypt, write, replace or delete as applicable, and also send data and metadata in encrypted byte streams to the client application. This application will also feature a web application which can view and download the files in the directory of each user, and simple steps can make the user account, and update personal information and password. Metadata that will keep are: file name, size, path, last date modified, deleted mark, this information is stored in a MySQL as the figure2.

IV. CONCLUSION

We propose a design & implementation for the server integration to backup and store data on a private cloud using PVFS2 file system for storage data in order to increase the performance of applications that requires high I/O data demands. PVFS2 is file system which is open source this option allows input / output parallel, so that will reduce the access times to data. On the client-side, a application is developed that allows data transfer fast and simple way. The advantages of this implementation are that it can reuse existing infrastructure (servers, cluster, and other devices) that reduces the cost.

REFERENCES

[1] Rajkumar Buyya, Chee Shin Yeo, and Srikumar, Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities". CoRR, (abs/0808.3558), 2008.
[2] Dropbox: https://www.dropbox.com
[3] https://www.google.com/intl/en_US/drive/start/index.html
[4] Windows Azure: http://www.windowsazure.com/es-es/
[5] Oracle Cloud File System (White paper): http://www.oracle.com/us/products/database/cloud-file-system/overview/index.html
[6] Panzura CloudFS file system (White paper) http://panzura.com/products/global-file-system/
[7] Samuel Lang, Philip Carns, Robert Latham, Robert Ross, Kevin Harms, William Allcock, "I/O Performance Challenges at Leadership Scale," Proceedings of Supercomputing, 2009
[8] Philip H. Carns, Walter B. III, Robert Ross, Pete Wyckoff, "BMI: a network abstraction layer for parallel I/O," Proceedings of IPDPS '05, 2005

