

运载火箭试验大数据存储架构设计与应用

连彦泽^{✉1}, 李鹏程², 赵 雷², 司洪泉¹, 陈旭东¹

(1 中国运载火箭技术研究院 北京 100076

2 北京宇航系统工程研究所 北京 100076)

摘要: 运载火箭试验产生的数据量呈现出爆炸式增长, 主要特点表现为数据种类多、数据密度大、数据持续时间长。传统单机部署和基于关系型数据库与文件的系统架构的不足逐渐显现, 不同种类的数据不做区分存储, 存储和查询效率低, 数据无备份, 存在单点故障导致数据丢失的问题, 无法满足海量数据场景下的存储计算业务需求。利用大数据技术思想, 针对运载火箭存储计算业务的需求, 设计出一套运载火箭试验大数据存储架构, 并给出了各存储组件的存储模型设计方法。通过实际工程应用表明: 该架构具备良好的可靠性、可扩展性和可维护性, 是一种切实可行的大数据存储架构设计, 能够满足运载火箭试验数据的存储计算等业务需求。

关键词: 运载火箭; 大数据; 存储架构

中图分类号: V557

文献标识码: A

文章编号: CN11-1780(2022)06-0078-11

DOI: 10.12347/j.ycyk.20220509002

引用格式: 连彦泽, 李鹏程, 赵雷, 等. 运载火箭试验大数据存储架构设计与应用[J]. 遥测遥控, 2022, 43(6): 78–88.

Architecture design and application of big data storage for launch vehicle test

LIAN Yanze¹, LI Pengcheng², ZHAO Lei², SI Hongquan¹, CHEN Xudong¹

(1. China Academy of Launch Vehicle Technology, Beijing 100076, China;

2. Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China)

Abstract: The amount of data produced by launch vehicle test shows an explosive growth, which is mainly characterized by complex data types, large data density and long time duration. The system architecture shortcomings of traditional stand-alone deployment based on relational database and file are gradually emerging. Different types of data stored together, low efficiency of writing and query, no backup of data, and single point of failure could not meet the needs of storage and computing business in the context of massive launch vehicle test data field. Using the idea of big data technology and aiming at the need of launch vehicle test data storage and computing, a set of big data storage architecture for massive launch vehicle test data is proposed, and the storage model design method of each storage component is designed. The practical engineering application effect shows that the architecture has good reliability, scalability and maintainability. It is a practical big data storage architecture, which could meet the launch vehicle test data business needs of storage and calculation.

Key words: Launch vehicle; Big data; Storage architecture

DOI: 10.12347/j.ycyk.20220509002

Citation: LIAN Yanze, LI Pengcheng, ZHAO Lei, et al. Architecture design and application of big data storage for launch vehicle test[J]. Journal of Telemetry, Tracking and Command, 2022, 43(6): 78–88.

引 言

随着信息技术和网络技术的不断发展, 在数字化世界中产生的数据呈现爆炸式增长, 各科技类企业需要存储、处理及提供在线服务的数据越来越大。其中, 谷歌公司的搜索引擎为了提供更好的服务, 需要抓取并存储全世界网站的所有网页数据^[1], 同时需要按照网页之间的反向链接关系进行全量计算以确定网页的排序结果, 并提供能够实时响应的在线搜索服务。为了解决存储、计算和在线服务这三个核心

✉通信作者: 连彦泽 (lianyz@email.cn)

收稿日期: 2022-05-09 收修改稿日期: 2022-06-02

需求,谷歌公司在2003年、2004年及2006年,分别发表了GFS^[2]、MapReduce^[3]和Bigtable^[4]三篇论文。GFS解决了数据存储的问题。作为支持上千个节点的分布式文件系统,可以很容易地实现所有需要的数据的存储。MapReduce通过Map和Reduce函数对海量数据计算进行了一次抽象,使得不需要深入掌握分布式系统的开发与实现即可进行海量数据的并行处理。Bigtable使用GFS作为底层存储,通过集群的分片调度算法解决了大集群、机械硬盘下的高性能随机读写问题。正是受这三篇论文的影响,产生了开源的HDFS、MapReduce和HBase的大数据组件或框架^[5],使得大数据技术真正走向了普及。通过这些通用的技术框架和解决方案,越来越多的企业和工程师使用大数据技术有效地解决了各自面临的问题,大数据技术也得到了快速的演进和发展。

大数据技术的核心理念主要有三点:大数据技术是一种能够伸缩到一千台服务器以上的分布式数据处理集群的技术^[6],正是因为这种集群的伸缩性,才能够实现PB级别的数据处理和应用;集群是采用廉价的PC架构搭建的,正是因为这种低廉的硬件成本^[7],才使得大数据技术能够让越来越多的人使用;把整个集群抽象为一台计算机,通过各类大数据计算存储框架的封装和抽象,使得开发者像在一台计算上开发自己的代码,不用考虑分布式系统的可用性、数据一致性问题^[8]。

正是这三个核心技术理念,降低了大数据技术使用的门槛和难度,使得整个大数据技术生态繁荣发展^[9]。在分布式计算领域,为了不断优化MapReduce的性能,产生了Spark^[10]等分布式计算框架^[11]。在分布式存储领域,Facebook公司为了解决海量照片小文件存储的问题,研发了Haystack分布式文件存储系统,并发表了Finding a needle in Haystack^[12]的论文。SeaweedFS正是基于论文的开源实现^[13],被阿里等多家公司采用。为了解决海量监控数据及物联网等时序类数据的存储,产生了TiDB^[14]、TDengine^[15]等优秀的开源时序数据库产品,同时还产生了ClickHouse^[16]等OLAP类的数据库产品^[17]。

