Computers in Industry 100 (2018) 21–30

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

WebGlusterFS: A web-based administration tool for GlusterFS with resource assignment for various storage demands

Qiuming Luo*, Cuiping Zhu, Gang Liu, Rui Mao

College of Computer Science and Software Engineering, Shenzhen University, China

ARTICLE INFO

Keywords:
GlusterFS
Resource allocation
DFS

ABSTRACT

Facing the complex tasks involves making decisions about assignment of workloads to storage backends as well as dynamic and timely adjustment according to the storage demands in Cloud and Big-data environment, an administration tool for GlusterFS, WebGlusterFS, is presented in this article to ease the management and help to assign the storage resource. WebGlusterFS is a web-based tool designed to substitute the command line console manager of GlusterFS and provides an interface for auto-assignment module to build volumes from heterogeneous backend devices. A simple demo module is also implemented to show how various storage demands are fulfilled by building the volumes from properly matched storage resource with minimum cost. The characteristics of underlying storage resource are obtained by benchmarking and used to make the assignment decision. WebGlusterFS setups a base framework for workload aware storage platform for large scale computing environments.

1. Introduction

With the rapid development of information technology, enormou volumes of data or called "big data" is being generated by many enterprises at all time. Scientific applications, weather forecasting, researches, hospitals, gene information processing and military services are few such major contributors. The need to provide efficient, easy to use solutions has become one of the main issues for these types of computations. The prosperous solution to this issue is the use of Distributed File Systems (DFS) or cloud storage.

There are a number of opensource solutions like GlusterFS, Ceph [3], HDFS [4], Swift [5] and vendor specific solutions like EMC ViPR [6], NetApp ONTAP [7], IBM Virtual Storage [8], etc. All these DFSs are qualified to the basic requirements of cloud storage. The most typical feature of cloud storage involves SAAS (storage as a service) [2], which makes the providing of storage as a service to the users. So, the system need to provide administrators and users with convenient operating and maintaining environment. Besides the easy to manage, there is a need to auto-configure and reconfigure the storage resource easily to provide the storage serve to meet the demand of on-going applications.

Storage administration and data management are challenging and expensive tasks [16], particularly in cloud and Big Data environments where resources are shared among multiple workloads and accessed by various patterns [9,10]. One major and complex portion of these tasks involves making decisions about assignment of workloads to storage backend as well as dynamic and timely adjustment according to changing demands in cloud environments [11]. The assignment decision can be made based on the detailed description of workload characteristics and the performance factors of the underlying storage resource. There are all beyond the scope of these DFSs. So, we need an enhancement to cooperate the workload classification, backend characterization and the resource assignment of the exist DFSs to handle the variety of storage demands from Big-data applications.

In this article, a web-based management tool for GlusterFS is designed and implemented to help managing the storage resource and provide the possibility to make the resource decision by considering the workload requirements and backend characterization. GlusterFS is a scalable open source parallel file system that offers a global namespace, distributed front end, and capable of scaling to hundreds of petabytes without difficulty. It also offers extraordinary cost advantages benefits that are unmatched in the industry. The advantages that make it an ideal cloud storage are as follows [1]: 1) High scalable. Cloud storage can be expanded according to the demand of applications. 2) High reliable and available. Cloud storage will automatically back up the data to software/hardware failures and capable of disaster recovery. 3) Resource controllable. It is capable of controlling the access permissions of resources. 4) High utilization. Cloud storage can consolidate all the storage resources and provide a unified access interface to users. 5) Cost effective. The use of cloud storage can dramatically reduce the cost to running a data center for enterprises, and reduce the need for

* Corresponding author.
E-mail addresses: lqm@szu.edu.cn (Q. Luo), 894662293@qq.com (C. Zhu), gliu@szu.edu.cn (G. Liu), mao@szu.edu.cn (R. Mao).

https://doi.org/10.1016/j.compind.2018.04.001
Received 31 May 2017; Received in revised form 2 April 2018; Accepted 4 April 2018
0166-3615/

removable storage devices and efficiently lower the cost of individual users and businesses. GlusterFS has indeed a large user base both in HPC computing farms, and in several Cloud computing facilities. It supports access to storage both in terms of POSIX file-system and via a REST gateway for object storage support. It is worth simplifying the management and enable the workload awareness.

The rest of the paper is organized as follows. Section 2 presents the background of the key challenges and the system architecture of GlusterFS. Section 3 gives the details of designing and implementing the web-based management tool of GlusterFS to ease the management. A scheme is figured out to provide an interface to create volumes based on the workload classification and the performance parameters. Section 4 details two algorithms to assign resource with lowest cost. Section 5 shows some the prototype of this system and the experiment results. We conclude in Section 6.

2. Background

2.1. The heterogeneous distributed storage systems

The heterogeneity of distributed storage systems relies both in hardware and software.

