

# 电网大数据量表设计优化技术及应用<sup>①</sup>

陈 刚, 郑浩泉, 吴淑伟

(国网电力科学研究院, 南京 211100)

**摘 要:** 阐述了在数据库设计阶段实施大数据量表设计优化工作的必要性. 介绍了电网大数据量表的概念及产生原因, 并针对因大数据量表而引发的数据库性能问题, 提出了大数据量表设计优化相关的关键技术. 在设备状态检修系统的数据库设计阶段, 以设备状态评价功能为例进行实践应用. 仿真实验表明对大数据量表实施设计优化后系统性能得到了显著提升. 为后续系统稳定运行提供保障, 也为类似的大数据量表设计优化应用提供了参考.

**关键词:** 大数据量表; 设计优化; 状态评价; 基于反范式的表设计; 基于表分类的存储设计

## Large-Scale Grid Data Table Design Optimization Technology and Application

CHEN Gang, ZHENG Hao-Quan, WU Shu-Wei

(State Grid Electric Power Research Institute, Nanjing 211100, China)

**Abstract:** This paper describes the necessity of implementing design optimization for the large-scale data table in the database design stage. It introduces the concept and causes of the large-scale data grid table. And for database performance problem caused by the large-scale data table, it proposes design optimization technology of the large-scale data table. On database design stage of equipment condition-based maintenance system, it uses these technology in device status evaluation function as an example. The simulation experiment results show that system performance has been significantly improved after design optimization for the large-scale data table. To provide protection for the subsequent stability of the system, it is also referable for those large-scale data table need to be optimized.

**Key words:** large-scale data table; design optimization; state evaluation; table design based on anti-paradigm; storage design based on classification table

随着国家电网公司企业信息化深入应用, 业务系统的数据量进入了爆炸式增长. 大数据量表带来的数据库性能问题已经成为业务系统面临的现实问题. 如果前期数据库设计存在缺陷, 则给后续运行与维护带来很大成本, 并且很难通过性能优化工作来提高系统性能. 因此, 在系统数据库设计阶段, 对大数据量表的逻辑结构和物理结构进行设计优化, 使之在满足业务需求的条件下, 解决系统因大数据量表而引发的性能问题<sup>[1-3]</sup>.

本文基于关系型数据库, 针对公司信息化高速发展过程中因大数据量表而引发的数据库性能问题, 介

绍了大数据量表设计优化的相关技术. 在新建的设备状态检修系统的数据库设计阶段, 对设备状态评价功能中进行实践应用. 仿真实验证明运用本文介绍的大数据量表设计优化技术对系统性能提升的实用性.

## 1 电网大数据量表概述

### 1.1 电网大数据量表的定义

关系型数据库表的数据量大小是相对于所运行的软硬件环境而言. 如果表数据量在软硬件处理能力之外, 就可以说当前表数据量过大; 反之, 则说明表数据量适应当前软硬件条件. 一般情况下, 在当前电网

<sup>①</sup> 收稿时间:2014-05-07;收到修改稿时间:2014-06-19

业务系统运行环境中,关系型数据库表数据量达到百万级别及以上的表,实现复杂的业务逻辑都可能会出现性能问题.因此,这些百万级及以上的表可以称之为“电网大数据量表”.

### 1.2 电网大数据量表出现的原因

当前电网业务系统出现大数据量表主要有下面几个方面原因:第一,信息系统长期以来积累了大量的业务数据,这些数据包括台账、日志、表单、票据、量测值等数据信息;第二,业务应用系统逐步深化和细化使系统中的数据越来越全面,越来越精细,这导致数据量急剧增长;第三,电网各项业务融合的程度不断提高,大量孤立的业务应用系统融合贯通为覆盖面更广、功能更全的大系统.不同层级的业务应用系统之间在纵向交互的信息越来越多,这些数据汇总到一起后会变得十分庞大.

## 2 大数据量表设计优化相关技术介绍

数据库表逻辑结构优化通过增加、减少或调整表的逻辑结构来提高应用效率.数据库表的物理结构优化是为表设计合适的物理存储位置或存储结构,为检索推荐合适的存取路径,从而达到改进整个数据库应用的性能.下面将重点介绍大数据量表逻辑与物理结构设计优化相关技术.

### 2.1 基于反范式的表设计优化技术

反范式是通过冗余和派生列来减少连接操作、外码、索引数目;通过分割表来降低表的数据规模或减少表属性列宽,提高I/O的检索速度,从而达到提升系统性能的目的.但反范式可能会影响表的完整性,降低更新速度.一般常用的反规范技术方式有:增加冗余数据、增加派生列、分割表或者合并表<sup>[4-6]</sup>.