上述的各类计算、存储、分析类产品,均是基于大数据的核心技术理念开发出的分布式系统,得以承载越来越大的数据量和计算量。

在航天领域,数据是运载火箭试验重要的分析对象和组成部分,数据的完整性与正确性是评估运载火箭“是否可用、是否好用”的重要依据与指标,也是使用人员进行判断与决策的重要基础。传统存储方式是基于关系型数据库和文件进行存储,将结构化试验数据,如总装测试数据、模飞试验数据、结构强度试验数据和环境试验数据等存储于关系型数据库中,而非结构化试验数据或文件存储于文件系统中。随着运载火箭的不断发展,运载火箭试验产生的数据量呈现出爆炸式增长,传统单机部署和基于关系型数据库与文件的系统架构已无法满足海量数据存储计算的业务需求。如何将种类繁多、数量越来越多的运载火箭试验大数据通过一套大数据存储架构实现高效的存储计算,同时具备良好的可靠性、可扩展性和可维护性,是运载火箭试验数据存储和应用需要解决的核心问题。本文借鉴大数据的核心技术理念,整合应用开源先进的大数据存储、计算组件,设计数据模型,构建一套满足海量数据场景下存储计算业务需求的运载火箭试验大数据存储架构。

1 试验数据分析

运载火箭的研制需要经过设计、制造、试验等多个阶段后,才能形成最终产品并交付使用,其中试验阶段是整个研制过程的重要组成部分,试验阶段中产生的数据是火箭功能性能分析和评定最主要的依据^[18]。运载火箭试验数据是运载火箭研制过程中产生的重要数据资产,既包括文本、图片、音频、图像、日志等非结构化数据,也包括试验数值结果等时序类的结构化数据,试验数据涉及的火箭型号多、试验多,数据周期跨度大、数据价值高、数据量大。本文以若干个运载火箭型号、2.6万余次试验、328万余个数据文件为研究对象,对数据文件的文件类型进行统计分析,详见表1。

表1 文件类型分布

Table 1 Distribution of file types

文件类型	数量占比/%
jpg	25%
bin	33%
txt	31%
dat	4%
bmp	2%
log	3.5%
cmf	1.50%

根据文件类型分布分析,可结构化的txt、dat、bin文件类型的文件数量占比达到了总文件数量的65%,在架构设计中需着重考虑、专门设计结构化数据的存储。对328万余个运载火箭试验数据文件的文件大小、数量分布情况进行统计,如图1所示,横坐标为文件大小,统计单位为1MB,纵坐标为文件数量以10为底对数值。从分布图可知,文件数量统计分布呈现随文件大小增加而整体下降趋势,且1MB以下的文件数量呈现尖峰特点,在整体下降趋势中存在个别峰值是由于试验数据文件在生成或存储时,试验数据量相近或按固定大小的文件存储导致的。

对运载火箭试验数据文件大小的数据量分布情况进行统计,横坐标为文件大小,纵坐标为数据量。从图2可知,数据量集中在较大的文件范围内,1785MB左右的尖峰是由于该大小附近的文件不仅单文件数据量较大,由图1可知其文件数量也较多,因此由单文件平均数据量乘以文件数量得出的数据量最高。总结可知:运载火箭试验数据的小文件占用存储空间不大,但数量极多;大文件占用存储空间极大,但数量较少。

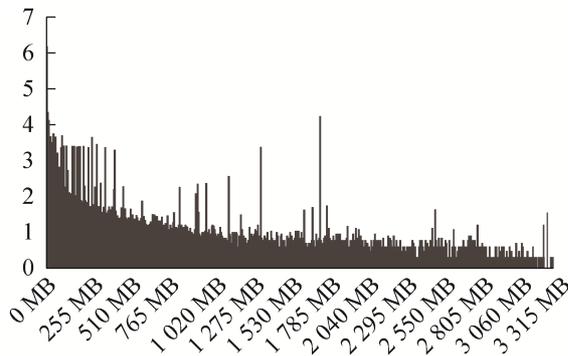


图1 文件数量随文件大小的分布
Fig. 1 Distribution of file quantity with different file size

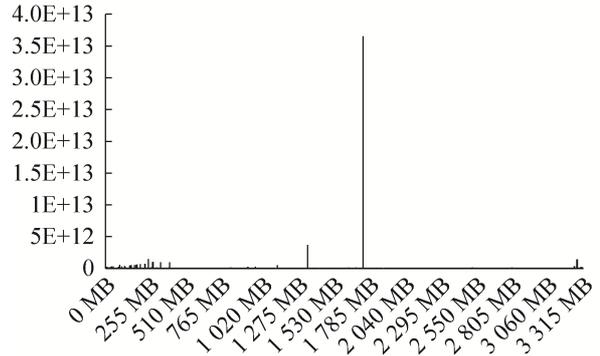


图2 数据量随文件大小的分布
Fig. 2 Distribution of data size with different file size

从运载火箭试验数据中区分结构化文件和非结构化文件,分别绘制文件数量随文件大小的分布、数据量随文件大小的分布如图3和图4所示,横坐标为文件大小,纵坐标为文件数量以10为底对数值或数据量,统计分布与不区分结构化和非结构化文件时的统计分布一致,小文件占用少但数量多、大文件占用多但数量少的结论依旧成立。

根据以上分析总结运载火箭试验数据并结合实际工作情况,得出试验数据特点如下:

- ① 试验数据的来源广泛,源于各型号、各类试验、各阶段流程;
- ② 试验数据种类多样,文件类型多,数据类型包括结构化和非结构化的多种数据类型;
- ③ 试验数据量大,传统单机或基于关系型数据库、文件难以满足数据存储和查询分析的需求;

