

Oracle数据库应用系统的优化策略

周洪宇

南京农业大学 江苏 南京 210000

摘要：随着社会科技的发展，信息系统也在不断完善。信息系统依靠数据库才能够平稳运行，数据库影响整个系统的功能。数据库在运转中可能会出现一些功能性问题。数据库的硬件设备、数据库服务设备、数据库运行设备、操作设备等一些设备非常容易在运行时间久了以后出现问题。Oracle数据库系统在运转时能够帮助用户快速找到相关数据。通过索引的建立，数据库的运行效率能够达到有效的提高。通过探究Oracle数据库应用优化可以帮助用户使用数据库寻找相关数据，为用户提供有益帮助。

关键词：Oracle系统；数据库；优化策略

Oracle数据库系统能够有效地整合数据，并且是目前世界上一种相对来说比较完善的数据库管理系统。Oracle数据库系统具有较强的性能，能够与其他系统兼容，应用在各种微机系统。Oracle系统可以高效率地运转，达到对数据库系统整体的管理与把控。

1 Oracle 性能优化的主要方面

1.1 SQL与索引优化。SQL语句优化。避免全表扫描（优先通过WHERE条件使用索引访问）。使用绑定变量减少硬解析，消除隐式类型转换。改写低效语法（如用EXISTS替代IN，UNION ALL替代UNION）。索引策略，在WHERE/ORDER BY/GROUP BY涉及的列创建索引。对复合查询构建组合索引（如CREATE INDEX idx_col1_col2 ON tb(col1,col2)）。定期重建碎片率超过15%的索引，删除低效冗余索引。

1.2 内存资源配置。SGA组件调优。动态调整共享池（shared_pool_size）、缓冲区缓存（db_cache_size）占比，建议占物理内存的60%-70%。启用自动内存管理（AMM）实现动态分配。PGA优化，设置pga_aggregate_target为物理内存的20%-25%。监控排序/哈希操作，避免临时表空间溢出。

1.3 存储与I/O优化。存储架构设计，对超过10GB表实施范围/列表分区。数据文件与日志文件分离存储，索引单独表空间。I/O负载均衡，使用ASM存储管理自动分散数据分布。调整DB_WRITER_PROCESSES为CPU核数的1/4。

1.4 参数与配置调优。合理设置log_buffer等参数。启用并行查询优化（parallel_max_servers）、并发控制、合理设置UNDO_RETENTION防止快照过旧、对批量操作启用并行DML提升处理速度。

1.5 统计信息管理。统计信息维护，每日自动收

集统计信息（DBMS_STATS.GATHER_DATABASE_STATS）。对数据倾斜字段创建直方图优化基数估算。执行计划控制，使用SQL_PROFILE固定高效执行计划。分析AWR报告识别TOP SQL。

1.6 架构设计优化。应用层优化。读写分离架构（OLTP与OLAP分离）。对频繁JOIN表实施反范式化设计。高可用设计。多租户架构（CDB/PDB）实现资源隔离。建立自动化备份恢复体系。优化验证路径：通过AWR报告定位性能瓶颈→SQL改写优化→索引调整→内存参数微调→分区改造→持续监控优化效果。该优化体系实施时优先解决响应时间TOP 5的SQL语句及关键资源争用点（如latch:cache buffers chains等待事件）。

2 Oracle SQL 优化技巧

2.1 索引优化策略。精准索引设计，对WHERE、JOIN、ORDER BY/GROUP BY涉及的列创建索引，优先选择高基数（唯一性高）的列（如主键）。组合索引遵循等值条件列在前、范围查询列在后的原则（如CREATE INDEX idx_dept_date ON emp(dept_id,hire_date)）。定期监控索引碎片率，碎片超过15%时需重建（ALTER INDEX...REBUILD）。避免冗余索引。删除低效索引（如性别等低基数列的索引）。对大表启用位图索引（CREATE BITMAP INDEX）需谨慎，防止DML锁争用。

2.2 SQL语句优化。语法改写规则，用EXISTS替代IN子查询，用UNION ALL代替UNION（避免去重排序）。避免SELECT *，仅查询必需字段以减少I/O开销。将函数操作移到右侧（如WHERE create_time>TRUNC(SYSDATE)改为WHERE create_time>:var）。绑定变量优化，使用绑定变量(:var)而非字面值，减少硬解析。动态SQL中优先采用DBMS_SQL或EXECUTE IMMEDIATE绑定参数。

2.3 执行计划控制。执行计划分析,通过EXPLAIN PLAN或DBMS_XPLAN.DISPLAY_CURSOR查看执行计划。使用AWR报告定位TOP SQL (SELECT * FROM TABLE(DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_HTML(...))。计划稳定性控制,对关键SQL使用SQL_PROFILE固定执行计划。使用DBMS_STATS.LOCK_TABLE_STATS锁定稳定表的统计信息。

