**BFC: High-Performance Distributed Big-File Cloud Storage Based On Key-Value Store**

**ABSTRACT:**

Nowadays, cloud-based storage services are rapidly growing and becoming an emerging trend in data storage field. There are many problems when designing an efficient storage engine for cloud-based systems with some requirements such as big-file processing, lightweight meta-data, low latency, parallel I/O, deduplication, distributed, high scalability. Key-value stores played an important role and showed many advantages when solving those problems. This paper presents about Big File Cloud (BFC) with its algorithms and architecture to handle most of problems in a big-file cloud storage system based on key value store. It is done by proposing low-complicated, fixed-size meta-data design, which supports fast and highly-concurrent, distributed file I/O, several algorithms for resumable upload, download and simple data deduplication method for static data. This research applied the advantages of ZDB - an in-house key value store which was optimized with auto-increment integer keys for solving big-file storage problems efficiently. The results can be used for building scalable distributed data cloud storage that support big-file with size up to several terabytes.

**Keywords**— Cloud Storage, Key-Value, NoSQL, Big File, Distributed Storage

**EXISISTING SYSTEM:**

People use cloud storage for the daily demands, for example backing-up data, sharing file to their friends via social networks such as Face book, Zing Me . Users also probably upload data from many different types of devices such as computer, mobile phone or tablet. After that, they can download or share them to others. System load in cloud storage is usually really heavy.  
Thus, to guarantee a good quality of service for users, the system has to face many difficult problems and requirements.

**Disadvantages:**

* Storing, retrieving and managing big-files in the system efficiently.
* Parallel and resumable uploading and downloading.
* Data deduplication to reduce the waste of storage space caused by storing the same static data from different users.

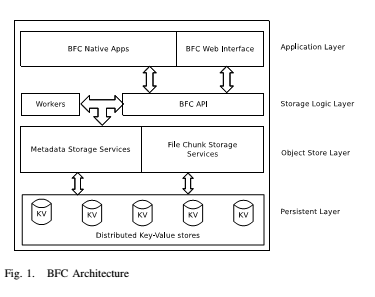
**PROPOSED SYSTEM:**

A common method for solving these problems which is used in many Distributed File Systems and Cloud Storages is splitting big file to multiple smaller chunks, storing them on disks or distributed nodes and then managing them using a meta-data system. Storing chunks and meta-data efficiently and designing a lightweight meta-data are significant problems that cloud storage providers have to face. After a long time of investigating, we realized that current cloud storage services have a complex meta-data system; at least the size of metadata is linear to the file size for every file. Therefore, the space complexity of these meta-data system is O (n) and it is not well scalable for big-file. In this research, we propose new big-file cloud storage architecture and a better solution to reduce the space complexity of meta-data.

**Advantages:**

* Propose a light-weight meta-data design for big file. Very file has nearly the same size of meta-data.
* Propose a logical contiguous chunk-id of chunk collection of files. That makes it easier to distribute data and scale-out the storage system.
* Bring the advantages of key-value store into big-file data store which is not default supported for big-value. ZDB is used for supporting sequential write, small memory-index overhead.

**BIGFILE CLOUD ARCHITECTURE:**



**MODULES:**

1. **Application Layer**
2. **Storage Logical Layer**:
3. **Object Store Layer**
4. **Persistent Layer** :

**Module description:**

**Application Layer**: It consists of native software on desktop computers, mobile devices and  
web-interface, which allow user to upload, download and share their own files.

**Storage Logical Layer**: it consisted of many queuing services and worker services, ID-Generator services and all logical API for Cloud Storage System. This layer implements business logic part in BFC.

**Object Store Layer**: It contains many distributed backend services. Two important services of Object Store Layer are FileInfoService and ChunkStoreService. FileInfoService stores information of files. Y-value store mapping data from fileID to FileInfo structure. ChunkStoreService stores data chunks which are created by splitting from the original  
files that user uploaded.

**Persistent Layer**: it based on ZDB key-value store. There are many ZDB instances  
which are deployed as a distributed service and can be scaled when data growing.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP/7.
* Coding Language : JAVA