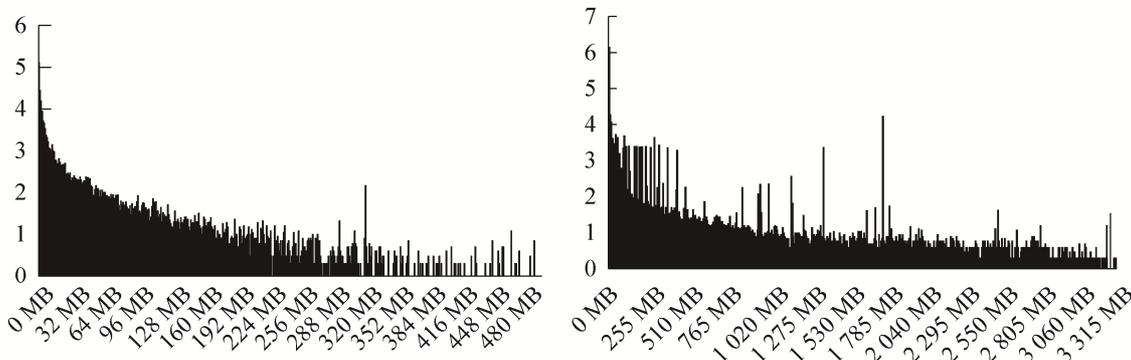


图3 结构化(左)和非结构化(右)文件的文件数量随文件大小的分布
Fig. 3 Distribution of file quantity with different file size in structured (left) and unstructured (right) files

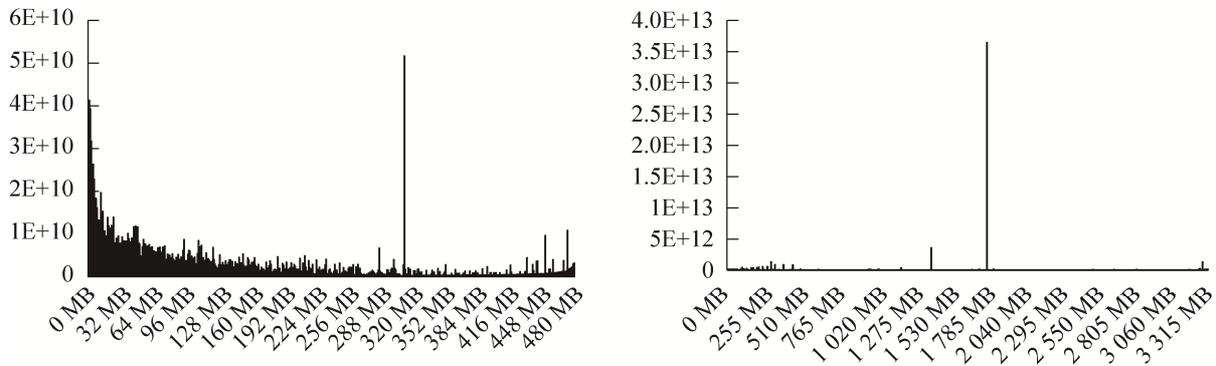


图 4 结构化（左）和非结构化（右）文件的数据量随文件大小的分布

Fig. 4 Distribution of data size with different file size in structured (left) and unstructured (right) files

④ 试验数据文件分布不均衡，结构化数据多，非结构化数据相对较少，小文件占用存储空间少但文件数量巨大，大文件占用存储空间大但文件数量极少；

⑤ 试验数值结果等结构化数据的数据量差异较大，数据采样频率不一，有的试验有几万或者十几万条数据，有的试验数据条数上亿甚至十几亿，数据采样频率从 50 Hz~500 Hz 不等，涉及参数几千个甚至上万个，单试验存储的数据量最大约 30 TB，单参数下存储约 15 亿条数据。

2 存储架构设计

目前在运载火箭领域，根据型号特点和数据特点的不同，开发了相应的试验数据系统，具备了一定的数据存储和查询分析的功能，但仍存在以下不足：

① 系统的通用性不足，试验数据种类繁多，无统一的试验数据系统以存储所有类型的试验数据，多为针对不同试验类型的主要数据类型作了专门的设计；

② 系统稳定性不够，试验数据系统多为单服务器部署，不支持集群部署，若节点宕机则可能导致正在录入系统的数据丢失以及系统服务临时瘫痪；

③ 系统的可扩展性不足，试验数据系统的架构设计多为基于关系型数据库或基于文件，系统容量有限，不适用海量试验数据的存储需求^[19]，系统难以通过增加硬件和存储空间的方式进行扩展；

为解决运载火箭试验数据类系统存在的上述问题，本文借鉴大数据技术的核心理念，通过整合大数据领域的数据存储组件或产品，设计出一套满足运载火箭试验大数据存储计算业务需求的存储架构。架构设计要能满足以下需求：

① 快速检索需求，按照型号-试验-文件的总体数据层次结构进行存储，能够通过型号属性、试验属性、文件属性等条件对数据进行快速检索；

② 结构化、非结构化数据存储需求，既能够存储结构化数据，也能够存储非结构化数据；

③ 小文件、大文件数据存储需求，既能够高效存储与读取数量极多的小文件，也能够用于大文件的高效存取；

④ 高效存取需求，既能够高效存储与读取数据条数达海量的结构化数据，也能够满足条数不多的结构化数据的高效存储与读取；

⑤ 稳定性、可扩展性需求，存储架构应具有良好的稳定性与灵活的可扩展性，以分布式部署提供容错、副本机制。

为满足设计需求，首先需划分文件大小的边界，按照文件大小分别选用相应的存储产品。根据统计分析，运载火箭试验大数据中以 32 MB 为划分阈值时，阈值以下的小文件数量刚好占 9 成；用于存储小文件的 SeaweedFS 虽然支持最大为 128 MB 的文件存储，但在 32 MB 时读写速度最优；HDFS 存储 32 MB~64 MB 的文件时虽然会有至少 50%的块空间浪费，但这个范围的文件数量仅占 1.8%，数据量仅

