基于Kubernetes的云原生海量数据存储系统设计与实现

申大为
北京华品博睿网络技术有限公司　北京　100028

摘　要：随着云技术的不断发展和普及，云计算中的数据量急剧增加，以及它在性能和稳定性上遇到的一些问题，本文介绍了一种新的基于Haystack的存储系统。对系统的容错和缓存进行了优化，使得它能够更好地适应本地的云计算服务，以适应数据的采集；存储和分析产业和日益增多的应用对文档的存取、读取和写入的要求越来越高。本系统采用了对象存储模式，以适应大量的数据存储，为企业提供了一个简洁的、统一的应用界面，通过文件缓存策略提高了资源的利用率。实验表明，与现有的主流对象存储和文件系统相比，这种内存系统在读取大于写入的情况下，具有更好的性能和稳定性。

关键词：Kubernetes；云原生海量数据；存储系统

Design and implementation of cloud native massive data storage system based on kubernetes

Dawei Shen
Beijing huapinborui Network Technology Co., Ltd., Beijing, 100028

Abstract: with the continuous development and popularization of cloud technology, the amount of data in cloud computing increases sharply, and some problems it encounters in performance and stability. This paper introduces a new storage system based on haystack. The fault tolerance and cache of the system are optimized, so that it can better adapt to local cloud computing services and data collection; The storage and analysis industry and increasing applications have higher and higher requirements for document access, reading and writing. The system adopts the object storage mode to adapt to a large amount of data storage, provides a concise and unified application interface for enterprises, and improves the utilization of resources through file caching strategy. Experiments show that compared with the existing mainstream object storage and file systems, this memory system has better performance and stability when the read is greater than the write.

Keywords: kubernetes; Cloud native massive data; storage system

云计算是未来网络发展的一个大趋势，其发展潜力是无法估计的。 云计算应用和云计算平台的流行，使得开发者可以把更多的时间放在开发和维护自己的应用上，而不必担心硬件体系结构；服务器资源和其他与商业无关的日常硬件维护工作。最近几年，越来越多的企业从传统的服务器体系结构向云计算转移，而我们也发现了很多从最初就打算在云计算中运行的商业。这些服务常常被称为“云本地”，意思是它们与云计算的高度融合。云本地业务经常比传统业务更全面和经常地使用微型服务；在云计算平台上，如灵活的计算和服务编排，这些技术可以使业务更加稳定和高效地运行。一些前沿的做法甚至试图采用像AlpineLinux这样的更轻量级的操作系统，以适应容器和虚拟化技术。

目前比较典型的云服务是移动网络时代的社会化服务，其中很多都是将云计算的优点发挥到极致，例如微博将阿里云作为其计算后端，建设大规模的混合云，在公共云平台上运行一些高频服务，以灵活地应对突发的热点事件。随着社交平台的使用人数越来越多，需要大量的计算和存储。这不仅要求系统的稳定性和灵活性，而且还需要先进的扩展和灾备能力，以保证系统在突

作者简介：申大为，1993年4月，男，满族，河北省秦皇岛市，本科，容器云研发工程师，现在主要从事kubernetes docker容器云相关研发工作。

发情况下仍然能够给用户带来更好的服务。Facebook在2010年发布了它的解决方案，即Haystack，以处理大量的图片和文档。Haystack是一种面向大量的文档储存的物件储存系统，它能很好的满足Facebook的档案储存要求。该软件利用一个文件区块将小型档案包装成一个叫做超级区块的实体卷，并将元数据缓存到记忆体中，以便更好地索引和查询。同时，它还支持集群部署，可以分散存储负荷，从而提高系统的可伸缩性。近年来，随着云计算的发展，为了更好地利用云平台带来的效率和稳定性，因特网呈现出一种全面的向云端迁移的趋势。但Haystack和它的大部分类似的实现，都是基于传统的硬件体系结构而设计的。

