

# 大型桥梁健康监测系统数据存储优化设计

王代君, 华鹏

(江西省交通运输科学研究院有限公司, 江西 南昌 330052)

**摘要:** 大型桥梁健康监测系统具有数据量大、采集频率高、架构复杂等特点, 为计算机资源的合理利用带来了巨大的挑战。该文根据桥梁健康监测系统的应用特点, 设计了一套高效的数据管理方案, 在数据的存储位置、存储方式和存储时限等方面进行了优化。为了解决硬盘空间占用率和数据监测可靠性之间的矛盾, 提出了“分层统计”的概念, 并综合运用了重采样和数据压缩等技术, 大大提高了数据管理的科学性。

**关键词:** 大型桥梁; 健康监测; 系统集成; 数据存储

对于大型桥梁, 健康监测是保证桥梁结构安全的关键措施。但桥梁结构状态的识别是建立在大量监测数据的基础上, 因而大型桥梁一般都安装了几百至上千个不同类型的传感器, 将同时产生海量的监测数据, 这给桥梁健康的在线监测带来了很大的技术挑战。为了解决监测系统的大数据存储问题, 目前一些学者采用了云计算、Hadoop、ODPS、多级缓冲等方法来提高各类监测系统的数据存储效率, 但这些方法无疑直接增加了系统的开发成本。从文献检索结果来看, 目前专门针对桥梁健康监测系统大数据存储管理的文献并不多见。该文以九江长江公路大桥的监测系统维护为例, 设计一套低成本、高效率、高可靠的大型桥梁健康监测系统数据存储管理的优化方案, 希望对桥梁健康监测系统的相关研究提供帮助。

## 1 工程概况

九江长江公路大桥全长 25 km, 主桥长 1 405 m, 为双塔混合梁斜拉桥, 主跨 818 m, 于 2013 年正式通车, 其结构健康监测系统也同时投入运营。九江长江公路大桥健康监测系统采用了 B/S 和 C/S 相结合的设计理念, 将查询和访问功能放在 B/S 架构体系中, 而需要进行大量运算的数据处理与采集控制则由 C/S 架构软件完成。全桥共安装包括振动、FBG、GPS、振弦式应变、压力变送器等在内的 13 种传感器, 共计 525 个(表 1)。

经过4年的连续运行, 九江长江公路大桥监测系

表 1 九江长江公路大桥主要传感器数据量统计

序号	传感器类型	数量/ 个	保存物 理量数 量/个	采集 频率	平均数 据量/ (个·s <sup>-1</sup> )
1	索力加速度计	110	3	100 Hz	33 000
2	伺服加速度计	55	2	100 Hz	11 000
3	强震仪	3	2	100 Hz	600
4	光纤温度计	72	2	1 Hz	144
5	光栅应变计	142	2	1 Hz	284
6	GPS	6	3	5 Hz	90
7	锚索计	12	3	1 Hz	36
8	风速仪	4	3	4 Hz	12
9	风压计	11	2	1 Hz	22
10	压力变送器	24	3	1 Hz	72
11	振弦式应变计	75	2	10 min	0
12	拉绳式位移计	3	2	1 Hz	6
13	温湿度计	2	2	10 min	0

统也随之进入一期维护阶段, 原有软件系统在对所有监测数据进行保存的情况下存在 CPU 占用率高、运行内存空间占用大、数据通信压力大、界面显示卡顿等问题。另外, 由于原始数据是以二进制文件存储的, 在提取数据时比较麻烦。该文将对上述问题进行优化。

从表 1 可以看出: 数据量最多的传感器每秒钟可产生几万个采样数据, 而数据量少的则可忽略不计。这些数据被采集出来后通常以原始数据、调试数据、统计数据 and 界面数据 4 种形式复现并存储, 如果全部数

据都通过网络传输并存储起来,不但会给软件的通信过程带来巨大压力,也会严重消耗工控机和服务器的硬盘空间,还会造成 WEB 端的数据刷新出现卡顿现象。因此很有必要重新制定一个高效、方便、可靠的数据存储管理体系。