占 2%，实际影响并不大，因此选取 32 MB 为大小文件的划分阈值。将数量占 9 成的 32 MB 以下的小文件存储于适合小文件分布式存储的 SeaweedFS，将 32 MB 以上的大文件存储于适合大文件存储的 HDFS，减轻 HDFS 约 9 成的存储压力和成本。

架构设计如图 5 所示。运载火箭试验大数据经数据采集和文件预处理后，按照单个文件大小区分存储，将采集上传的压缩包和在共享存储中解压后的文件各存储一份，超过 32 MB 的大文件存储在 HDFS 中，小于 32 MB 的小文件存储在 SeaweedFS 中；对于解压后的结构化文件，解析后按照试验时长分别存储，时间短的试验数据存储于分布式数据库 HBase 中，时间长的试验数据存储于时序数据库 TDengine 中；对于系统的状态信息和运行过程中产生的数据或状态信息等通过关系型数据库如 MySQL、Oracle、SQLserver 等存储，以上存储产品中数据的存储关系映射统一存储于大数据全文检索组件 Elasticsearch 中，以便快速检索和定位查找。

架构设计中涉及到的主要存储组件选型决策如下：HDFS 是 Hadoop 分布式文件系统，适合存储海量的大文件，SeaweedFS 是专门适用于海量小文件的分布式文件系统。针对运载火箭试验数据里小文件众多的实际情况，选用 SeaweedFS 来存储小文件，减缓 HDFS 至少 90% 的文件存储压力；在解压压缩包过程中，采用共享对象存储产品作为解压空间，成本较低且容量动态伸缩灵活；对于结构化的解析数据而言，短时试验数据的表数量较少，约为 2 000 个，且数据量通常在千万字节以内，通过分布式数据库 HBase 实现列式存储和快捷查询，且 HBase 较传统关系型数据库更优。传统关系型数据库对于空值的存储是占用空间的，而 HBase 采用列式存储可以跳过空值存储，节省了存储资源，而试验数据的表数量较多，单次试验可达 100 张~300 张参数表，百次试验即可达上万张参数表，且参数表下的数据量通常可达千万字节以上。采用专用的时序数据库 TDengine 存储可以提供 90% 左右的高压缩率和按时间段、时间点查询的高效读取服务，且能有效缓解 HBase 因表数量过多导致的性能不稳定、存储速度降低问题；对于数据量一般、并发较低的系统状态信息和运行数据，采用关系型数据库 MySQL 存储，同时为提供快速、便捷的查询服务，采用 Elasticsearch 负责数据、文件的元数据信息存储，按单表 1 亿条记录横向分表，提供实时全文检索，如文件的存储位置、表名、关系等。

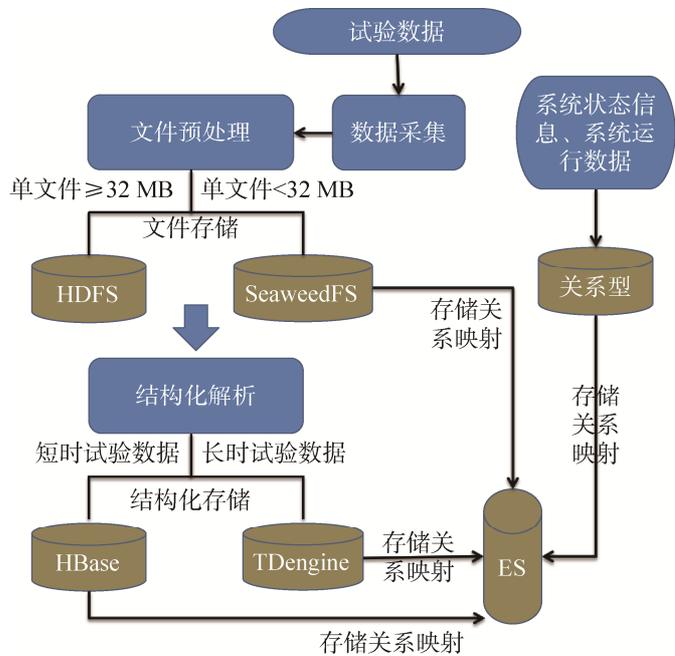


图 5 架构设计

Fig. 5 Structure design of system

3 数据模型设计

在运载火箭试验大数据存储架构设计中，存储是计算的前提，存储的性能高低影响计算的性能优劣，存储的性能不仅受限于存储的产品，也依赖于存储数据在存储产品上的建模。为满足运载火箭试验大数据的压缩存储和快速计算业务需求，对关键的存储产品进行模型设计。

3.1 HDFS 模型设计

HDFS 是 Google 的 GFS 论文的实现，充分利用廉价的计算机节点提供高吞吐量的数据访问，适合海量文件尤其是大文件的分布式存储。HDFS 模型设计的存储定位分析如下：单个文件的元数据约 200 B，按照 NameNode 节点内存 256 GB，则 HDFS 集群最多能容纳 13.7 亿个文件，而按照集群 1 PB 容量（目

前容量为 0.8 PB 左右)、HDFS 块大小 128 MB 计算, HDFS 集群最多能容纳 840 万个文件, 取较小值容量为 840 万个文件。而现阶段运载火箭试验数据已有 220 万个文件, 文件数量还在快速增长, 若按冗余容量: 已用容量=5:1 计算, 仍需扩容 0.6 PB 的集群容量, 对应存储成本将额外花费 60%。因此采用 SeaweedFS 和 HDFS 混合模式分布式存储文件, 两者各司其职, SeaweedFS 存储小于 32 MB 的小文件, 而 HDFS 专注存储超过 32 MB 的大文件。HDFS 的模型设计包括文件存储目录结构, 如图 6 所示。

