

引用格式: 孟令达, 朱琳, 陈瑞琼, 等. 基于非关系型的时间频率科学数据存储策略研究[J]. 时间频率学报, 2021, 44(4): 331-344.

# 基于非关系型的时间频率科学数据存储策略研究

孟令达<sup>1,2</sup>, 朱琳<sup>1,2,3</sup>, 陈瑞琼<sup>1,2</sup>, 刘娅<sup>1,2,3</sup>, 李孝辉<sup>1,2,3</sup>

1. 中国科学院 国家授时中心, 西安 710600;
2. 中国科学院 时间频率基准重点实验室, 西安 710600;
3. 中国科学院大学, 北京 100049)

**摘要:** 针对时间频率科学数据入库与检索效率低、多源异构数据存储管理困难、多用户高并发数据服务能力差的问题, 提出了一种基于非关系型 (NoSQL) 数据库的时间频率科学数据混合存储策略。首先, 分析了时间频率科学数据特征及主流 NoSQL 数据库的特点, 结合数据及信息服务需求, 提出了基于 NoSQL 数据库的混合存储策略; 然后, 基于提出的混合存储策略设计了数据存储系统架构及模型; 最后, 对提出的存储策略进行可行性和有效性进行验证。结果表明, 提出的混合存储策略较传统存储策略性能有较大提升, 能够有效地满足时间频率科学数据服务、数据分析及知识获取中对数据实时请求、高并发访问及扩展性的需求。

**关键词:** 时间频率; 科学数据管理; 数据存储策略; 数据库

DOI: 10.13875/j.issn.1674-0637.2021-04-0331-14

## Research on the storage strategy of time and frequency scientific data based on non-relational database

MENG Ling-da<sup>1,2</sup>, ZHU Lin<sup>1,2,3</sup>, CHEN Rui-qiong<sup>1,2</sup>, LIU Ya<sup>1,2,3</sup>, LI Xiao-hui<sup>1,2,3</sup>

1. National Time Service Center, Chinese Academy of Sciences, Xi'an 710600, China;
2. Key Laboratory of Time and Frequency Primary Standards, Chinese Academy of Sciences, Xi'an 710600, China;
3. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** In order to solve problems in storage and retrieval of time and frequency data, such as low retrieval efficiency, management of heterogeneous data, poor service capability of multi-user high concurrent data, a hybrid storage strategy of time and frequency data, based on non-relational (NoSQL) database, was proposed. The reasons of the proposed hybrid storage strategy of time and frequency data are firstly discussed, which are mainly due to the properties of time and frequency data and requirements on services of the data. Then, we design a architecture model of the data storage system. Finally, the feasibility and effectiveness of the proposed hybrid storage strategy are verified. The results show that, compared with traditional storage strategy, performances of the proposed storage strategy are greatly improved, on subjects of real-time data request, high concurrent access, and

收稿日期: 2021-05-31; 接受日期: 2021-07-05

基金项目: 国家自然科学基金资助项目 (12003040)

expandability of time and frequency data.

**Key words:** time and frequency; scientific data management; data storage strategy; database

## 0 引言

时间频率科学数据来源于时间频率研究、重大科学仪器研制与试验和国家大科学装置建设与运行过程中产生的科学研究实验数据、大科学仪器试验数据, 和大科学装置产生的观测数据及国内外时间频率科学研究机构和组织的相关时间频率资料和信息等数据。加强和规范时间频率科学数据管理, 保障科学数据安全, 提高数据存储管理与开放共享水平, 将进一步促进数据资源共享、按需使用、个性化定制等新数据及信息服务模式。同时, 为大数据时代海量时间频率科学数据知识获取和科学发现提供基础, 更好地支撑国家科技创新、经济社会发展和国家安全。

近年来, 时间频率科学的发展以及信息技术尤其数据获取技术的发展和在时间频率科学中的广泛应用, 推动了时间频率科学数据的获取、传输、存储和处理技术的进步。随着时间频率科学大科学装置、科学工程以及重大科学仪器的研制和建设, 各系统都开展了时间频率科学数据库建设工作。这造成了时间频率科学数据种类越来越多、数据获取周期越来越短、数据量爆炸式增长, 各系统建设的时间频率科学数据中心在科学数据的采集、存储管理和数据服务在地理上呈分布式态势, 在科学数据业务类型上呈互补态势。这一方面促使时间频率科学数据产品的种类、数据量不断增加, 时间频率应用领域不断拓展, 另一方面海量、异构、分布式的时间频率科学数据为传统的数据管理方法带来了巨大挑战。因此, 如何高效地对海量时间频率科学数据进行存储管理, 实现面向服务的多用户并发数据处理, 以更好地满足知识获取, 成为亟待解决的技术问题。

## 1 时间频率科学数据存储策略

### 1.1 时间频率科学数据管理现状

时间频率科学数据按照时间频率科学业务系统类型并结合时间频率科学研究特点, 可将我国时间频率科学数据来源划分为守时系统类数据、授时系统类数据、授时监测系统类数据和科学实验与仪器研制类数据四个方面, 具体时间频率科学数据来源和数据种类如表 1 所示。

表 1 时间频率科学数据种类汇总表

数据来源	数据种类	
守时系统类数据	主钟比对数据	综合原子时比对数据
	国际原子时与我国地方原子时比对数据	协调世界时与我国地方原子时比对数据
	各国原子钟与国际原子时计算取权数据	《时间频率公报》电子期刊
	原子钟状态参数数据	原子钟运行维护日志
	卫星共视比对数据	卫星双向比对数据
	公报数据 (IGRS、CIRT)	用户与机构数据
授时系统类数据	守时系统设备信息	
	授时参数控制数据	授时参考频率源比对数据
	地球自转数据	授时系统时间信号数据
	授时信号传播延迟数据	授时系统设备信息
	授时系统状态监控数据	授时运行保障日志

续表 1

数据来源	数据种类	
授时监测系统类数据	授时系统授时偏差数据	载波相位测量数据
	信号质量数据	传播路径时延数据
	授时监测原始数据	信号质量评估数据
	授时性能评估数据	授时监测设备信息
	授时监测系统状态数据	授时监测系统产品数据
科学实验与仪器研制类数据	原始实验数据	实验过程数据
	实验结果数据	实验过程文档
	研制测试数据	研制过程文档
	设备标校数据	实验设备信息