2 数据管理方案设计

2.1 数据库选用

目前主流数据库主要有 SQL Server、Oracle、MySQL 和 SQLite 等。根据大型桥梁健康监测系统的应用特点,采集站的原始数据量非常大,但并发性不高,而且需要满足可查询的功能,这些要求与 SQLite 的轻量级、单一文件等特点十分吻合,因此采集站的原始数据保存选用 SQLite 数据库。对于数据服务中心,由于数据需要永久保存,并且要求为管理者的决策提供数据支持,对数据的查询和分析要求非常高,考虑到服务器一般具有较大的硬盘容量,因此 Oracle 数据库成为最佳选择。

2.2 数据管理方案设计

该文对九江长江公路大桥健康监测系统的数据存储方式进行了调整,包括原始数据、调试数据、统计数据和显示数据的存储优化,总体方案如图 1 所示。

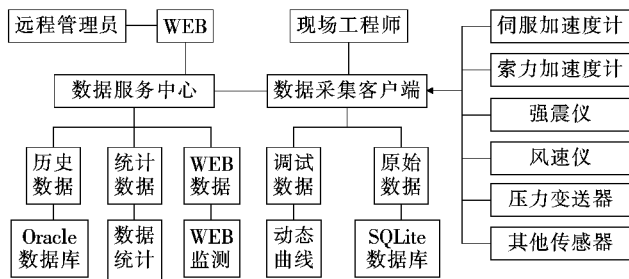


图 1 数据存储优化总体方案

首先由运行在各采集站的客户端软件对所有传感器进行数据采集,采集到的原始数据直接保存一份至 SQLite 数据库中,然后分为两路,一路经过重采样后以动态曲线的形式供调试用;另一路经过有效性检测、物理量换算、数据统计等复杂处理过程后发送至数据服务中心,由服务端程序接收并保存至 Oracle 数据库的相应表中,作为前台 WEB 界面的数据源。考虑到调试数据与界面数据并不参与任何计算,仅仅是用于显示观察,为了降低界面刷新频率,减少 CPU 资源占用,采用不大于 1 Hz 的频率进行暂存。

各类数据存储方式见表 2,具体说明如下:

表 2 各类数据存储方案设计

数据类型	数据功能	保存频率	保存位置
原始数据	原始备份	100%保存	采集站 SQLite 数据库
调试数据	安装调试	≤1 Hz	采集站内存
统计数据	历史查询	100%保存	数据服务中心 Oracle 数据库
界面数据	WEB 监控	≤1 Hz	数据服务中心 Oracle 数据库

(1) 原始数据的保存

原始数据是指采集客户端直接采集回来的各类传感器数据,原始数据主要用于问题追溯,当对数据处理结果有疑问时,可通过查询原始数据进行问题确认。原始数据的特点是数据量大,并且没进行过任何的处理直接存储在计算机硬盘中。根据 JT/T 1037—2016 的要求,后处理数据保持不少于 3 个月。考虑到程序对原始数据的后处理可能出现异常,因此在桥梁健康监测工程应用中,通常保存近 3 个月左右的原始数据。

大型数据库虽然功能强大,但对于客户端的原始数据存储而言就显得功能严重过剩,并且会给管理者带来额外的数据库维护工作。二进制存储方式由于空间占用非常小,已被许多监测系统所采用,然而二进制的存储是基于计算机存储结构来考虑的,它并不支持对文件中的数据进行高效筛选与检索,只有通过第三方软件编程才能完成数据查询。而 SQLite 数据库作为轻量级的文件型数据库,可以在很大程度上弥补二进制存储方式的不足,另外,SQLite 是一种通用数据库,具备最基本的数据库存储结构,并且可以通过 SQL 语句直接对文件的数据进行筛选与初步分析。这一点是二进制文件所无法比拟的。为了解决 SQLite 数据库文件占用存储空间较大的问题,该文对其进行了压缩处理,压缩后的文件大小比原来减少了 90%。对于采集频率大于 1 Hz 的传感器,按 1 Hz 对原始数据进行重采样后发送至数据服务中心。对于根据原始数据计算而得的物理量,则全部发送至数据服务中心。

(2) 调试数据的处理

调试数据用于现场安装实施人员调试传感器,当传感器安装完毕后,需要对传感器进行调试与校准,这就需要传感器数据以表格、图形或曲线的形式动态、直观地显示出来。显然,这些数据不需要保存到硬盘,只需暂存在内存中,数据一旦显示出来,就会被后续的数据所覆盖。然而图表插件的调用需要占用较多 CPU 资源,为了减轻资源消耗,对于数据量特别大的传感器,对其原始数据进行 1 Hz 的重采集再显示出