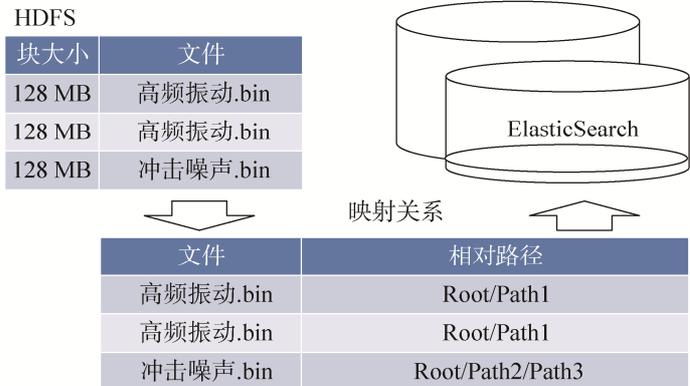


图 6 HDFS 模型设计

Fig. 6 Model design of HDFS

文件存储目录结构按照/datacenter/modelData 创建数据存储根目录, 在根目录下按照“根目录-文件”的层级关系存储, 被上传的数据文件在压缩包内的相对路径以键值对的映射关系存储于 ElasticSearch 中。

其中, 为方便查询文件, 实际存储于 HDFS 的文件名格式为“文件名_文件 ID_文件分片序号.文件类型后缀”, 文件 ID 全局唯一, 便于检索特定文件; 文件分片序号标识由于该文件对应的原始试验数据量过多而分裂的不同文件分片, 具体如图 7 所示。

文件名	文件ID	文件分片序号	文件类型后缀
高频振动	211204140959190501149	001	txt

图 7 HDFS 文件名格式设计

Fig. 7 Filename design of HDFS

3.2 SeaweedFS 模型设计

SeaweedFS 是 Facebook 的 Haystack 论文的实现, 集群内中央主服务器只管理文件卷, 无需管理全部文件的元数据, 而是由卷服务器管理, 减轻了中央主服务器的并发压力, 适合海量小文件的分布式存储, 还能提供约 12:1 的压缩比, 其模型设计主要包括文件存储目录结构和卷配置, 如图 8 所示。

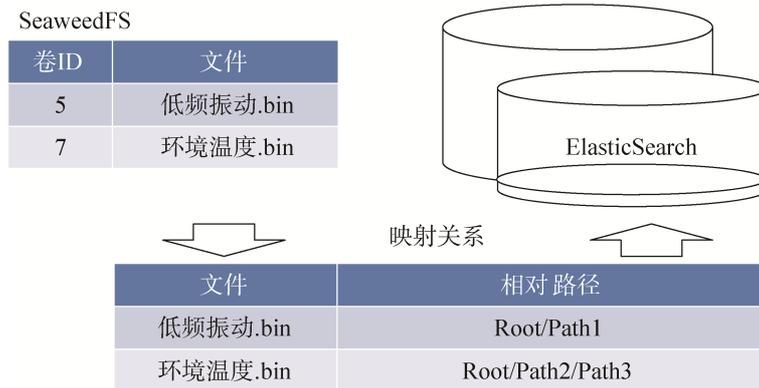


图 8 SeaweedFS 模型设计

Fig. 8 Model design of SeaweedFS

文件存储目录结构按照/home/data/seaweedfs/seaweedfs_volume 创建各个卷, 在各卷下按照“volume-文件”的层级关系存储, 被上传的数据文件在压缩包内的相对路径以键值对的映射关系存储于 ElasticSearch 中; 根据服务器数量和性能, SeaweedFS 在部署时采用分布式集群部署, 每个节点创建 3 个服务实例, 单节点平均创建 10 个卷。

3.3 TDengine 模型设计

分布式时序数据库 TDengine 的存储结构通常为: 数据库 Database-超级表模板 Stable-子表 Table 这三级。考虑到跨库查询和存储的不便, 仅创建一个数据库 modeldata。由于运载火箭试验数据的参数表的字段数量和字段类型各不相同, 因此不创建超级表模板 Stable, 而是在数据库下直接创建子表 Table。子表名的格式为“ods_文件 ID”, 文件 ID 到原始结构化文件的映射关系存储于 ElasticSearch 中; 给出子表的模型设计如图 9 所示。ts 为 TDengine 默认时间戳主键字段, 最高支持纳秒, 满足试验数据的时间戳精度以毫秒为主的存储需要, a0~aN 为参数表的所有字段名代号, 字段名代号和原始字段的映射关系存储于 ElasticSearch 中, 通过字段名代号映射建表, 不仅保障了数据安全和保密, 而且适应存储格式不一的结构化参数表; 根据服务器数量和性能, 分布式时序数据库 TDengine 按照集群分布式部署, 配置至少 2 个服务器节点部署为 Mnode 角色, 提供高可用的数据元数据管理服务, 所有节点均部署有 Vnode 角色, 提供分布式数据存储计算服务。

ts	a0	a1	aN
2020-09-19 17:00:17.245	61217.245738846	153.12199	1

图 9 TDengine 子表模型设计
Fig. 9 Table model design of TDengine

3.4 HBase 模型设计

3.4.1 列族模型设计

HBase 存储产品通过列族划分数据的存储和组织管理, 每个列族支持包含多列, 可实现灵活的数据存取。但当某列族存储数据达到落盘阈值时会触发所在表的所有列族同时落盘, 引发大量不必要的 IO 开销, 因此列族数量不宜过多, 实际配置为 2 个列族, 减少不必要的 I/O 开销, 提高写入效率。列族模型设计示意图如图 10 所示, col_alldata 列族中的列是对 col_data 列族的信息汇总, 其中 col_all 记录了 n 个参数值的集合, col_no 记录了表中数据记录的行序。

col_alldata列族		col_data列族			
col_all列	col_no列	col_参数1	col_参数2	col_参数n

图 10 HBase 列族模型设计
Fig. 10 Column family model design of HBase

查询业务通常为按列查询某参数和查询全部参数, 对于面向列族的 HBase 数据库而言, 更适合于按列查询单个参数, 查询某行键 Rowkey 的全部参数值需遍历所有列族的所有列, 查询 n 次效率较低, 改为查询 col_all 列后, 1 次查询即可完成全部参数的查询业务。

