

# 性能优化在信息系统全生命周期中的研究<sup>①</sup>

茆德柱, 钱晓东, 闫训超

(国网电力科学研究院, 南京 211106)

**摘要:** 随着当今社会信息化程度的提高, 企业对信息系统的功能和性能要求越来越高, 系统的性能优化是依赖于信息系统分析与设计、系统开发、系统运行与维护、系统更新四个阶段. 在众多企业信息系统的设计和开发过程中, 仅关注系统功能的实现, 在系统优化方面考虑较少, 将系统优化的重点放在系统运行和维护阶段. 这造成系统被“救火式”优化, 只能治标不能治本或造成系统优化的成本巨大, 甚至系统无法优化, 导致系统不得不提前下线. 本文就如何在信息系统全生命周期的性能优化进行了一些探讨, 提出各阶段性能优化的若干策略, 并结合实际应用进行详细说明.

**关键词:** 性能优化; 全生命周期; 信息系统; SQL 优化

## Research of Performance Optimization in the Life Cycle of Information System

MAO De-Zhu, QIAN Xiao-Dong, YAN Xun-Chao

(State Grid Electric Power Research Institute, Nanjing 211106, China)

**Abstract:** With the advancement of informationization in our society, the Information system of function and performance required more and more. System performance optimization was dependent on information system analysis and design, system development, system operation and maintenance, system updates four stages. During information system design and development of numerous enterprises, the system function was concerned but the system optimization was few concerned. The system optimization only was focused on system operation and maintenance stage. This causes the system to be “fire-fighting” optimization, only suppress the symptoms not the illness or cause the system to optimize the cost was huge, and even the system cannot be optimized, and the system was forced to retired early. This article on how to optimize the performance of the whole life cycle of information system were discussed, some strategies were put forward to optimize the performance of each stage, and combining the practical application in detail.

**Key words:** performance optimization; life cycle; information system; SQL optimization

随着信息技术的飞速发展, 信息系统的功能日益庞杂, 数据量越来越大, 并发访问量也与日俱增, 这对系统的性能提出越来越高的要求. 因此, 对信息系统而言, 面临不但要实现好功能而且更重要是系统的功能高效运行. 但目前众多信息系统为了快速实现系统功能加快系统上线速度, 往往会在系统的规划设计和开发阶段只重视功能实现而忽视性能优化. 这样的系统在系统上线后随着数据量的日益增多, 性能瓶颈就会突显. 此时再加大力量进行系统优化会造成系统

被“救火式”优化, 这种被动式的优化往往治标不治本. 若想彻底解决系统的瓶颈会消耗巨大的成本, 甚至无法优化导致系统性能的问题日益严重, 乃至系统不得不提前下线给企业带来巨大的资源浪费.

系统优化是贯穿信息系统分析与设计、系统开发、系统运行与维护、系统更新四个阶段.<sup>[1]</sup>在每个阶段都有不同策略和侧重点, 这样既能提高系统性能也能节省后期的运维成本, 提高系统的使用年限, 同时也节省了总的投资, 给企业带来巨大的效益. 本文从信息

<sup>①</sup> 收稿时间:2013-09-01;收到修改稿时间:2013-09-29

系统全生命周期的四个不同阶段,对系统优化进行阐述,提出若干性能优化策略,并结合实际介绍优化技术在设备(资产)运维精益管理系统(以下简称 PMS2.0)中的应用。

## 1 系统分析与设计阶段

系统分析与设计阶段是系统的起始阶段,根据企业的整体目标和发展战略,确定信息系统的发展战略,明确企业总的信息需求,指定建设总计划,其中确定系统的总体目标、功能、大致规模和所需资源,并根据需求的轻重缓急以及资源和应用环境的约束,将信息系统的建设内容进行细化和分解。