时间频率科学由于业务面向对象、业务系统分布区域、时间频率测量比对手段、时间频率数据处理方法、时间频率科学业务系统关联性等方面的因素，时间频率科学数据具有数据量大（TB 或 PB 级）、数据种类繁多、数据结构复杂、数据计算密集、数据交叉融合、多层次数据迭代演化、短期应用时效性和长期研究高价值特征。

现阶段，时间频率科学数据主要采用集中式存储，数据管理方式包括文件方式、关系数据库及文件加数据库索引 3 种管理方式。其中，基于文件方式的时间频率科学数据管理是采用操作系统中的文件管理系统实现数据文件组织管理。文件管理系统是操作系统用于明确存储介质和分区的方法与结构，对文件存储介质的空间进行组织分配，并对存储的文件的保护和检索提供系统接口、对象操作和管理。通过附加的文本数据描述信息和文件组织规范，实现数据组织和数据实例之间的关联<sup>[1]</sup>。关系型数据库是一种广为应用的数据库管理系统，利用数据库系统提供的接口实现数据的查询和存取。采用数据库存储时间频率科学数据是将测量比对设备产生的科学数据，通过关系表和表的列属性实现数据记录的存储和数据之间关系的描述<sup>[2-3]</sup>。但由于时间频率科学数据存在种类、结构与格式多样化的特征，采用单一且更适用于结构化、事务型的关系型数据库管理方式，将难以实现海量、异构数据的有效处理；文件加数据库索引是采用文件系统和关系型数据库组合的存储管理模式，数据文件中实体数据存储于文件系统中，文件存储目录和元数据信息则存储于关系型数据库中<sup>[4-5]</sup>。文件加数据库索引方式实现了复杂时间频率科学数据文件目录的有效管理和数据文件检索，但由于数据文件和元数据的分别存储会导致数据一致性差以及管理困难，同时在进行数据服务和数据分析时，仍存在数据量大、数据结构复杂时传统文件系统读写效率低的问题。

综上分析，现有的时间频率科学数据存储上要采用集中式存储架构和基于文件系统或关系型数据库的统一数据存储管理方式，这类系统数据入库与检索效率低、存储容量有限、扩展能力差、数据模型固定，无法满足云计算和大数据计算环境下海量、异构、分布式数据在知识获取与面向服务应用中的高效数据存储管理和多用户高并发访问需求。

## 1.2 NoSQL 数据存储管理系统

大数据技术的分布式文件系统与 NoSQL (not only SQL) 数据库为海量数据高效存储、快速检索、大数据知识获取和在线数据分析与信息服务时间频率科学数据存储管理提供技术解决途径。NoSQL 数据库提供了比关系型数据库简单、灵活的架构和数据关系，具有高可用性、可伸缩性、负载均衡、高容错能力和异构环境运行的特征<sup>[6-7]</sup>。典型的 NoSQL 产品有谷歌的 Big Table、Apache Hadoop 的 Key-Column 数据库 HBase、文档数据库 MonogoDB、Key-Value 数据库 (InfluxDB) 和图数据库 Hyper-GraphDB 等<sup>[8-9]</sup>。

各类 NoSQL 主要特征如表 2 所示。

表 2 典型 NoSQL 数据库及特征

NoSQL 数据库	主要特征
BigTable	分布式、高度可扩展的结构化数据存储管理
HBase	通过读/写请求提供对可伸缩 Big Table 的一致、随机和实时访问
MongoDB	为基于文档的动态模式提供关系数据模型工具，支持跨多个接口 从不同应用程序快速一致地访问
HyperGraphDB	设计用于人工智能和语义 Web 项目的持久存储模型，提供关系 和面向图的数据管理
InfluxDB	时间序列数据库，旨在处理高写入和查询负载

HBase 在 HDFS (hadoop distributed file system) 之上提供了对大数据的可伸缩、分布式、容错和随机读写导向。HBase 根据列族来存储数据，列族包含多个列且可任意增加和删除 (百万级); HBase 通过添加 RegionServer 提升 HBase 上层的处理能力，通过添加 Datanode 扩容; 高并发时 HBase 集群中的单个节点 IO 延迟相对平稳，能获得高并发、低延迟的服务; 稀疏主要是针对 HBase 列的灵活性，在列族中指定任意多的列，在列实体数据为空的情况下不占用存储空间<sup>[10-11]</sup>。

InfluxDB 是一个专门用于处理时间序列数据的高写入和查询负载的数据库。InfluxDB 采用 LSM (log-structured mergetree) 结构，数据先写入内存，当内存容量达到一定阈值之后 flush 到文件，将时间序列数据按照时间线进行管理，使得同一数据源的 tags 不再冗余存储，降低了数据的存储量; 时间序列和数值可在同一个 Block 内独立存储，极大地提高了数据压缩和检索效率<sup>[12]</sup>。

### 1.3 基于 NoSQL 时间频率科学数据存储策略

时间频率科学数据具有非结构化、半结构化、时序化及结构化数据并存的特征。同时，数据存储管理策略需满足数据服务、数据分析与知识获取的高效存取性能需求。因此，本文采用分布式文件系统、分布式数据库管理系统、时间序列数据库和关系型数据库管理系统组合存储管理策略，设计符合各数据特征的数据存储模型。基于 NoSQL 时间频率科学数据混合存储策略如图 1 所示。