3.4.2 行键模型设计

HBase 采用 Key-Value 的键值对存储模型, 行键 Rowkey 是键值对存储模型的 Key, 表示唯一一行, 用于构建表中数据记录的索引, 行键模型的设计影响查询的效率和应用的便捷性。根据数据关联、遍历查询、特定试验查询、特定文件查询、数据记录行过滤查询等业务需要, 按照一个试验一张表, 给出 HBase 表名设计和 HBase 表的行键模型设计, 如图 11 所示。表名格式为“ods_modelData_型号 ID_试验 ID_data”, 在查找特定型号或特定试验的表时, 可方便地通过遍历表名获取。

表名前缀	型号ID	试验ID	表名后缀
ods_modelData	XH000106	211229155803977003728	data

图 11 HBase 表名设计
Fig. 11 Table name design of HBase

Rowkey 格式为“Region 分区编号_试验 ID_文件 ID_数据记录所处序号”，试验 ID 和文件 ID 是全局唯一的，在查询特定试验或文件时、查询特定数据记录行时，无需查询表中数值，只需遍历行键 Rowkey 即可定位；在查询某文件对应的全部数据记录时，配置 Startkey 和 Endkey 分别为“:”和“#”，即可快捷匹配查询。

分区编号	试验ID	文件ID	数据记录序号
0000	211204140959190501149	211204141000048551185	00000000005

图 12 HBase 行键模型设计
Fig. 12 Rowkey design of HBase

3.4.3 Region 模型设计

Region 是 HBase 中分布式存储和负载均衡的最小单元，不同 Region 分布在不同的 RegionServer 节点上，每个 Region 包含多个列族。关于 Region 的数量，官方建议每个 RegionServer 节点最多创建 10 GB×100 个 Region，通常系统部署在 5 台~10 台 RegionServer。按官方建议最多创建 500 个~1 000 个 Region，因此需合理设计 Region，使得 Region 数量不要超过此最大值，根据实际物理机器数量和性能，最大值可延伸到 10 000 个 Region 左右。运载火箭试验大数据不同于互联网大数据，数据具有不稳定性，数据量不定且差异很大，因此在 Region 模型设计时，按照一个试验一张表，一张表两个列族，不进行 Region 预分区，而是采用默认的分区机制，即一张表默认存于一个 Region 中，若该表数据量达到 Region 最大容量阈值，则由 HBase 自动分区为两个或更多 Region，以避免不必要的存储占用和 Region 数量过多导致的 HBase 性能降低。相比传统互联网大数据按照表行号取余 Region 总数进行预分区的方式，Region 数量减少了(Region 总数-1)个，对于 N 个试验，共减少 N×(Region 总数-1)个 Region 数量，明显缓解 HBase 的存储压力。

3.5 ElasticSearch 模型设计

ElasticSearch 模型主要设计了四种映射关系，分别为 ods_modeldata_dat、ods_modeldata_subdat、ods_modeldata_subdat_trial、ods_modeldata_meta，存储的映射关系如下所示。

表 2 ElasticSearch 映射关系明细
Table 2 Relationship of matching in ElasticSearch

映射关系名	映射关系明细	映射关系含义
ods_modeldata_dat	文件→文件的相对路径	原始文件属性存储关系映射库
ods_modeldata_subdat	转换部分元数据属性后的文件→文件的相对路径	原始文件属性存储关系映射库
ods_modeldata_subdat_trial	试验元数据→属性信息	试验信息映射库
ods_modeldata_meta	结构化文件的参数元数据→属性信息	结构化信息映射库

4 应用效果分析

4.1 部署模式

根据上述存储架构设计和模型设计，将系统部署于实际服务器集群中，服务器配置见表 3，按照服务器角色、内存、硬盘的配置差异，区分为 3 组，分别包含 3 个、21 个、3 个服务器节点。

表 3 服务器配置
Table 3 Configuration of servers

节点数	主机名	服务器角色	内存/GB	磁盘/TB	CPU 核数	操作系统	CPU 架构
3	MG-01~03	管理服务器	512	4	32	CentOS7.6	x86_64
21	WK-01~21	工作服务器	256	110	32	CentOS7.6	x86_64
3	SC-01~03	索引服务器	256	33	32	CentOS7.6	x86_64

对系统架构中涉及的存储产品 HDFS、SeaweedFS、HBase、TDengine、ElasticSearch 的具体部署模

式展开见表 4~表 8。

4.2 应用效果分析

在以上软硬件部署模式下, 对几种涉及到的存储产品的应用效果进行实测, 得出以下结果。

HDFS 和 SeaweedFS 存储试验数据的统计分布见表 9, HDFS 存储的文件数量占比仅为 8.4%, 数据量占比为 60.7%, 与上述运载火箭试验大数据特点分析的结果一致, 即大文件数据量高但文件数量少, 小文件的文件数量多但数据量少; 同时以 SeaweedFS 集群的某存储节点为例, 存储原始数据量为 438.4 GB, 实际占用 86.2 GB, 压缩率为 80.3%。随着运载火箭试验数据量的持续增长, 小文件的空间占用需求增加, SeaweedFS 能大大缓解存储空间有限的问题。通过表 9 分析可知: 运载火箭试验大数据存储架构设计能够有效解决由于文件数量过多导致 Hadoop 的 NameNode 存储瓶颈的问题, 又能解决 SeaweedFS 无法高效存储大文件的问题, 充分发挥两个存储组件的各自优势。