① 增加冗余数据:当两个或多个表在频繁的查询操作中经常需要进行连接时,可以在查询的主表上适当增加冗余数据列来避免或减少表之间的连接操作.但是这样需要同步维护冗余列来保证数据的一致性,所以建议对那些经常不变的属性列进行冗余;

② 设计合适的派生列:在进行数据查询汇总操作时,通过设计合适的派生列,可以大大缩减在数据汇总时的统计运算时间.达到提高查询效率的目的.

③ 分割表:一般分为水平分割和垂直分割.水平分割是将一个表按照行记录分割为多个表,减小表的数据规模,提高单表的查询速度.但由于查询、更新需

要路由选择不同的表,查询统计也要汇总多个表,增加了开发难度,程序相对会更加复杂.垂直分割是根据表属性列的访问频率进行分割.当某些属性列被访问的频率远高于其它属性列时,可将主键和这些属性列组合成一张表,而主键和其它属性列组合成另一张表.通过减少列的宽度,增加每个数据页的行数,一次I/O就可以扫描更多的数据行,从而提高了访问单表的速度.但由于造成了多表连接,所以建议在分割表中的列的查询或更新比较少的情况下使用.

④ 合并表:在查询过程中经常需要将几张表进行连接操作时,可以考虑将这几张相关表组合成一张表,减少连接操作,从而改善查询性能.

### 2.2 基于表分类的存储设计技术

数据表的分类有利于对不同类型的数据设计不同的存储模式.依据核心数据和过程数据的不同特点,数据表大致分为核心表和过程表 2 大类,核心表可分为恒数表、递增表;过程表分为流水表、状态表<sup>[7]</sup>.数据表分类定义如下表所示.

表 1 数据表分类

表类型	表说明
核心表	数据生命周期长,读比写的访问频率高,数据增长慢.
恒数表	数据量小,总是读,很少写.
递增表	数据量大,经常读,很少写.某些情况会频繁写.
过程表	数据生命周期短,写比读的访问频率高,数据增长快
流水表	数据量大,经常写,很少读.只插不改,定期删.
状态表	数据量大,经常写,经常读.边插边改边删.

在数据表存储设计时,应该遵循以下几个要点:

① 对于核心表设计,务必要遵循范式.主要考虑读操作性能,应该注意字段个数、字段长度以及查询范围.

② 对于过程表设计代表数据生命周期终止的字段.过程表生命周期短,从增删改的性能代价考虑,插入代价最小,修改需要检索数据,删除最为昂贵.使用逻辑修改代替删除.

③ 恒数表类似我们使用字典表.它存储设计常见的误区是把数据分配参数设得太大,使得表所占用的空间是实际所需要空间的几倍甚至几十倍.数据分配参数设置太大不仅造成空间浪费问题,还会造成数据表的高水位.处于高水位下的恒数表与大数据量的递增表联合查询,全表扫描的浪费就很惊人,严重影

响访问性能。恒数表增长量小,不流动的特点,设计符合表大小的数据分配参数,较小的恒数表不建议建立索引来提高查询效率。

④ 对于递增表设计,重点需要区别过程数据和核心数据,合理设计分区方式和字段。递增表索引的键值选择性尽量高,谨慎使用复合索引,递增表的关联查询对数据库的逻辑读和物理读消耗很大,在进行存储设计时要按照关联关系设计合适的分区和索引。

⑤ 对于流水表设计,关键在于做好分区和索引设计。具体应遵循以下几个原则:

- 依据数据量大小,可以选择按天、月、年等不同的时间分区方式。
- 可考虑按业务类型或其他键值建立子分区,减少并发插入冲突。
- 流水表是按时间顺序插入的,因此在时间字段上不建议建立索引。
- 应尽量不建或者少建索引,以保证插入速度以及减少并发插入冲突。

⑥ 对于状态表设计需要考虑多状态的转换。状态表一般是指记录某一行行为的状态过程,数据生命周期很短,涉及数据的查询、修改以及写回滚。通常可按某种状态标示数据生命周期是否终止的字段来建立分区。

### 2.3 索引技术

对大数据量表设计时,选择合适的索引能够大大提高检索效率。一般选择索引过程中除了最好的单一属性索引,还需要考虑附加索引,以便有更好的选择方案。但有些情况下,索引虽然提升了查询效率,却严重降低了数据更新性能。因此,在选择一个新索引是否可用时,需要综合考虑查询和更新的性能组合效益。设计索引时要考虑以下几点关键问题:

- (1) 新增的索引是否会提升查询效率;
- (2) 考虑索引的类型以及是否可以与其他属性进行组合;
- (3) 是否影响表的更新操作效率;
- (4) 索引对存储方面的需求。

综合考虑上述关键点,确定是否新增一个索引。当一个已用索引的性能变得较差时,需要选择新的设计方案。大数据量表在创建索引时应该遵循以下原则:

- (1) 表的主键、外键必须有索引;
- (2) 索引字段应该建立 NOT NULL 约束;

(3) 经常在 where 字句中且过滤性很强的字段,特别是大数据量表的字段,应该建立索引;

(4) 经常与其他表进行连接操作的字段,应该建立索引。

(5) 如果大数据量表频繁进行 DML 操作,建议不要建立过多索引,并尽量不要将频繁修改的字段建立索引。

(6) 建立复合索引时,尽量考虑使用单字段索引来代替,原则如下:

(7) 复合索引第一个字段,一般是选择性比较高的且在 where 字句中常用字段;

- 几个字段经常以 AND 方式出现在 where 字句中,且单字段查询极少,可以选择建立复合索引,否则考虑单字段索引;
- 如果复合索引的几个字段经常单独出现在 where 字句中,则分解为多个单字段索引。

### 2.4 分区技术

使用分区设计是对大数据量表进行设计优化的首选方案。表的分区技术通过改善可管理性、性能和可用性,为数据库应用程序带来了极大的好处。一般表的分区设计时应具备如下原则<sup>[8]</sup>:

(1) 表数据量规模大小

当表的大小超过 1.5GB 至 2GB 或者表的记录数超过 1000 万的时候,都应考虑对表进行分区。

(2) 数据访问特性

基于大数据量表的大部分查询应用,只访问表中少量的数据。对于此类型的表进行分区,可充分利用分区排除无关数据来提高访问性能。

(3) 数据维护

按时间段批量删除数据,例如按月删除历史数据。可考虑进行分区,以满足维护的需要。

(4) 数据备份与维护

按时间周期对表空间的备份时,可将分区与表空间建立对应关系,实现按分区备份与恢复。

(5) 并行数据操作

对于经常执行并行操作(如 PARALLEL INSERT, PARALLELUPDATE 等)的表应考虑进行分区。

## 3 在设备状态检修系统中的应用

设备状态检修是国家电网公司生产管理系统的高级应用。通过获取并处理反映输变配设备健康状态的

状态量, 评价设备当前状况. 对状态劣化和趋势不良的设备及时发布状态预警消息, 并进行有效的原因分析, 提出检修决策建议和检修计划.

### 3.1 设备状态评价功能介绍

设备状态评价是在影响设备状态评价的状态量元素发生变化之后, 系统启用定时任务通过设备状态检修接口每次读取 1000 条状态量变化记录保持到设备状态量中间表中. 通过关联相关的评价导则对设备状态量、设备部件分别进行评价, 形成对应的设备评价结果. 最终, 根据设备评价结果给出设备决策建议. 具体业务流程如下图所示:

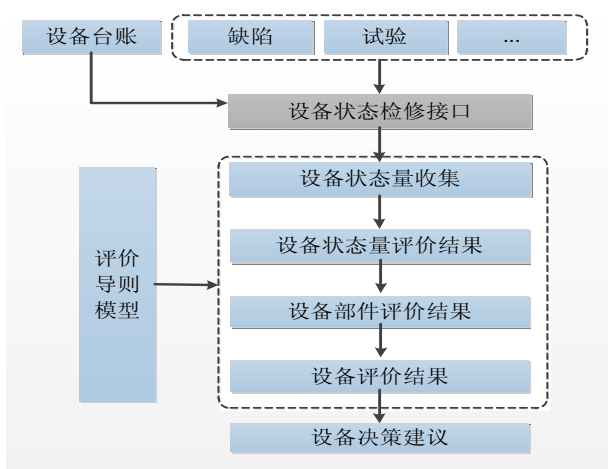


图 1 设备状态评价流程图

在实际运行过程中, 由于状态评价涉及的设备状态量元素众多, 随着数据积累造成状态评价功能中的设备状态量中间表、设备状态量评价结果表达到了千万级别. 设备部件评价结果、设备评价结果都是好几百万级别的. 这些都属于“电网大数据量表”. 由于状态评价业务过程逻辑复杂, 形成这些大数据量表之间的紧密关联. 当用户批量录入试验报告、缺陷记录等状态量元素进行动态评价时, 对这些大表的频繁变更操作造成了系统性能的急剧下降. 系统调度任务总是处于繁忙状态, 数据库资源的不能进行有效地释放, 甚至在前一个调度任务没有完成的情况下, 后续调度任务已经开始运行. 降低评价过程中数据规模, 减少大数据量表的关联和变更操作, 是解决状态评价性能问题的有效办法之一.