在此阶段,性能优化尤其重要的。应从应用需求入手,详细了解业务特性,分析出系统是联机事务处理(OLTP)、联机分析处理(OLAP)、混合系统(OLTP 和 OLAP 兼有)<sup>[2,3,4]</sup>。对不同的系统有不同的优化策略和方法。目前企业的信息系统多以混合系统为主。对于混合系统而言,其中的 OLTP 功能主要由企业基层操作人员和低层管理人员使用的日常操作处理,主要数据是当前最新的,数据量较小。例如 PMS2.0 的变电站值班日志,线路工区的巡线记录等。OLAP 功能主要由企业的决策人员和高级管理人员进行分析决策。主要数据是历史的,聚集的,多维的,集成的,数据量也是百万级别以上。例如 PMS2.0 的全省变电站全生命周期内的故障缺陷原因统计分析等。

由于混合系统中的 OLTP 和 OLAP 的业务特点不同,导致采用的优化策略不同所以要将属于 OLTP 的功能和 OLAP 的功能分开设计。

由于 OLTP 和 OLAP 的读写数据的方式不同,需要将两者进行分表或分库设计。将两者的数据读写分开,避免高并发、高吞吐量的 OLTP 操作被高消耗 CPU 和 IO 资源的 OLAP 操作影响。在 PMS2.0 中为了减少日常操作功能和辅助决策功能的互相影响,采用的分库设计。将日常操作的数据存放在读写性能和数据安全较高 RAID10 存储上构建的生产业务库中,将辅助决策的数据存放在较为经济且读性能较高的 RAID5 存储上构建的管理数据库中。

在系统分析与设计阶段,即确定系统的应用架构,数据架构和软硬件架构的过程。信息系统的设计阶段,架构极其重要,良好的架构有助于系统性能的提升,不好的架构将成为系统的瓶颈,导致系统性能

无法提升<sup>[5,6]</sup>。

在应用架构设计要注意,模块功能内部实现高内聚,即模块把相关的功能内聚到一起。模块之间的关系要低耦合,即不同模块之间互连程度的块间联系要低。高内聚,低耦合的好处体现在系统持续发展的过程中,高内聚,低耦合的系统具有更好的重用性,维护性,扩展性,减少模块之间的调用,减少模块对数据的调用,提高 IO 的效率,提高系统性能。

在数据架构设计一般采用 3NF 范式数据模型,为了性能提升,一些大数据量表适当的引用了数据冗余,根据业务再结合采用了当前表+历史表的数据模型。一般原则是需要

- ① 主从库分离,读写分离;
- ② 大量冗余数据要拆;
- ③ 需要频繁访问的数据,有冗余也不要拆;
- ④ 没有频繁访问的数据,没有冗余也要拆;

软硬件架构设计要注意,应用能支持群集,数据在持久层或持久层之上做缓存,基础公共模块平台化,业务模块基于基础模块平台进行二次开发等技术,减少对数据库的访问;数据库能支持 RAC,提高数据库的被访问能力以及可靠性;<sup>[8,9]</sup>硬件可扩展性和易维护性;要有统一的界面进行信息扎口监控管理,便于及时发现故障和性能问题,分析问题找出解决方案,提高系统的可用率和稳定性。例如 PMS2.0 系统采用基于 SG-UAP1.5 统一开发平台进行二次开发,既提高了开发效率又提高了系统的性能。

## 2 系统开发阶段

系统开发过程是系统设计的具体实现过程,在此阶段,优化主要分为两个部分,对应用程序代码的优化,对数据库表结构以及 SQL 代码优化。<sup>[10]</sup>

应用程序代码的优化,主要体现在:

- ① 页面静态化,通过静态页面的生成和访问能有效地降低系统负载;
- ② 页面分区将动态和静态的区域分开,动态采用 AJAX 技术将 JSON 数据流传送到客户端,再聚合出最终的页面将服务端压力分担到客户端;
- ③ 数据缓存,数据缓存通常包括持久层层面的缓存和外部接口调用的缓存,数据缓存可以减少各类 I/O 调用,提高用户响应;
- ④ 组件服务化,这样便于将组件的处理并行化,增

加系统的响应速度. 遵循 SOA 的方式, 系统中使用高性能的 ESB 企业服务总线来进行服务调度和任务分派;

数据库表结构的优化, 主要体现在:

① 数据库表的大字段剥离, 保证单条记录的数据量最小, 非结构化数据单独存放在存储上, 不要放在数据库中, 减少对数据库 IO 的影响;