针对上述问题，本文设计了一种新型的存储器。该系统是以Haystack为基础的，但是它特别优化了诸如服务发现和自动容错等宏观计划，使得它更符合云技术的定义，更适用于云计算，更适用于云上，更适用于驱动数据密集型云业务的高效、稳定的存储，以及Haystack在磁盘上的性能瓶颈。下面我们把它叫做Kubestorage，意思是在Kubernetes上运行的一个存储系统。

一、有关的工作

比云计算更早地进行了对象存储的研究。卡内基梅隆大学1996年就已经开展了有关的研究，并于1998年第一次提出了有关的概念和基本结构。自那以后，在不同的存储公司的支持下，物件储存继续繁荣。根据数据显示，1999至2013间，物件储存领域的风投资金已经超过三亿美金，这使得物件储存技术有了长足的发展，并且很快被广泛地运用于各种成熟的产品当中。美国亚马逊公司AWS（AmazonWebServices）推出亚马逊S3物件储存服务，这是2006年早期推出的一项开放服务。单是2012年，S3物件储存系统就新增了一兆多个物件，一年内更是达到二兆。另外，S3保证了在任何时候都可以存取这些数据，并且可以随意读取和写入。根据亚马逊的官方报道，任何S3中的任何一个文件都能在一秒钟内被读到1100，000次。另外一种比较出名的对象存储方案是Weil；Brandt；在2006 USENIX操作系统设计大会（OSDI2006）中，Miller、Long和Maltzahn提出了Ceph[9]。Ceph在软件方面做了很多的优化，可以用一般的硬件完成海量的文件存储，与IBM、EMC等公司的产品相比具有很好的兼容性。Facebook在2010年发表了一篇文章，介绍了一种更实际、更灵活的面向对象的储存模式，这种模式被称为Haystack。Haystack的特点是把小型的档案整合成一个叫做实体卷的档案，这样就可以维持较好的索引速度，同时可以储存很多小型档案。而文件索引则包括文件的基本信息、文件的偏移等元资料。另外，将文件索引读取到记忆体中，这样可以加快索引的速度，减轻磁盘的负荷。

二、库比斯特存储系统体系结构

库贝斯特储存系统包括三个部分：

1）一个目录服务器，它存储节点信息，映射文件到存储节点，自动管理节点和负载平衡。

2）用于文件存储、管理元数据以及存储一致性的存储服务器。

3）高速缓冲器，为提高性能，高速存取文件。

整个存储系统是在Kubernetes集群的基础上运行的，它是用Kubernetes的组态来实现的，Kubernetes提供管理、发现服务和修复服务。为了克服Haystack和其它存储方案的不足，Kubestorage做了如下改进：

1）利用Raft一致性算法[10]来支持多目录伺服器，使得目录伺服器更加稳定、可靠，并避免因单一的错误而导致整体储存系统崩溃。

2）与Haystack相比，Kubestorage没有将存储服务器直接暴露给外部的用户或者商业，它通过反向代理来实现对服务的读和写操作，从而提高了系统的安全，并且提高了目录服务器的总体控制力。考虑到虚拟交换机在Kubernetes中的带宽要比传统的网络连接带宽大，并且可以利用负载平衡器进行集群部署，所以相比于带宽瓶颈，相应的安全和便捷度的提高更加有意义。目录服务器为企业提供了一个单一的API，它包含了所有的数据读取和管理，从而简化了业务的发展，并避免了由于存储服务的暴露而造成的安全风险。

3）对所有文档进行周期性巡视，对长期闲置的资料进行自动压缩，节约存储空间.