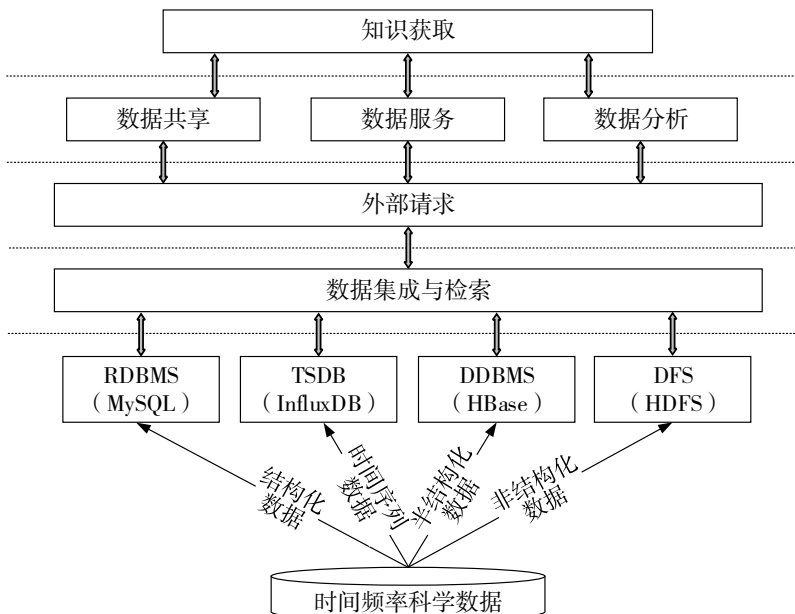


图 1 基于 NoSQL 的时间频率科学数据存储策略

本文采用 HDFS 对时间频率科学数据中非结构化数据文件存储，HDFS 多副本、高吞吐量等特性，结合其计算引擎可满足海量历史数据批量处理、格式转换和用户导入和读取海量的数据文件存储需求。同时，HDFS 用于 HBase、MapReduce 和 Spark 等数据管理和并行计算引擎的底层数据文件永久性存储。采用 HBase 分布式数据库存储管理时间频率科学数据中的半结构化和无关联的结构化科学数据，HBase 的易扩展、高并发及稀疏等特征，可满足数据格式变化和面向服务时用户高并发低延迟的存储管理需求。采用 MySQL 关系型数据库存储管理时间频率科学数据中的具有关联特性的设备信息、用户信息等事务型结构化数据，满足关联型、事务型数据存储管理需求。采用 InfluxDB 时间序列数据库存储管理具有时间序列特征的测量比对、状态监控等时间频率科学数据。InfluxDB 在数据模型的存储上针对时间序列数据进行多项优化及其时序性、类 SQL 语法、支持时间和无结构的特性，满足多源时序数据的高写入和实时分析的查询负载需求。

## 2 时间频率科学数据存储模型

### 2.1 基于 HBase 的半结构化数据存储模型

#### 1) 基于 HBase 的时间频率科学数据存储架构

HBase (hadoop database) 是一个高可靠性、高性能、面向列、可伸缩的建立在 HDFS 上的分布式数据库，HBase 以 HTable 数据表存储数据。HTable 是一张多维增强的稀疏映射表，具有海量、面向列、极易扩展、高并发和稀疏等主要特征，在半结构化与结构化大数据存储、分析方面具有很好的性能。基于 HBase 数据管理系统的特性结合时间频率科学数据存储管理需求和应用场景，设计了基于 HBase 的时间频率科学数据存储架构，如图 2 所示。

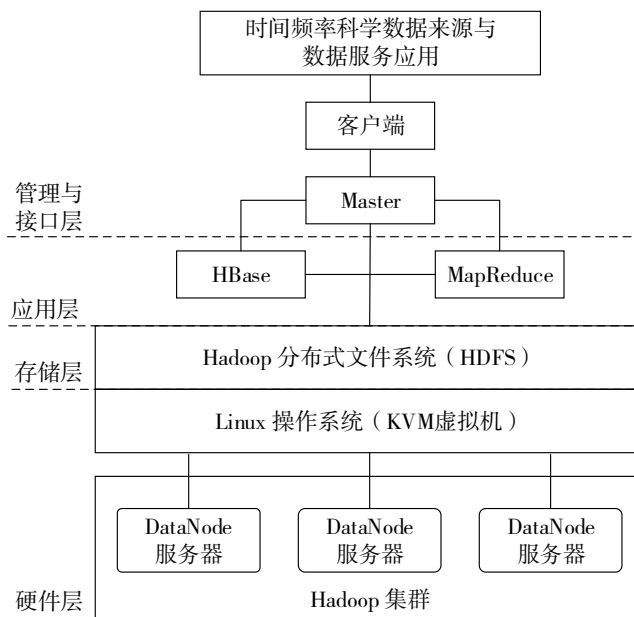


图 2 基于 HBase 的时间频率科学数据存储系统

基于 HBase 的时间频率科学数据存储系统分为以硬件层、存储层、应用层和管理与接口层。硬件层有多台服务器组成集群，采用虚拟化技术构建资源池并创建系统所需的 Linux 操作系统主机；存储层为 Master 管理下的 Hadoop 集群，采用 Hadoop 平台建立 HDFS 文件系统持久存储 HBase 表数据；应用层包括 HBase 相关组件与 MapReduce 编程模型，提供数据存储以及应用开发的访问接口。管理与接口层由数据产生区域、客户端和 Master 组成，并通过客户端完成海量结构化或半结构化数据存储和快速访问。

## 2) 基于 HBase 的 GNSS 系统时差监测数据存储模型

GNSS 系统时差监测数据是时间频率科学数据中典型的半结构化大数据。主要是由于 GNSS 系统和卫星数量多使得 GNSS 系统时差监测数据大, GNSS 系统时差监测系统平均每天采集 19.8 万条数据。同时 GNSS 定时接收机输出数据格式随着观测周期内可视卫星数量而变化, 而各 GNSS 升级更新后监测数据参量类型也在变化, 导致监测数据格式不固定。

GNSS 系统时差监测一条数据记录主要包括观测时间的年月日时分秒 (TIME)、计数器值 (TICV)、可见卫星数量 (SUM)、卫星号 (PRN)、接收机钟差 (RCLK)、卫星钟差 (SCLK)、伪距 (R1)、X (SATX)、Y (SATY)、Z (SATZ)、电离层延迟 (IONC)、对流层延迟 (TROP) 等, 数据长度随可见卫星数量不同而变化, 数据格式如图 3 所示。

GNSS 系统时间监测原始数据							
TIME	TICV	SUM	PRN	RCLK	SCLK	...	PRN
20191003235001	321.202 0	3	2	-105.651 9	-95 478.252	...	3
RCLK	SCLK	...	PRN	RCLK	SCLK	...	
-90.559 7	-5 409.731	...	6	-91.764 7	-23 499.378	...	