② 依据业务恰当地使用索引, 必要时建立多级索引, 组合索引和倒序索引;

③ 依据业务逻辑将大表、日志表等进行分区和拆分;

④ 将经常一起访问的数据进行字段冗余, 减少跨库查询和大表连接查询;

⑤ 对实时性要求不高的查询统计分析工作, 通过 JOB 或定时任务在业务低谷时事先查询出结果保存, 减少业务高峰时的实时查询和统计分析, 减轻业务高峰时刻的系统压力;

⑥ 进行表的读写分离, 避免 IO 的争夺;

⑦ 将表数据和索引数据分开存储存放在不同的物理磁盘会商, 分别使用独立的表空间, 避免表数据和索引的 IO 操作产生影响系统性能的 IO 竞争, 提高系统的响应效率;

⑧ 将实时数据和历史归档数据分开存储, 分库存储, 提高实时数据的访问效率, 这样归档库可以单独部署, 充分利用服务器的资源, 提高归档库的相关高级业务的分析计算速度;

⑨ 将日常业务数据和查询统计辅助决策类的数据库分开存储, 分库存储, 提高日常业务数据访问效率, 也提高决策类数据的分析效率;

SQL 代码优化, 要注意:

合理的选用 Oracle 优化器, 基于规则(RULE), 基于成本的(COST)和基于选择性的(CHOOSE), 一般而言对表进行统计信息的对象进行 CBO 优化器, 没有统计信息的对象进行 RBO, 为了避免不必要的全表扫描要避免选择 CHOOSE;<sup>[11]</sup>

① 共享 SQL 语句, 在 SQL 书写过程中引入参数变量为了减少 Version count 提高 SQL 的共享, 避免重复 SQL 的硬解析, 提高 SQL 的执行效率;

② 对 SQL 中进行需要排序和连接操作的字段建立索引;

③ 在不会产生重复记录的应用中用 UNION ALL 代替 UNION, 这样可以避免排序和删除重复记录这一

步骤, 提高结果返回的速度.

④ 通过 WHERE 子句限制记录数目, 避免使用 HAVING 子句所造成的查询出所有记录之后再对结果进行过滤, 这可以避免开销较大的排序和合计等操作;

⑤ 合理使用索引, 避免对大表进行全表扫描, 同时注意 WHERE 语句的连接顺序, 将过滤掉最大记录数的条件必须写在 WHERE 语句的末尾;

⑥ 尽量避免在 SQL 语句用 OR, 或数学计算运算符, 这些都会导致索引失效而全表扫描, 降低查询效率;

⑦ 子查询展平, 把子查询转换成连接来实现, 因为主查询的每一条记录, 子查询都要执行一次, 嵌套的层次越多效率越低;

### 3 系统运行维护阶段

系统开发项目完成并进行测试后即投入应用, 进入上线运行和维护阶段. 这是信息生命周期中历时最长的阶段, 也是管理信息系统实现其功能, 获得效益的阶段. 科学的组织与管理是系统正常运行, 充分发挥其效益的必要条件, 而及时、完善的系统维护是系统高效运行的基本保证, 是系统性能提升的关键.

在上线阶段要针对现场环境的软硬件环境参数(操作系统内核参数, 数据库配置参数, 中间件启动参数, 负载均衡设备的流量分发算法, 健康检查参数)进行优化, 一般依据软件测试结果进行初步优化, 待运行一段时间(一般 3 个月左右)结合上线运行效果再进一步的优化, 以满足达到系统最优.

系统维护可以分为纠正性维护、适应性维护、完善性维护和预防性维护四个类型的维护. 为了提高系统性能, 重点在日常维护, 首先要建立运维管理制度和流程, 专业的运维队伍, 日常巡查, 和应急措施和应急预案, 并对预案做定期的演练工作, 变被动为主动确保在突发事件来临时也能安全运行. 对不同的维护制定不同工作方式具体如下:

纠正性维护是指对系统进行定期的和随机的检修, 纠正运行阶段暴露的错误, 排除故障, 消除隐患, 更新易损部件, 检查数据存储情况, 保障系统按预定要求完成各项工作.