### 3.2 设备状态评价大数据量表的设计优化

为了降低设备状态评价过程中数据规模, 减少大数据量表的关联和变更操作, 我们进行以下几个方面

的设计优化工作:

① 利用反范式设计技术, 将原有设备评价模型按照输电、变电、配电三个专业分为三个不同专业的设备评价模型. 这样设计在满足了不同设备评价模型之间差异化需求的同时, 也降低了原有设备评价模型的数据规模.

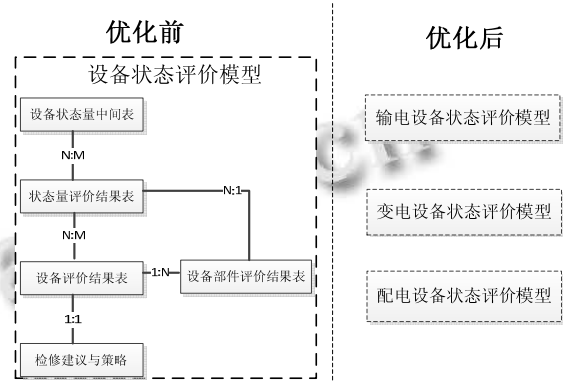


图 2 设备状态评价模型优化前后对比示意图

② 利用水平分表技术, 设计了最新状态量评价结果表、最新设备评价结果表, 从而避免每次设备评价过程中需要关联大数据量的历史数据表, 降低每次评价过程中的数据量规模. 最新状态量评价结果表用于存储设备最近一次设备状态量的状态评价、人工评价、动态评价报告及其明细; 最新设备评价结果表用于存储设备的最近一次设备状态评价、人工评价、动态评价报告及其明细. 原有设备状态量评价结果、设备评价结果表用于存储历史记录. 每次评价结束后将最新数据保存到相应的历史数据表中. 下面以变电设备状态评价模型为例, 优化前后的对比示意图如下:

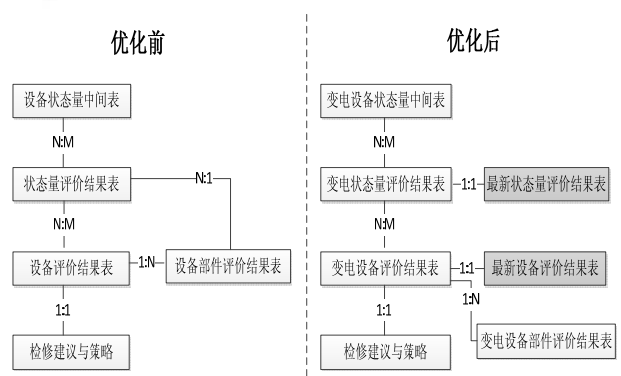


图 3 状态评价分表设计优化前后对比示意图

③ 在查询设备评价过程中, 需要设备状态量评价结果表、设备部件评价结果表、设备评价结果表分

别与千万级别的设备台帐表进行关联获取设备台帐信息. 这种关联操作相当影响系统性能. 通过在这些过程中增加设备名称、设备编码、设备类型等冗余字段. 减少与千万级别的设备台帐表之间关联操作, 达到以小部分空间换取系统快速响应的目的.



图 4 状态评价冗余设计优化前后对比示意图

④ 按照表分类定义, 设备状态量中间存储表属于流水表; 设备评价结果表属于递增表. 依据表分类的存储设计原则, 对设备状态量中间存储表按照创建时间字段按年进行分区设计, 每个分区数据规模控制在 300-500 万之间; 对设备评价结果表按照归档时间字段按年进行分区设计, 每个分区数据规模控制在 100-300 之间; 并将每一个分区分别存放在不同的数据文件中, 映射到不同的物理磁盘上来分散 I/O. 下面以变电设备状态结果表为例, 建立分区表的关键语句如下所示:

```

/*****/
T_ZH_ZTJX_LS_BDZTLPJG_LS
/*****/
create table T_ZH_ZTJX_BDZTLZCCB
(
    .....
    SDSJ          DATE,
    .....
)
partition by range(SDSJ)
(
    .....
to_date('2012-01-01', 'yyyy-mm-dd'))
tablespace ts_zt10,
to_date('2013-01-01', 'yyyy-mm-dd'))
    
```

```

tablespace ts_zt10,
to_date('2014-01-01', 'yyyy-mm-dd'))
tablespace ts_zt11,
partition p_y12 values less than
(maxvalue)) tablespace ts_zt12);
    
```