图 3 GNSS 系统时间监测原始数据格式

基于传统关系型数据库存储的 GNSS 系统时差监测原始数据, 由于关系型数据库数据存储模型的限制, 将同一时间观测到的数据按照卫星编号转换成结构化数据进行存储, 数据存储模型如图 4 所示, 由于篇幅所限, 图中只列出了前 7 项数据列中少数几行数据记录。

ID	NUM	PRN	TIME	TICV	RCLK	SCLK
1	7	2	20191003235001	321.202	-105.651	-95 478.252
2	7	3	20191003235001	321.202	-90.559	-5 409.731
3	7	6	20191003235001	321.202	-91.764	-23 499.378
4	7	17	20191003235001	321.202	-63.322	41 225.824
5	7	19	20191003235001	321.202	-34.969	-77 535.866
6	7	22	20191003235001	321.202	-11.658	-228 085.188
7	7	28	20191003235001	321.202	-67.322	227 928.015
8	7	2	20191004001705	320.931	-95.580	-95 481.155
9	7	3	20191004001705	320.931	-85.203	-5 412.224
10	7	6	20191004001705	320.931	-92.243	-23 505.452
11	7	12	20191004001705	320.931	-17.987	59 255.486
12	7	17	20191004001705	320.931	-69.317	41 230.970
13	7	19	20191004001705	320.931	-29.289	-77 534.799
14	7	23	20191004001705	320.931	-109.353	-49 000.276
15	8	2	20191004025012	314.625	-84.060	-95 488.614
16	8	5	20191004025012	314.625	-65.984	-595.450
17	8	6	20191004025012	314.625	-109.988	-23 539.626
18	8	9	20191004025012	314.625	-62.399	-14 230.682
19	8	12	20191004025012	314.625	-59.365	59 242.058
20	8	17	20191004025012	314.625	-64.621	41 252.965
21	8	19	20191004025012	314.625	-45.751	-77 524.699
22	8	25	20191004025012	314.625	-17.037	-4318.001

图 4 GNSS 系统时差监测原始数据 RMDB 数据存储逻辑视图

GNSS 系统时差原始监测数据经结构化转换后，原来一条数据记录转换为多条数据记录存储在数据库的表中，这样虽然实现数据的结构化存储管理，满足数据的查询和分析。但面对海量数据存储管理时，这种结构化转换导致数据记录翻倍增长，NUM、TIME、TICV 等数据列出现大量冗余存储，使得数据存储容量增大且降低了数据管理系统性能。同时，传统关系型数据库的范式要求也限制了数据表的结构，当监测数据中数据内容列进行增加或删除时需要重新设计数据库表结构。

针对 GNSS 系统时差原始监测数据的半结构化和海量数据记录的特点，结合数据用户需求和科学数据分析场景需求，本文提出了基于 HBase 分布式数据库的 GNSS 系统时差原始监测数据存储模型。根据图 4 中的数据格式，设计了 HBase 数据管理系统中表的结构，存储模型如图 5 所示。

RowKey ( TIME+SUM+ PRNS )	Column family ( Parameter )								
	002 RCLK	002 SCLK	...	003 SCLK	003 RCLK	...	005 RCLK	005 SCLK	...
201910032350010 070020030060170 19022028	-105.651	-95 478.252	...	-90.559	-5 409.731	...	/	/	...
201910040017050 070020030060120 17019023	-95.580	-95 481.155	...	-85.203	-5 412.224	...	/	/	...
201910040250120 080020050060090 12017019025	-84.060	-95 488.614	...	/	/	...	-65.984	-595.450	...
RowKey ( TIME+SUM+ PRNS )	Column family ( Parameter )								
	006 SCLK	006 RCLK	...	009 RCLK	009 SCLK	...	012 RCLK	012 SCLK	...
201910032350010 070020030060170 19022028	-91.764	-23 499.378	...	/	/	...	/	/	...
201910040017050 070020030060120 17019023	-92.243	-23 505.452	...	/	/	...	-17.987	59 255.486	...
201910040250120 080020050060090 12017019025	-109.988	-23 539.626	...	-62.399	-14 230.682	...	-59.365	59 242.058	...
RowKey ( TIME+SUM+ PRNS )	Column family ( Parameter )								
	017 RCLK	017 SCLK	...	019 SCLK	019 RCLK	...	022 RCLK	022 SCLK	...
201910032350010 070020030060170 19022028	-63.322	41 225.824	...	-34.969	-77 535.866	...	-11.658	-228 085.188	...
201910040017050 070020030060120 17019023	-69.317	41 230.970	...	-29.289	-77 534.799	...	/	/	...
201910040250120 080020050060090 12017019025	-64.621	41 252.965	...	-45.751	-77 524.699	...	/	/	...

图 5 GNSS 系统时差监测原始数据 HBase 存储逻辑视图

RowKey (TIME+SUM+ PRNS)	Column family (Parameter)						Column family (EP)
	023 SCLK	023 RCLK	...	025 SCLK	025 RCLK	...	TIC
201910032350010 070020030060170 19022028	/	/	...	/	/	...	321.202
201910040017050 070020030060120 17019023	-109.353	-49 000.276	...	/	/	...	320.931
201910040250120 080020050060090 12017019025	/	/	...	-17.037	-4 318.001	...	314.625

续图 5

基于数据常用的读场景即基于时间、卫星号和少数情况下的可视卫星数进行检索查询,表的 Row Key 由数据采样时间 (TIME)、可视卫星总数 (SUM) 和可视卫星的编号集合 (PRNS) 组合而成;根据 GNSS 系统时差原始监测数据来源分别定义了两个列族,反映卫星参数的 Parameter 列族和其他数据来源的 EP 列族。Parameter 列族中采用卫星编号和卫星参数组合的方式定义多个列,EP 列族中定义了一个计数器值的列。这些列标识可以在建表时创建,也可以在表创建完以后动态插入数据时添加。但列标识名字的长度和列族名字的长度都会影响 I/O 的读写性能和发送给客户端的数据量,所以它们的命名应该遵循简洁的原则。