4）采用多层快取机制，尽量减少因从磁碟中读取档案而导致的效能损耗。

5）将频繁存取的档案自动拷贝至多个存储器，以分散档案的读出压力，达到平衡负荷。另外，Kubernetes还利用了Kubernetes所提供的强大的容器和强大的功能，确保了在各种软件和硬件上的运行环境之间的一致性，并且以此为基础，实现了以下更多的功能：

1）对所有Kubestorage节点进行动态监控，当磁盘空间、CPU配额、内存空间不足时，将会自动进行扩展。

2）自动操作状态检验，包含一致性检验、有效性检验、延时检验、压力检测和硬体健康检验。

3）根据kube-apiserver和kube-dns实现了对存储群

集结构的动态更改，而不会出现服务中断，从而提高了Kubestorage的可用性。图1显示了Kubestorage的具体体系结构，它包括三个主要组成部分：

2.1 Catalog Server

Catalog Server主要负责下列工作：

1）在逻辑卷与实体卷之间进行管理。

2）管理Catalog Server层高速缓存。

3）在一个特定的文件访问数量急剧增加时，启动一个自动的文档冗余，并在多个存储器上创建一个文件的拷贝，从而提高了传输速度。

4）充当负载平衡器，对存储服务器执行逆向代理，并将统一API提供给外部服务。与此同时，Kubestorage的特性使得它更加适用于自动化的集群部署，并且提供了连贯和可靠的保证：

1）采用Raft一致性算法，保证了高的一致性和可靠性，避免了因单一故障而造成的整个存储系统崩溃，同时也可以通过外部服务向辅助结点传送。该方法能够确保数据的一致性和可靠性，从而在每一循环中只有一个目录服务器对用户的请求进行响应。

2）Kubernetes将为每个目录服务器提供三种默认标签：角色、状态和健康。比如，一个角色是kube-storage-directory，状态为leader或slave，它的运行状态有四种：运行、初始化、繁忙、离线。在每个目录伺服器都想了解其它目录伺服器的状况及数目时，可以将这三个选项做为选择器，向kube-apiserver查询目录伺服器清单，kube-apiserver会传回符合选取条件的伺服器。

3）所有的目录服务器共用一个数据库后台，以保证数据的一致性，避免由于读取/写入锁定而导致的系统的逻辑复杂性。数据库后台的类型可以根据需要随意地确定，但是Redis的内存数据库经常能够提供最好的性能。另外，目录伺服器将最新的档案读取要求存入资料库中，并将其存入档案对应的资料库，并定期读取最后一次巡查期间的档案存取数目。当文件频率超出设定阈值时，编目伺服器会利用储存伺服器的冗余写API，藉由在多台伺服器上建立档案的复本，以提高效能，而当下次巡检时发现已有档案存取少于临界值时，档案伺服器会使用冗余的删除API来移除已有的档案。

2.2 Store Server

存储服务器主要负责下列工作：

1）负责对其中所存储的文件、文件的元数据的管理、对文件的写和读的管理。

2）对储存伺服器层级快取进行管理。

3）执行数据压缩，解压缩和有效性检查。

4）将其状态定期报告给目录服务器，以实现基于统计数据的自动管理。以下是存储服务器技术的实现：

1）元数据和实体卷的基础存储与Haystack基本相同，但是Kubestorage对该文件进行了更多的支持，它将文件的状态保存在元数据中，在写时进行压缩、存入、读时进行解压。

2）由Kubernetes提供的永久卷将作为存储服务器的后台。该方法具有较高的抽象性，不依赖于特定的操作系统、存储体系结构和硬件，能够很好地兼容已有的存储方式，并且可以避免存储和计算之间的强烈耦合。为了让Kubestorage尽可能的简洁、冗余度和高可用性是通过真正的后台自动完成，从而可以充分发挥硬盘/网络存储产业在中期和长期的技术积累。

3）Kubestorage采用了更宽松的后端元数据结构，通过界面的形式来实现元数据的存储。另外，Kubestorage还将leveldb用作它的缺省数据库，以便于部署和使用。这种数据库在数据结构单一、数据容量小、数据结构单一的情况下，能够在较低的资源开销下，达到较好的性能，同时克服了常规文件数据库的频繁磁盘读取和写入、存储结构的复杂性。