适应性维护是指随着系统运行数据量的变化, 系统中部分的工作内容与方式已不能适应变化了的环境, 因而影响系统预定功能的实现. 要依据现场的运行情况(内核参数, 数据库参数, 检查 SQL 的执行情况确保

执行路径的正确,及时发现系统的瓶颈和风险点进行适当的调整、修改,以满足系统安全运行的需要,确保系统的性能。

完善性维护是指用户对系统提出了某些新的信息需求,因而在原有系统的基础上进行适当的修改、扩充,完善系统的功能,以满足用户新的信息需求。在此过程要注意新增功能和原功能的有机结合一体化设计,避免新增功能影响原功能的性能。

预防性维护是对预防系统可能发生的变化或受到的冲击而采取的维护措施。在信息系统中特别要注意到业务特点,要有充分应对业务高峰时刻的应对措施,比如在 PMS2.0 中,要应对电网的春秋集中两检过程中的输变电巡检,任务单下发和修缺模块的性能要求;要应对电网夏天的迎峰度夏时刻的高负载,运行调度模块的性能要求;要应对集中整改设备参数情况下的流程管理和设备管理模块性能要求,对不同的业务高峰,要有充分识别和理解并针对业务特点提前做出相应的优化策略,做好防患于未然。

#### 4 系统更新阶段

企业面临的内、外环境不断变化,组织的目标、战略和信息需求也必须与环境的变化相适应。可是信息系统的维护工作只限于通过小范围内局部调整来适应变化不很显著的情况。当现有系统或系统的某些主要部分已经不能通过维护来适应环境和用户信息需求的变化时,或者用维护的办法在原有系统上进行调整已不经济时,则整个管理信息系统或某个子系统就要淘汰,新的系统建设工作或项目开发工作便随之开始。

在此阶段性能优化工作要注意总结客户反馈的性能表象和业务运行特性,找出目前系统不适应的功能和瓶颈,对性能和功能需求进行考量和评估是进行部分子系统替换升级还是淘汰整个信息系统进行新建系统来满足企业的管理工作的需求。

#### 5 结语

性能优化是一个系统工程,是贯穿整个信息系统

全生命周期。要变被动为主动,避免被动的救火式优化,彻底的解决性能瓶颈,提高系统运行效率。在企业大型信息系统中性能是否优化已成为衡量系统成败的关键,因此在信息系统生命周期里(从系统架构的选型,参数优化,到数据库的优化,SQL的优化,日常运行与维护)时刻都要有性能优化的思想。随着信息化云技术的进一步的发展,信息系统的数量越来越大,功能越来越复杂,系统越来越集中,信息系统性能优化将显得更重要。

#### 参考文献

- 1 任玉辉,玉成良.企业级应用系统中的数据库性能优化策略.计算机系统应用,2008,17(3):7-9.
- 2 郭珉.ORACLE 数据库 SQL 优化原则.计算机系统应用,2010,19(4):170-173.
- 3 施伯乐,丁宝康,汪卫.数据库系统教程.北京:高等教育出版社,2008.
- 4 潘文林.ER 模型转换为关系模式的实用规则.计算机工程与应用,2006,(1):169-173.
- 5 Belknap P, Dageville B, Dias K, et al. Self-tuning for SQL performance in Oracle database 11g. IEEE International Conference on Data Engineerint. 2009. 1694-1700.
- 6 Hussain T, Shamail S, Awais MM. Eliminating process of normalization in relational database design. Proc. of the 7th International Multi Topic Conference. INMIC 2003. 2003. 408-413.
- 7 谷小秋,李得昌.索引调整优化 Oracle9i 工作性能的研究.计算机工程与应用,2005,(26):174-176.
- 8 戴小平.Oracle 9i 数据库性能调整与优化.安徽工业大学学报,2006,(3):315-319.
- 9 杜庆峰,张卫山.Oracle 的中大型应用系统性能优化分析.计算机工程,2005,(14):91-93.
- 10 Vaidyanatha GK, Deshpande K, Kostelac J.钟鸣,石永平译.Oracle 性能优化技术内幕.北京:机械工业出版社,2002.
- 11 闫训超,茆德柱.国网公司生产管理系统系统优化探析.中国电力教育,2012,12:111-113.