## 2.2 基于 InfluxDB 的时序数据存储模型

### 1) 基于 InfluxDB 的时序数据表存储模式

InfluxDB 是基于时间序列数据索引实现高效数据查询,索引是以 Measurement 表中数据记录的时间戳 (time stamp) 和数据属性 (Tags) 组合的方式建立。时间频率科学数据中的具有时间序列特征的数据主要来源于业务系统中的测量比对数据、监控数据,数据结果由元数据信息和包括采集时间、多个测量参数的数据记录组成,数据模型如表 3 所示。

表 3 时间频率测量数据的 InfluxDB 表存储模式

数据记录	采集时间	测量参数 1	测量参数 2	...	测量参数 $m$
1	$T_1$	$V_{11}$	$V_{21}$	...	$V_{m1}$
2	$T_2$	$V_{12}$	$V_{22}$	...	$V_{m2}$
...	...	...	...	...	...
$n$	$T_n$	$V_{1n}$	$V_{2n}$	...	$V_{mn}$

注:元数据信息:设备 ID,参考源 ID,观测站 ID,设备基本信息参数,...

在时间频率测量比对中存在大量基于不同测量比对技术的同源测量结果数据,即元数据信息中参考源 ID、观测站 ID 相同,而对应的设备 ID、设备基本信息参数不同;数据记录中的采集时间和测量参数也基本相同。对于 InfluxDB 的数据表,如果每条数据记录都包含一次元数据信息中的设备基本信息,将造成数据的大量冗余,且影响数据记录的查询效率。针对这一特点,可采用时间序列数据库与关系型数据混合存储管理模式。

### 2) 基于 InfluxDB 的溯源比对数据存储模型

溯源比对数据和系统监测数据具备典型的时间序列特征,传统的关系型数据库存储管理方法,在应对千万或上亿条数据记录存储管理时性能无法数据分析要求。以某系统的系统时间主钟溯源比对链路为



例,溯源比对数据和溯源系统监测数据根据溯源比对系统类型分为共视溯源比对和光纤溯源比对两个数据表进行存储,对应的关系型数据存储模型如表4和表5所示。

表4 光纤溯源比对钟差数据模型

ID	TIME	OP_DATA	OP_CPU	OP_MEM	OP_STATE
1	20161223001000	-1.983	0.40	0.20	0
2	20161223001001	-1.980	0.10	0.14	0
3	20161223001002	-1.975	0.12	0.16	0
4	20161223001003	-1.964	0.14	0.16	0
5	20161223001004	-1.993	0.16	0.15	0

表5 GNSS common-view 比对数据模型

ID	TIME	CV_DATA	CV_CPU	CV_MEM	CV_STATE
1	20161223001000	1.550 0	0.16	0.20	0
2	20161223002600	3.028 6	0.28	0.24	0
3	20161223004200	1.740 0	0.25	0.36	0
4	20161223005800	2.750 0	0.20	0.31	0
5	20161223011400	2.483 3	0.19	0.295	0

存储在 InfluxDB 中的数据通常是基于时间的数据点序列,包括时间戳和其他 Fields、Tags。结合溯源比对数据特征及 InfluxDB 时序数据库表 Tag 与 Field 字段的查询特点,设计了某系统的系统时间溯源比对数据表存储模式,如图6所示。

name: Trace					
time stamp	Fields			Tags	
time	clock	cpu	mem	type	state
20161223001000	-1.983	0.40	0.20	op	0
20161223001000	1.550 0	0.16	0.20	cv	0
20161223001001	-1.980	0.10	0.14	op	0
20161223001002	-1.975	0.12	0.16	op	0
20161223001003	-1.964	0.14	0.16	op	0
20161223001004	-1.993	0.16	0.15	op	0
20161223001001	-1.980	0.10	0.14	op	0
...	...	...	...	op	...
20161223002600	-1.131	0.12	0.21	op	0
20161223002600	3.028 6	0.28	0.24	cv	0
...	...	...	...	op	...
20161223004200	-1.277	0.15	0.16	op	0
20161223004200	1.740 0	0.25	0.36	cv	0
...	...	...	...	op	...
20161223005800	-1.047	0.16	0.18	op	0
20161223005800	2.750 0	0.20	0.31	cv	0
...	...	...	...	op	...
20161223011400	-1.000	0.13	0.19	op	...
20161223011400	2.483 3	0.19	0.295	cv	0

图6 基于 InfluxDB 的比对数据表存储模式

针对某系统的系统时间主钟两种溯源比对数据和溯源监测数据特点,将没有被索引的列 clock、cpu 和 mem 定义为 Fields,将可用于索引的列 type 和 state 定义为 Tags。表 Trace 中有 type 和 state 两个 Tag Key,其中 type 对应的维度值 Tag Values 为{op, cv}, state 对应的维度值 Tag Values 为{0, 1}, Tags 组合作为记录的主键,通过 time 进行数据过滤得到一条数据记录;表中有两个数据源 (op, cv),每个数据源有 3 种指标 clock、cpu 和 mem。

### 3 时间频率科学数据存储策略实践

#### 3.1 HBase 数据存储测试与分析

测试数据采用 GNSS 系统时差监测原始实际观测数据,平台分别部署了 4 个节点的 HBase 集群和四个节点的 MySQL 集群,各节点硬件配置均为 2 核 CPU (Intel (R) Xeon (R))、16 G 内存、200 G SDD 云盘。HBase 集群在搭建的 Hadoop 中部署 HBase1.3.5 组件,MySQL 集群在配置相同的 4 台服务器上部署 Centos 7.5 操作系统以及 JDK 1.8,MySQL 5.5,MyCat 1.6 等组件。测试分别对比分析了 HBase 与 MySQL 集群数据导入性能和 HBase 集群的读/写和并发性能。

##### 1) GNSS 系统时差监测数据 HBase 导入性能测试

测试使用 HBase Shell Put 接口将实际观测的 2019 年共 1 亿条 GNSS 系统时差监测原始数据记录分别写入设计的 HBase 表和 MySQL 集群数据库表中,比对验证所设计存储模式的数据导入性能,测试结果如图 7 所示。