表 4 HDFS 部署模式

节点数	角色	存储产品	主机名
1	NameNode 主	HDFS	MG-01
1	NameNode 备	HDFS	MG-02
21	DataNode1~21	HDFS	WK-01~21

表 6 HBase 部署模式

节点数	角色	存储产品	主机名
1	ThriftServer	HBase	MG-01
1	ThriftServer/HMaster 主	HBase	MG-02
1	ThriftServer/HMaster 备	HBase	MG-03
5	RegionServer1~5	HBase	WK-01~05

表 8 ElasticSearch 部署模式

节点数	角色	存储产品	主机名
3	EsMaster1~3	ElasticSearch	MG-01~03
3	EsNode1	ElasticSearch	SC-01~03
3	EsNode2	ElasticSearch	SC-01~03
3	EsNode3	ElasticSearch	SC-01~03
3	EsNode4	ElasticSearch	SC-01~03

表 5 SeaweedFS 部署模式

实例数量	卷数量	节点数	存储产品	主机名
3	10	1	SeaweedFS	WK-01
3	8	1	SeaweedFS	WK-02
3	10	1	SeaweedFS	WK-03
3	13	1	SeaweedFS	WK-04
3	10	1	SeaweedFS	WK-05
3	10	1	SeaweedFS	WK-06
3	10	1	SeaweedFS	WK-07

表 7 TDengine 部署模式

节点数	角色	存储产品	主机名
5	Mnode	TDengine	WK-01~05
7	Vnode	TDengine	WK-01~07

表 9 HDFS 和 SeaweedFS 存储统计分布

存储产品	数据量/MB	文件数量	数据量占比/%	文件数量占比/%
HDFS	62 082 505	386 927	60.7	8.4
SeaweedFS	40 250 325	4 209 301	39.3	91.6

HBase 的单表占用存储空间的数量统计分布柱状图如图 13 所示, 横坐标为表的存储空间占用量, 纵坐标为占用量对应的表个数以 10 为底的对数值, 可见单表占用的存储空间分布较为均衡。根据统计, 单表最大占用存储空间 610 GB, 可见 HBase 支持大数据量单表和短时试验数据表的存储。

TDengine 和 HBase 存储试验数据量的分布情况为 HBase 存储数据量占比 96.42%, TDengine 存储数据数据量占比 3.58%, 可见短时试验数据占比较长时试验数据更多, 短期试验的数据量约为长时试验数据的 10.3 倍, TDengine 解决了长时试验数据的存储和按时间查

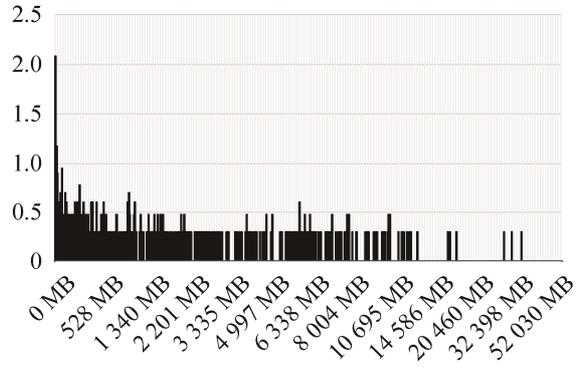


图 13 表数量随表大小的分布
Fig. 13 Distribution of data types

询的问题,同时提供90%的压缩率,极大缓解了服务器的存储压力。

综上所述,本文贴合运载火箭试验大数据的存储计算业务的实际需求,梳理调研国内外主流大数据组件和技术,针对不同的数据类型和不同的存储产品特性,通过组合集成,给出了一种基于大数据技术的运载火箭试验大数据存储架构。该系统架构相较传统基于关系型数据库与文件的系统架构具有以下效果:

① 高性能。经实际运行部署测试,该大数据存储架构满足院级的运载火箭试验大数据存储计算业务需求。以院级运载火箭试验大数据系统为例,数据采集服务器单节点能够处理2400个并发数据上传请求,实际部署五节点支持每秒1.2万次的并发请求;Seaweed-FS对小文件的存储读取性能优,空间占用少,压缩率高达80.3%;时序数据库TDengine对试验数据的存储速度达每秒70万点,存储压缩率达90%;HDFS、SeaweedFS、TDengine等存储产品提供天然的数据同步和副本机制,为数据的可靠性提供了保证;传统系统受限单机的磁盘读写、内存缓存速度、CPU计算能力、CPU核数等资源瓶颈制约,且存在单点故障,可靠性低、性能一般。

② 扩展性。ES单库支持300亿条数据的查询,集群能够支撑未来约十年的运载火箭试验大数据的存储需要,后续配合数据治理,在数据治理归档后数据文件的量也会缩减,减轻ES的负载量,提升系统未来的存储查询能力。此外,存储架构设计中HDFS、SeaweedFS、HBase、TDengine等存储产品均支持集群部署,提供良好的存储资源和计算资源弹性扩展;传统系统仅支持单机部署,往往需要高性能的服务器作为系统运行环境,难以提供弹性资源扩展。

③ 稳定性。该大数据存储架构具有良好的稳定性,架构设计中的各存储组件以分布式部署提供容错、副本机制,满足三个副本的基本备份要求;克服了传统试验数据系统在单服务器部署模式下存在的单点故障问题,保障了系统服务的稳定。

④ 通用性。满足目前所有类型运载火箭试验数据的统一存储,并针对结构化数据作了区分存储;满足目前工程应用中通过对型号属性、试验属性、文件属性等条件过滤的常规检索需求;满足结构化、非结构化数据存储需求;满足小文件、大文件数据的通用存储需求。