⑤ 最新状态量结果表结构与状态量评价结果表基本一致, 只是保存数据内容范围有所差异. 状态评价业务频繁, 数据操作主要为新增、修改、删除等变更操作, 而数据查询频率较小. 在物理实现中, 因为最新状态量结果表中的数据需要删除操作频繁. pctused 设置为 50-60%, 新增数据量较小, pctfree 值可以保持不变 10%. 而状态量评价结果表(历史表)则恰恰相反, 其新增数据量较大, 删除数据量较小, 其合理 pctused 值为 30%, pctfree 值 20%-30%. 以变电设备最新状态量结果表为例, 其参数设置如下:

```

/*****/
T_ZH_ZTJX_BDZXZTLPJG
/*****/create
table T_ZH_ZTJX_BDZXZTLPJG
(
    .....
tablespace TS_ZHYW
pctfree 10
pctused 50
initrans 1
maxtrans 255
    .....
)
    
```

⑥ 建立合适的索引. 在评价结果历史数据的查询统计功能中, 要求使用人员首先选择查询、统计的时间段, 通过时间来定位到具体的分区. 所以, 为 GDSJ 字段建立本地索引, 从而提高数据查询、统计的速度. 建立本地索引的关键语句如下:

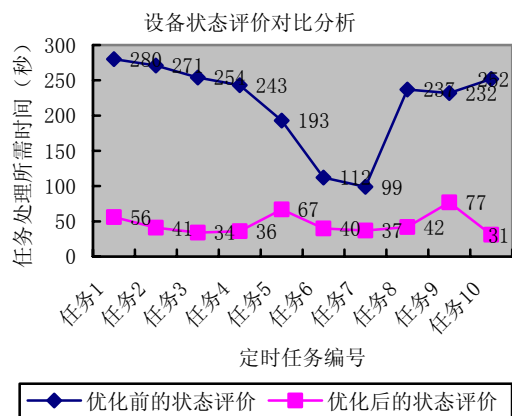
```

/**建立本地分区索引**/
create index IDX_BDZTLPJG_LS_GDSJ
on T_ZH_ZTJX_LS_BDZTLPJG_LS_local (GDSJ)
local;
    
```

### 3.3 模拟实验

在同一网络和硬件环境中分别搭建了优化前、优化后的设备状态检修系统. 模拟构造 20 个并发用户集中录入试验、缺陷等与设备状态评价相关数据, 平均

用户操作时间在3分钟左右,新增录入的数据在1000条左右,后台任务启动10个任务.通过对这10个任务测试结果对比分析如下图所示:



优化后的设备动态评价自动任务的执行时间相比较原有系统,基本保持平稳.虽然每个任务分配存在一定的不确定性,但优化后的单个任务都比原有任务执行时间少.总体执行时间是没有优化前的4.7倍.

#### 4 结语

本文探讨了大数据量表设计优化的关键技术优缺点及设计原则,即逻辑结构设计优化重点介绍表的反范式设计,物理结构重点介绍表分类存储设计、表分区技术、索引技术.在设备状态检修系统的数据库

设计阶段,对设备状态评价功能进行实践应用.通过模拟实验表明对大数据量表设计优化后的系统性能得到了显著提升,对保障系统运行与维护有着重要意义.但本文缺少考虑大量用户并发使用的情况,因此,大数据量表在高并发下的应对策略有待于进一步研究.

#### 参考文献

- 1 苏大威,张乐.基于 Oracle 数据库开发系统的物理设计优化策略.计算机工程,2002,28(2):115-117.
- 2 崔跃生,张勇,曾春,冯建华,邢春晓.数据库物理结构优化技术.软件学报,2013,24(4):761-766.
- 3 乐立鸾,李明.Web 应用系统性能优化.计算机与信息技术,2007,1(11):294-295.
- 4 马惟哲,吕红兵.Web 数据库系统物理模型的优化策略.计算机系统应用与软件,2005,22(1):35-37.
- 5 黄杰圣,李传目.Oracle 大数据量表的处理方法.计算机系统应用,2003,12(1):71-73.
- 6 杨德仁,马晓燕.大型数据库优化设计方案.系统建设,2001,1(5):16-18.
- 7 肖寒,胡广平.数据库大数据量存储结构的设计研究.成组技术与生产现代化,2011,28(2):36-38.
- 8 蒋勇.ORACLE 数据库分区技术及其应用.科技信息,2011,29(1):53-54.