来,满足数据观察要求即可。

(3) WEB 监测数据的保存

WEB 监测数据是为了远程管理员监测传感器实时数据而在浏览器中显示的数据。由于 WEB 页面中的数据也是以数据有无和数据变化趋势的判断为目的的,因而也不需要全部数据进行显示。对于采集频率大于 1 Hz 的传感器,按 1 Hz 进行重采样进行显示。

(4) 分层统计数据的保存

如果数据服务中心仅保存重采样之后的数据,很有可能将一些超标的的数据忽略;而如果为了数据可靠而将所有原始数据都发送到数据服务中心保存,又会占用大量的硬盘空间。为了解决硬盘空间占用率和数据监测可靠性之间的矛盾,该文提出了对海量数据进行“分层统计”的概念,即在采集客户端中将原始数据和各物理量按每 10 min、每 0.5 h 和每天 3 个尺度进行统计。尺度越大,统计粒度也越大。为了使统计值尽量精确地反映原始值,应采用较小的统计尺度,于是采用 10 min 统计,每 1 h 可以取得 6 组统计值;按小时统计和按天统计,一方面是考虑到时间段的完整性;

另一方面是为统计某些物理量的“时均值”、“日均值”等常用参量的变化情况提供数据源。分层统计过程中主要统计最大值、最小值、平均值、中值、均方差等参数,并将这些参数发送至数据服务中心,而原始数据则留在客户端所在的工控机中进行备份。这样就将层内的大量数据转化为少数几个特征参数,从而大大减少存储空间占用。当数据超标时,必然会在最大值或最小值中体现出来,这就解决了数据可靠性的问题。

(5) 历史数据的保存

分层统计数据发送至数据服务中心后,由服务端程序将其保存至数据库表中,作为永久保存的数据。在进行问题追溯时,如果原始数据由于超过保存期限而被删除,就可以在历史数据中进行检索,查得相应时间段的分层统计参数,也能在很大程度上对问题进行反推。

3 数据管理方案实现

在程序其他模块逻辑不改变的情况下,对系统软件的运行参数进行对比,如表 3 所示。

表 3 优化前后运行参数对比

项目	CPU 占用率/%		内存占用/M		总数据量	
	采集站 工控机	数据服务中 心服务器	采集站 工控机	数据服务中 心服务器	采集站硬盘 占用(每天)/G	数据服务中心数据 库行数(每秒)
优化前	11	8	386	181	14	$1.45\times10^9$
优化后	2	0	75	63	2.03	$1.95\times10^7$

从表 3 可以发现,优化后的系统软件在运行过程中 CPU 占用率、内存占用和总数据量都大幅减小,优化效果明显。

4 结论

在原有监测系统其他功能模块逻辑不改变的情况下,该文的优化方案使 CPU 占用率、内存占用和总数据量都大幅减小。虽然数据服务中心没有保存完整的原始数据,但对分层统计后的主要特征值进行了永久保存,因此这是一个兼顾硬盘容量和数据可靠性的优化方案。另外,在对系统进行优化的过程中,并未引进新的硬件设备和复杂的算法,未明显增加维护成本,达

到了预期的设计目标。

参考文献:

[1] 朱永利,李莉,宋亚奇,等.ODPS 平台下的电力设备监测大数据存储与并行处理方法[J].电工技术学报,2017(9).  
[2] 刘黎志,吴云韬.环境空气质量监测大数据存储模式研究[J].环境科学与技术,2016(5).  
[3] 易诗.基于 MongoDB 的光纤传感海量监测数据存储方法研究[D].武汉理工大学硕士学位论文,2015.  
[4] 陶悦.新光大桥健康监测数据处理方法与应用研究[D].华南理工大学硕士学位论文,2012.  
[5] 张申宇,邱雷.导波结构健康监测系统软件数据管理模块设计[J].国外电子测量技术,2017(2).  
[6] 侯立群,欧进萍,李宏伟,等.东营黄河公路大桥振动监测系统设计与实现[J].世界桥梁,2008(3).