4）各存储器服务器会定时执行自我检测服务，获得目前的存储器能力，同时读取和写入；文件的数量，硬件的健康状况，网络状况，以及其它需要的信息。自检服务让目录服务器，Kubernetes，甚至操作人员都能实时地掌握当前各节点的状况，并对其进行及时的管理。

2.3 Cache Server

缓存伺服器负责整合资料快取，因为档案必须储存，所以将Redis用作储存后端，从记忆体读取档案，确保执行效能。快取伺服器采用目录/储存两层快取的设计，以提高快取效能，最小化伺服器的反应速度。

三、性能测试

3.1 Beta环境

性能分析部分采用阿里云ECS服务器，采用IntelXeonPlatinum8163（8个虚拟内核）、2 GB的存储空间，并安装6 TB SSD硬盘作为测试用的数据盘（额定IOPS为25000，读取/写入为256 MB/s）。我们以EXT4、ZFS、SeaweedFS为标准，分别表示传统文件系统、现代文件系统以及传统的物件储存系统。SSD的云端硬盘会被格式为EXT4，ZFS文件系统，并且经过一系列的测试。这个云盘还会被格式化成EXT4格式，并在它上面配置Kubernetes（利用Flanel虚拟网络）的单个Kubernetes集

群，用于测试Kubestorage和SeaweedFS。

3.2检验方法

测试方式是：在一个测试伺服器上运行Golang开发的档案读取和写入API，并在LAN中使用其他配置相同的伺服器来远程存取这个API，以避免测试工具自身的影响。Kubestorage将在Kubernetes上建立，它由2个目录服务器，8个存储服务器和一个高速缓存服务器组成一个小型Kubestorage集群。SeaweedFS将会和Kubestorage一样，在Kubernetes上运行它的硬件配置和环境配置。

四、结语

综上所述，我们的Kubestorage存储系统可以更好地适应海量的小型文件，特别是在社会网络中，它具有“读”大于“写”的特点。以上的测试显示了Kubestorage在体系结构设计方面的强大和潜能，并且它的部署流程非常简单，只需要几个指令就可以通过预先设定的脚本来构建任何大小的Kubestorage存储集群。Kubestorage可直接使用，更适用于本地的云计算业务部署，为云端应用程序提供更快速、更稳定的存储解决方案。

参考文献：

[1]李俊江.基于Kubernetes的机器学习云平台设计与实现[D].南京邮电大学，2021.

[2]程仲汉，郑清安，陈淑珍.一种基于Kubernetes的Web应用部署与配置系统[J].成都信息工程大学学报，2021，36（05）：503-507.

[3]闫娟雅.基于Kubernetes的海量网络数据存储方法研究[J].电脑知识与技术，2021，17（27）：28-29.

[4]谢剑刚，肖小红，薛云兰.基于Kubernetes的数据库技术课程远程实验平台[J].信息技术与信息化，2021（07）：204-206.

[5]李梦超，史运涛，孙卫兵.基于Kubernetes及KubeEdge的社区燃气风险评估预警系统架构的探索[J].电子制作，2021（13）：22-26.

[6]张智.基于Kubernetes的批流融合数据处理支撑环境研究与实现[D].北方工业大学，2021.

[7]刘福鑫，李劲巍，王熠弘，李琳.基于Kubernetes的云原生海量数据存储系统设计与实现[J].计算机应用，2020，40（02）：547-552.

[8]涂俊英，李志敏.云计算下非结构化大数据存储系统设计[J].现代电子技术，2018，41（01）：173-177.

[9]李晓.基于MongoDB和Asio的传感器数据存储系统设计[J].电子技术与软件工程，2016（08）：200+253.

[10]马浩田.基于HBase的嵌套式数据存储系统设计与实现[D].浙江大学，2015.