5 结束语

遵循通用性、可靠性及可扩展性的原则,本文给出了基于大数据核心技术的运载火箭试验大数据存储架构设计,首先从整体层面给出了系统的存储架构设计,接着分析了存储架构设计中涉及到的关键存储产品选型和模型设计。相较传统单机部署的基于关系型数据库或文件的系统,本文提出的系统充分利用主流大数据技术和大数据组件,集成各组件的优势,针对存储计算业务的实际需求,整合相应的存储组件,并给出了选型分析和存储模型设计方法。通过实际工程应用表明,该架构是一种切实可行的大数据存储架构设计,能够满足运载火箭业务的需求。

参考文献

- [1] 杨露. 大数据平台上的隐私保护及合规关键技术研究[D]. 成都: 四川大学, 2021.
- [2] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system[J]. *Acm Sigops Operating Systems Review*, 2003, 37(5): 29-43.
- [3] DEAN J. MapReduce: Simplified data processing on large clusters[C]// *Symposium on Operating System Design & Implementation*, 2004.
- [4] CHANG F, DEAN J, GHEMAWAT S, et al. Bigtable: A distributed storage system for structured data[C]// *7th Symposium on Operating Systems Design and Implementation (OSDI '06)*, November 6-8, 2006, Seattle, WA, USA.
- [5] 张维. 基于大数据技术的制造企业信息化平台的设计与实现[D]. 西安: 西安理工大学, 2021.
- [6] 李梓杨. 大数据流式计算环境下的弹性资源调度策略研究[D]. 乌鲁木齐: 新疆大学, 2021.
- [7] 李薇. 基于云计算的大数据处理技术探讨[J]. *数字技术与应用*, 2017(8): 218-219.
- LI Wei. Discussion on big data processing technology based on cloud computing[J]. *Digital and Application*, 2017(8):

- 218–219.
- [8] 赵鹏, 朱祎兰. 大数据技术综述与发展展望[J]. 宇航总体技术, 2022, 6(1): 55–60.
ZHAO Peng, ZHU Yilan. Overview and development prospect of big data technology[J]. Aerospace General Technology, 2022, 6(1): 55–60.
- [9] 侯晓芳, 王欢, 李瑛. 一种基于 HIVE 和分布式集群的大量数据高效处理方法研究[J]. 中国电子科学研究院学报, 2018, 13(3): 315–320.
HOU Xiaofang, WANG Huan, LI Ying. Research on an efficient processing method of large amount of data based on HIVE and distributed cluster[J]. Journal of Chinese Academy of Electronic Sciences, 2018, 13(3): 315–320.
- [10] 胡俊, 胡贤德, 程家兴. 基于 Spark 的大数据混合计算模型[J]. 计算机系统应用, 2015, 24(4): 214–218.
HU Jun, HU Xiande, CHENG Jiaying. Big data hybrid computing model based on Spark[J]. Computer System Application, 2015, 24(4): 214–218.
- [11] 周宇, 曹英楠, 王永超. 面向大数据的数据处理与分析算法综述[J]. 南京航空航天大学学报, 2021, 53(5): 664–676.
ZHOU Yu, CAO Yingnan, WANG Yongchao. Overview of data processing and analysis algorithms for big data[J]. Journal of Nanjing University of Aeronautics and Astronautics, 2021, 53(5): 664–676.
- [12] BEAVER D, KUMAR S, LI H C, et al. Finding a needle in haystack: Facebook's photo storage[C]/9th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2010, October 4–6, 2010, Vancouver, BC, Canada.
- [13] 管登荣. 基于 SeaweedFS 的分布式文件管理系统的设计与实现[D]. 南京: 南京大学, 2018.
- [14] 陈辰. TiDB 开源社区最佳实践[J]. 软件和集成电路, 2021(12): 56–57.
CHEN Chen. TiDB open source community best practices[J]. Software and Integrated Circuits, 2021(12): 56–57.
- [15] 董雪, 高远, 敖炳. 基于 TDengine 的智能电网监控系统数据存储方法研究[J]. 电气应用, 2021, 40(8): 68–74.
DONG Xue, GAO Yuan, AO Bing. Research on data storage method of smart grid monitoring system based on TDengine[J]. Electrical Applications, 2021, 40(8): 68–74.
- [16] 张宇耀. 基于大数据的企业用户数据分析平台的设计与实现[D]. 北京: 北京交通大学, 2019.
- [17] 李亚臣. 基于 ClickHouse 的用户事件分析系统的设计与实现[J]. 信息与电脑, 2021, 33(9): 87–90.
LI Yachen. Design and implementation of user event analysis system based on Clickhouse[J]. Information and Computer, 2021, 33(9): 87–90.
- [18] 连彦泽, 何信华, 李鹏程, 等. IPL: 运载火箭测试数据自动判读语言[J]. 遥测遥控, 2022, 43(2): 36–45.
LIAN Yanze, HE Xinhua, LI Pengcheng, et al. IPL: Automatic interpretation language for launch vehicle test data[J]. Journal of Telemetry, Tracking and Command, 2022, 43(2): 36–45.
- [19] 何巍, 胡久辉, 赵婷, 等. 基于模型的运载火箭总体设计方法初探[J]. 导弹与航天运载技术, 2021(1): 12–17, 32.
HE Wei, HU Jiuhui, ZHAO Ting, et al. Research on model based launch vehicle overall design[J]. Missile and Space Delivery Technology, 2021(1): 12–17, 32.

[作者简介]

- 连彦泽 1985 年生, 硕士, 高级工程师, 主要研究方向为航天数据分析与软件工程。
李鹏程 1997 年生, 硕士, 助理工程师, 主要研究方向为运载火箭计算机辅助与设计。
赵雷 1982 年生, 硕士, 研究员, 主要研究方向为运载火箭软件系统。
司洪泉 1982 年生, 硕士, 工程师, 主要研究方向为航天领域信息化。
陈旭东 1979 年生, 硕士, 高级工程师, 主要研究方向为航天领域信息化。

(本文编辑: 傅 杰)