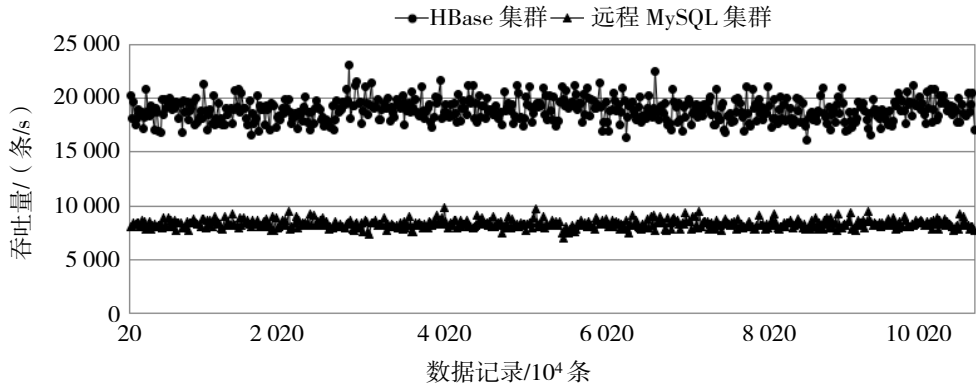


图 7 HBase 与 MySQL 集群数据导入性能

由图 7 可知,GNSS 系统时差监测数据 HBase 存储集群的平均吞吐量达到 18 932 条/s,MySQL 集群平均吞吐量达到 8 290 条/s,HBase 集群的平均吞吐量是 MySQL 集群平均吞吐量的 2 倍。

##### 2) GNSS 系统时差监测数据 HBase 集群读/写性能测试

测试采用云服务基础测试工具 YCSB 验证 HBase 集群的频繁数据读取和更新操作请求的性能。模拟多站、多地 GNSS 系统时间监测和多用户数据服务应用中海量数据快速读/写操作。测试中对本文设计实现的 GNSS 系统时差监测数据 HBase 表执行 300 000 次的访问操作,包括 50%的数据查询操作和 50%的数据写入操作,每组测试重复执行了 10 次。测试结果中的平均访问延迟和吞吐量的变化情况如图 8 所示。

从图 8 可看出,GNSS 系统时差监测数据 HBase 存储系统在数据读取和更新操作频繁的应用场景下,重复 10 次的读/写测试平均访问时延与系统吞吐量变化平缓、系统运行稳定。测试中 10 次 Read 平均访问时延为 279  $\mu$ s、Update 平均访问时延为 568  $\mu$ s,整体上 HBase 集群的 Read 操作性能高于 Update 操作性能,HBase 更适用于历史数据快速查询场景。

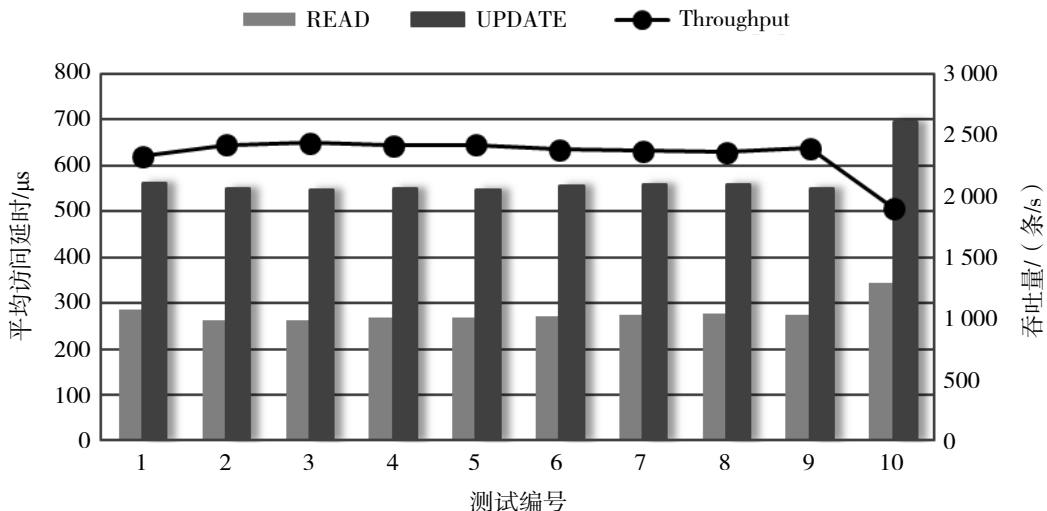


图 8 GNSS 数据存储系统读写操作平均延迟

### 3) GNSS 系统时差监测数据 HBase 集群只读测试

测试针对 GNSS 系统时差监测数据服务和分析型应用中大量的数据读取、少量的数据更新或写入场景，测试了 HBase 集群性能。测试通过设定 YCSB 对 GNSS 系统时差监测数据 HBase 表执行包括 95% 查询和 5% 写总计共 300 000 次访问操作，每组测试重复 10 次，测试结果中的平均访问延迟和吞吐量的变化如图 9 所示。

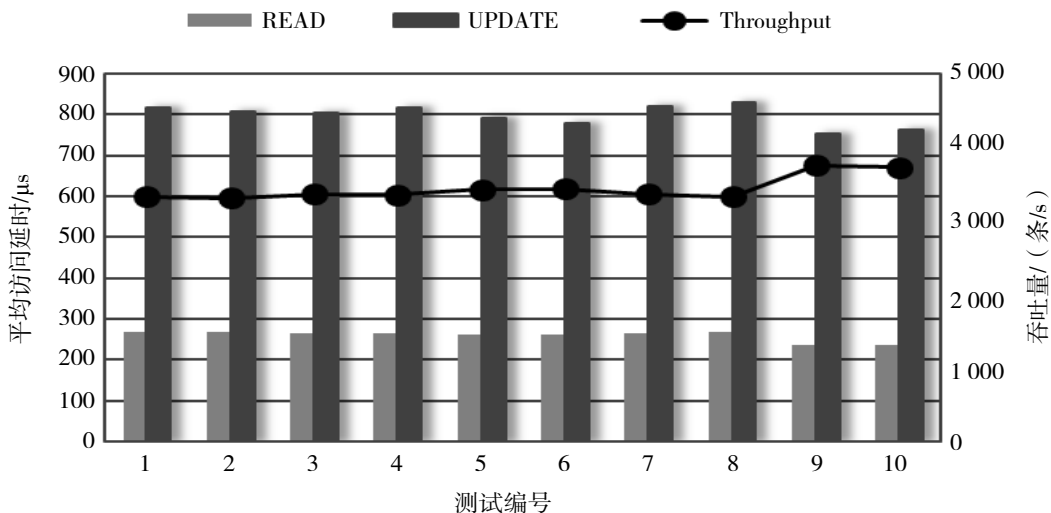


图 9 GNSS 数据存储系统只读平均延迟

从图 9 中可以看出，对于 GNSS 系统时差监测数据 HBase 存储集群，在频繁数据读取操作场景，测试中 10 次只读操作的平均访问延迟和系统吞吐量平稳、平均访问时延 259 μs、平均吞吐量为 3 438 条/s，数据读取操作平均延迟远小于更新操作平均延迟。