The nodes of the distributed storage system are likely to become heterogeneous which means different specifications [19]. The first case is a system expansion. In general, specifications of new storage nodes for the expansion may be different from storage nodes used in the existing system. Because computer technologies may evolve during the long time from the day of system installation to the day of system expansion. Therefore, aged distributed storage systems tend to become heterogeneous. The second case is the reuse of computer resources. When a storage administrator composes a distributed storage system of unused computer resources which were used for another systems before. In this case, it must be difficult to collect many computers with same specifications. Therefore, reused distributed storage systems tend to become heterogeneous.

Each distributed storage system can be characterized by its strengths and weaknesses owing to the underlying algorithm taking the decisions. The experiments show Ceph performs better than HDFS when the average file size is small (like a transactional log workload) though HDFS tends to perform better when the average workload size is large [20]. Also, the classes of workloads can be pretty diverse. For instance, the workload can be transactional characterized by small files with high IOPS or photo stream characterized by mid-range file size with average IOPS. A workload can be hybrid of different workload classes, too. For instance, a web server may serve photos to users and at the same time have logging daemon running in background.

2.2. Key challenges

The key challenges to implement a workload aware storage with heterogeneous platform involves workload characterization, backend characterization and backend selection [11].

2.2.1. Workload characterization

In order to provide the storage serve to meet the demand of ongoing applications, the storage system should be workload-aware. It needs to have the mechanisms to characterize storage workloads. This is not a trivial task. Current approaches usually rely on pre-defined policies [26] or offline storage workload trace analysis [27] and classification [29]. Application-specific workload characterization efforts exist for web servers [21,22] decision support systems [23], and personal computers [25]. The WASP classified the workload by access pattern, read/write bias, latency sensitivity, bandwidth, IOPS, data unit size, dataset size, dataset change rate, active dataset size and avg. IO size.

2.2.2. Backend characterization

Some works attempted to model a storage backend and its components to characterize its capabilities. However, this analytical approach proved to be too complex, and it is practical to adopt an empirical approach [11]. The expected performance of a given backend under specific workload can be captured and provided in many ways. The one possible manner is to run offline performance tests for a specific backend under a variety of workloads and then represent the characterization of a backend by a function of workload and backend configuration. For example, IOzone can run the test for patterns of read, write, re-read, re-write, read backwards, and read strided, etc., for different block sizes and file sizes.

2.2.3. Backend selection and configuration

One simple way is to pre-define a mapping between a set of workload classes and a set of backends. When creating the dataset, the application provides some parameters such as storage type (object, file, block), expected size, requested durability to describe the dataset characteristics and provides the workload class, the workload class maps to the preferred backend using a simple table [11,24]. By establishing a function that estimates the required storage backend configuration needed to fulfill the expected performance of a given workload, a configuration that specifies the amount of resources (e.g., number of nodes, %CPU, RAM, storage space) required to satisfy that workload. HetStore proposed a solution to optimal IO workload assignment using statistical modelling to estimate measures of performance such as Throughput, IOPS, et al. The proposed system uses support vector regression to estimate the performance of individual IO workloads on each available DFS systems for optimal assignment and maximized aggregate bandwidth [17].

Even with only one type of backend DFS on heterogeneous hardware, the different configurations result in various performance. A simple guideline for data placement in heterogeneous distributed storage systems is provided to prevent the performance degradation cause by the disks with lower specifications in a client side [19]. This simple solution for this issue is tried to rebalance the tasks in proportion to the specifications of nodes. It ignores the variety of servers, workloads and the types of GlusterFS volumes, which makes it incapable for the problem of optimal assignment for backend storage for various storage demands.

The previous works mostly focus on improving the performance by assigning the workload carefully, while this work tries to provide the need QoS with lowest cost. Janus improves the hit rate of flash tier in tiered storage system by allocating flash between different workloads [26]. IO Tetris try to achieve the maximal resource utilization and minimal performance penalties [13]. GReM uses a global SSD resource management scheme to allocate SSDs to heterogeneous VMs and improves the overall IO hit rate for SSD [14]. ECStor allocates storage space according to the role of users and gets a better performance by achieving good load balance [1].

By associating a cost to each of those resources, the total cost of servicing a given workload by each of the available backends can be determined. The selection of the backend is done by selecting the backend with minimal cost while providing the requested QoS.

2.3. GlusterFS

GlusterFS is a scalable open source clustered file system that offers a global namespace, distributed front end, and scales to hundreds of petabytes without difficulty. It also offers great cost advantages benefits that are prominent in the industry. GlusterFS gives users the ability to deploy scale-out, virtualized storage scaling from terabytes to petabytes in a centrally managed and commoditized pool of storage, which is available to users in a single mount point, making it simple for the user. GlusterFS aggregates various storage servers over Ethernet or Infiniband RDMA interconnect into one large parallel network file