Chapter 39
A Cloud-Aware Distributed Object Storage System to Retrieve Large Data via HTML5-Enabled Web Browsers

Ahmet Artu Yıldırım
Utah State University, USA

Dan Watson
Utah State University, USA

ABSTRACT

Major Internet services are required to process a tremendous amount of data at real time. As we put these services under the magnifying glass, it's seen that distributed object storage systems play an important role at back-end in achieving this success. In this chapter, overall information of the current state-of –the-art storage systems are given which are used for reliable, high performance and scalable storage needs in data centers and cloud. Then, an experimental distributed object storage system (CADOS) is introduced for retrieving large data, such as hundreds of megabytes, efficiently through HTML5-enabled web browsers over big data – terabytes of data – in cloud infrastructure. The objective of the system is to minimize latency and propose a scalable storage system on the cloud using a thin RESTful web service and modern HTML5 capabilities.

INTRODUCTION

With the advent of the Internet, we have faced with a need to manage, store, transmit and process big data in an efficient fashion to create value for all concerned. There have been attempts to alleviate the problems emerged due to the characteristics of big data in high-performance storage systems that have existed for years such as: Distributed file systems: e.g., NFS (Pawlowski et al., 2000), Ceph (Weil et al., 2006), XtreemFS (Hupfeld et al., 2008) and Google File System (Ghemawat et al., 2003); Grid file

DOI: 10.4018/978-1-4666-9840-6.ch039

systems: GridFTP (Allcock et al., 2005) and recently object-oriented approach to the storage systems (Factor et al., 2005).

As an emerging computing paradigm, cloud computing refers to leasing of hardware resources as well as applications as services over the Internet in an on-demand fashion. Cloud computing offers relatively low operating costs that the cloud user no longer needs to provision hardwares according to the predicted peak load (Zhang et al., 2010) via on-demand resource provisioning that comes with pay-as-you-go business model. In realization of this elasticity, virtualization is of significant importantance where hypervisors run virtual machines (VMs) and share the hardware resources (e.g. CPU, storage, memory) between them on the host machine. This computing paradigm provides a secure, isolated environment that operational errors or malicious activity occurred in one VM do not affect directly the execution of another VM on the same host. Virtualization technology also enables the cloud providers to further cut the spendings through live migration of VMs to underutilized physical machines without downtime in a short time (Clark et al., 2005), thus, maximize resource utilization.

The notion of an object in the context of storage is a new paradigm introduced in (Gibson et al., 1997). An object is a smallest storage unit that contains data and attributes (user-level or system-level). Contrary to the block-oriented operations that perform on the block level, object storage provides the user higher-level of abstraction layer to create, delete and manipulate objects (Factor et al., 2005). Backends of most object storage systems maximize throughput by means of caching and distributing the load over multiple storage servers, and ensuring fault-tolerance by file replication on data nodes. Thus, they share similar characteristics with most high-performance data management systems, such as fault-tolerance and scalability.

Modern web browsers have started to come with contemporary APIs with the introduction of the fifth revision of the HTML standard (HTML5) to enable complex web applications that provide a richer user experience. However, despite a need on client-side, web applications still are not taking advantage of HTML5 to deal with big data. In regards to the server-side, object storage systems are complex to build and to manage its infrastructure.

We introduce an experimental distributed object storage system for retrieving relatively bigger data, such as hundreds of megabytes, efficiently through HTML5-enabled web browsers over big data – terabytes of data – using an existing online cloud object storage system, Amazon S3, to transcend some of the limitations of online storage systems for storing big data and to address further enhancements.

Existing systems exhibit the capability of managing high volumes of data, retrieving larger size resources from a single storage server might cause an inefficient I/O due to unparalleled data transfer at the client-side and underutilized network bandwidth. The main objective of the implemented system is to minimize latency via data striping techniques and propose a scalable object storage system on top of an existing cloud-based object storage system. For the client side, we implemented a Java Script library that spawns a set of web workers – which is introduced with HTML5 to create separate execution streams on web browsers – to retrieve the data chunks from the storage system in parallel. We aim to increase the data read rates on the web browser by utilizing full Internet bandwidth. Our approach is also capable of handling data loss by automatically backing up the data on a geographically distinct data center. The proposed distributed object storage system handles a common error gracefully, such as if a disaster takes place in the data center that might result in data inaccessibility, the implemented client detects this issue and then starts retrieving the data from the secondary data center. We discuss advantages and disadvantages of using the proposed model over existing paradigms in the chapter.