### 4) GNSS 系统时差监测数据 HBase 并发访问性能测试

为了验证 GNSS 系统时差监测数据 HBase 存储系统多用户并发访问性能，测试采用 YCSB 设定 HBase 集群并发线程数量由 10 个增至 100 个，每个线程分别对 HBase 集群的 GNSS 系统时差监测数据执行 300 000 次的读取、更新请求，包括 50% 的 Read 操作和 50% 的 Update 操作，测试结果中平均吞吐量以及访问时延的结果如图 10 所示。

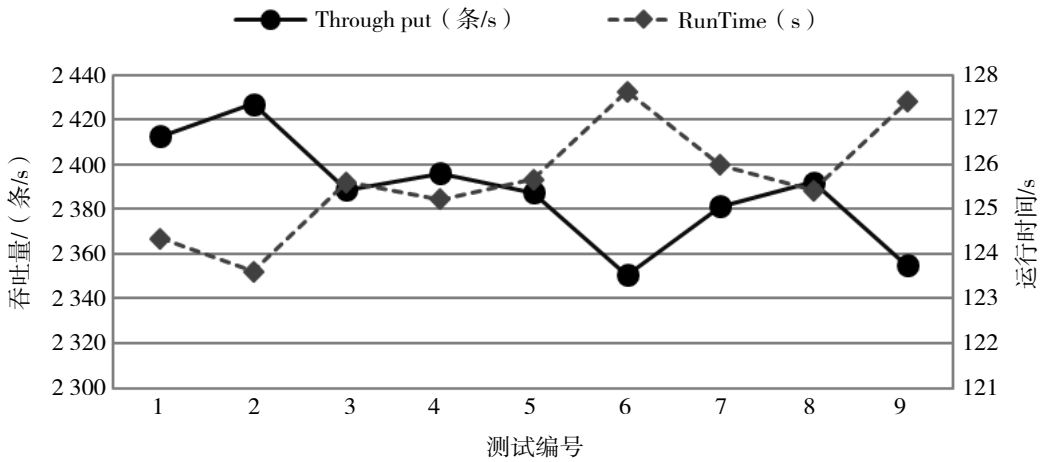


图 10 GNSS 数据存储系统并发访问性能

从图 10 中可以看出，当系统并发访问线程的不断增多时，系统吞吐量和平均执行时间趋于平稳，未出现明显的性能变化。测试结果表明 HBase 数据存储管理系统在高并发访问场景中系统性能稳定，适合历史数据高并发快速查询访问场景。同时，本测试方法和测试结果为生产环境下 HBase 数据库管理系统集群并发访问数据量和性能变化趋势提供参考。

### 3.2 InfluxDB 数据存储测试与分析

测试以某系统的系统时间主钟溯源比对数据为例，在所搭建的服务器平台上分别部署单机版的 InfluxDB、MySQL 和 4 节点的 MySQL 集群。测试平台的 4 个节点服务器配置均为 2 核 CPU (Intel (R) Xeon(R))16 G 内存,200 G SDD 硬盘,操作系统使用 Centos 7.5,部署的组件或软件包括 JDK 1.8,Influxdb 1.5.2, MySQL 5.5, MyCat 1.6。

主钟的光纤溯源比对数据是时间频率科学数据中典型的时间序列数据，数据来自于实测数据，测试数据记录总计 9 百万条。将数据存入表 3~5 所示的 InfluxDB 表和 Mysql 数据表中，比对验证了 3 种存储管理方法数据写入和查询的性能。

#### 1) 溯源比对数据 InfluxDB 导入性能测试

在本地客户端将数据文件通过访问接口，将数据一次性批量导入，以 10 万条数据记录递增的方式记录数据库各节点数据每秒平均导入数据记录的个数，测试结果如图 11 所示。

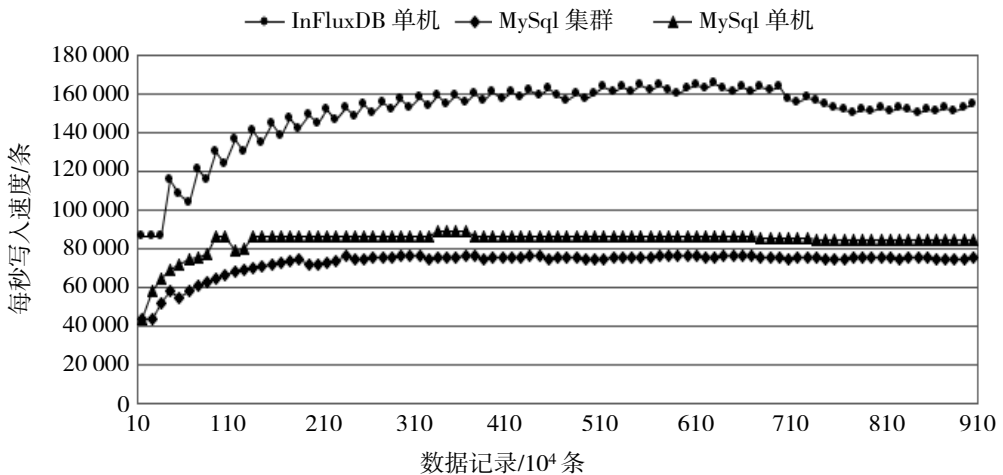


图 11 InfluxDB 数据导入性能

InfluxDB 数据库表中统计的平均每秒写入数据记录达 149 587，MySQL 单机的平均每秒写入数据记

录达 84 133, MySQL 集群的平均每秒写入数据记录达 72 789。InfluxDB 的每秒写入速度远大于单机 MySQL 和 MySQL 集群, 写入速度是近 MySQL 数据库的 2 倍。

## 2) 溯源比对数据查询性能测试

测试分别对 InfluxDB 和 MySQL 表中的数据记录进行查询访问, 查询策略是随机读取某一天中 1 h 的数据记录, 即随机查询 86 400 个数据记录中的 3 600 个数据点, 每次查询重复执行 1 000 次并计算查询时间的平均值, 测试结果如图 12 所示。

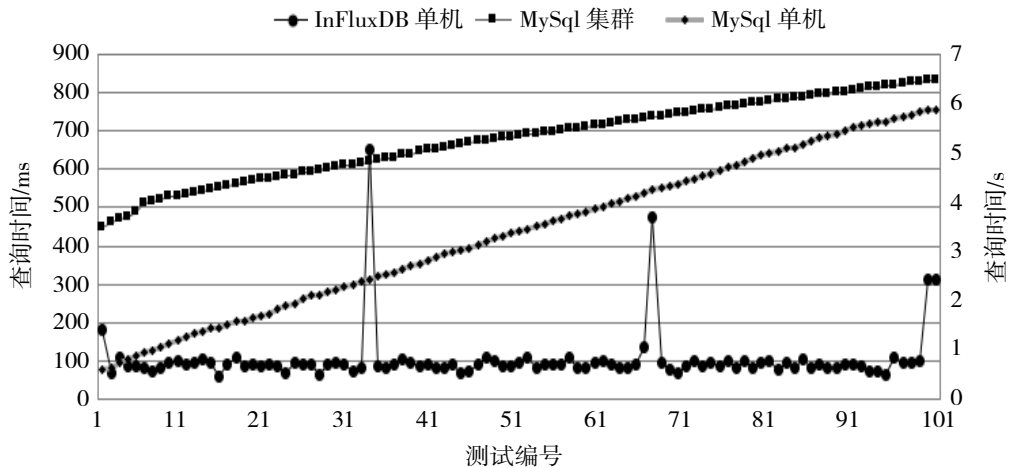


图 12 InfluxDB 数据查询性能

InfluxDB 的平均查询时间为 105 ms、MySQL 单机平均查询时间 3 371 ms、MySQL 集群的平均查询时间为 674 ms。MySQL 单机平均查询时间是 InfluxDB 的 33 倍、MySQL 集群平均查询时间是 InfluxDB 的 6 倍, 且 MySQL 单机和 MySQL 集群在连续查询场景下查询时间递增。测试中 InfluxDB 查询时间出现的突跳点, 主要原因是由 GO 语言开发的内存型数据库 InfluxDB, 内存积累到临界点后会执行内存回收操作影响性能导致的查询时间抖动。测试结果表明 InfluxDB 的数据查询性能明显且远优于 MySQL 集群和 MySQL 单机且其性能更加稳定, 更适合作为时间频率科学数据中时间序列数据的存储引擎。

## 4 结论

为解决海量时间频率科学数据存储管理面临的挑战, 提出了基于 NoSQL 的时间频率科学数据混合存储策略。在时间频率科学数据特征和 NoSQL 数据库特征基础上, 构建时间频率科学数据混合存储策略和存储模型。采用某系统的时间频率系统运行过程中积累的 GNSS 系统时差监测原始数据、溯源比对数据对传统关系型数据库和混合策略数据存储、并发效率进行测试。结果表明, 本文提出的 HBase 数据存储方法的平均写入性能是 MySQL 集群的 2.1 倍、高并发和读写性能在任务量增加时性能平稳; 基于 InfluxDB 的溯源比对数据平均写速度是 MySQL 集群数据库的 2.3 倍、查询速度是 MySQL 集群的 6.4 倍。文章提出的时间频率科学数据混合存储策略, 能够为海量时间频率科学数据分析、数据共享及信息服务提供高效和及时的服务, 为时间频率科学大数据知识获取和科学发现提供技术支撑。

## 参考文献:

- [1] 王玉琢, 王伟波, 张爱敏. 守时原子钟数据交互系统[J]. 计算机系统应用, 2016, 25(9): 119-123.
- [2] 王燕平, 董绍武, 袁海波. 时间频率科学数据库的建设和应用[C]//2009 年全国时间频率学术会议论文集, 成都:2009 全国

时间频率学术会议组委会, 2009: 554-558.

- [3] 高海军. 时间频率科学数据库系统设计与实现[D]. 成都: 电子科技大学, 2013.
- [4] 杨军. iGMAS 数据中心 MySQL 高可用系统的研究与实现[D]. 西安: 西安电子科技大学, 2019.
- [5] 徐道柱. 面向服务的空间数据管理关键技术研究[D]. 郑州: 解放军信息工程大学, 2017.
- [6] 黎建辉, 李跃鹏, 王华进, 等. 科学大数据管理技术与系统[J]. 中国科学院院刊, 2018, 33(8): 796-803.
- [7] AN T. Science opportunities and challenges associated with SKA big data[J]. Science China (Physics Mechanics & Astronomy), 2019, 62(8): 121-126.
- [8] 关兆雄. 面向非结构化的分布式存储系统的性能分析系统研究[J]. 自动化与仪器仪表, 2018, 34(2): 40-43.
- [9] RAO T R, MITRA P, BHATT R, et al. The big data system, components, tools, and technologies: a survey[J]. Knowledge and Information Systems, 2019, 60(3): 1165-1245.
- [10] 施晓峰. 基于分布式 NoSQL 数据库的档案大数据存储与检索方案研究[J]. 计算机应用与软件, 2019, 36(5): 15-20.
- [11] 季一木, 张宁, 尧海昌. HOS: 一种基于 HBase 的分布式存储系统设计与实现[J]. 南京邮电大学学报(自然科学版), 2019, 39(5): 63-71.
- [12] CHEN L X, ZHANG N, SUN H M, et al. Secure search for encrypted personal health records from big data NoSQL databases in cloud[J]. Computing (prepublish), 2019: 1-25.