2.4 统计信息管理。自动收集机制,每日自动收集统计信息(DBMS_STATS.GATHER_TABLE_STATS)。对数据倾斜字段创建直方图(METHOD_OPT = '>FOR COLUMNS SIZE 250')。动态采样配置,对临时表启用动态采样(DYNAMIC_SAMPLING = 2)。

2.5 高级调优技巧。并行处理优化,对大表查询启用并行(如SELECT * FROM emp PARALLEL(emp,4))。设置parallel_max_servers限制并行度。物化视图加速,对复杂聚合查询创建物化视图(CREATE MATERIALIZED VIEW...),定时刷新。

3 Oracle 数据库内部优化策略

3.1 优化器配置优化。CBO参数调整,设置optimizer_mode=ALL_ROWS优先保证批处理效率,联机交易系统可尝试FIRST_ROWS(N)模式。调整optimizer_index_cost_adj参数(建议值10-50降低索引访问成本)。启用optimizer_dynamic_sampling = 3提升复杂查询执行计划准确性。并行处理控制,根据CPU核数设置parallel_max_servers(推荐值为CPU核数×2)。对大表查询添加PARALLEL(table_name,degree)提示实现可控并行。

3.2 内存管理策略。动态内存分配,SGA分配采用SGA_TARGET自动管理,建议占物理内存的60%-70%。PGA设置pga_aggregate_target为物理内存的20%-25%。缓冲池分化策略,对热点表启用KEEP POOL保留缓存(如ALTER TABLE tab STORAGE(BUFFER_POOL KEEP))。对大表设置RECYCLE POOL防止全表扫描冲击主缓冲池。

3.3 执行计划控制。计划稳定性保障,使用SQL_PROFILE绑定高效执行计划(需DBMS_SQLTUNE包)。对关键SQL执行ALTER SYSTEM FLUSH SHARED_POOL清空低效计划缓存。资源消耗监控,定期分析AWR报告定位TOP SQL及等待事件。优化实施路径:分析AWR报告→锁定TOP SQL→执行计划优化→内存参数微调→监控缓冲池命中率→持续迭代优化。通过上述策略组合实施,典型OLTP系统很多情况下可提升吞吐量,缩短批处理任务执行时间。优化过程中需关注v\$sysstat中的Parse Time CPU、Buffer Cache Hit Ratio(缓冲命中率)等关键

指标变化。

4 评估 Oracle 优化效果的方法

4.1 性能基准测试验证,A/B测试对比。优化前后执行同一批关键SQL,对比BUFFER_GETS(逻辑读)和ELAPSED_TIME(执行时间)下降幅度。通过ALTER SESSION SET SQL_TRACE = TRUE生成跟踪文件,使用TKPROF分析优化前后物理读/逻辑读比例变化。压力测试验证。使用HammerDB等工具模拟高并发场景,监测TPS(每秒事务数)提升率和DB file sequential read等待事件减少幅度。

4.2 执行计划优化验证。进行执行计划对比,使用EXPLAIN PLAN FOR生成优化前后执行计划,重点检查:全表扫描转为索引扫描(TABLE ACCESS FULL→INDEX RANGE SCAN)、低效Join方式改进(如HASH JOIN替代NESTED LOOPS)、绑定执行计划。对关键SQL采用SQL_PROFILE固定优化后的执行计划,通过DBA_HIST_SQLSTAT监控执行稳定性。

4.3 资源消耗监控。监控资源消耗可使用如下指标:内存效率指标Buffer Cache Hit Ratio,PGA Aggregate Target统计Work Area Executions-optimal比例,CPU与I/O负载,v\$sysstat中Parse Time CPU降低幅度,v\$filestat统计优化后平均单次物理读耗时下降比例。

5 Oracle 数据库系统性能的优化策略

5.1 内存区域调整与优化。Oracle数据库的SGA和PGA为重要的内存结构,SGA包含缓冲区缓存、共享池、重做日志缓冲区三个部分,这三个部分对Oracle的操作有很大的影响。(1)对缓存链接进行了优化。数据缓存就是将正在执行的数据存储到数据库中,当使用者请求数据的时候,可以将缓冲链路的数据直接传递到使用者,从而为使用者节约查询的时间。当使用者所提出的资料并不存在于缓冲区中时,必须先从资料库中取出资料,再将资料搜集、传送至缓冲区,再经由缓冲区传送到使用者。这种方式会耗费更多的时间,也会给使用者带来不便。其中,缓存部分可以发现新的数据,并将其存储起来,从而大大节约了使用者寻找的时间。将采集到的数据存储于缓冲区中,可以将这些数据直接传输到使用者的手中,从而缩短了使用者搜寻的时间,让使用者可以更好地完成对数据缓冲区的最优和调节,从而使数据的运算性能得到更好的提升。(2)最优的分享链接。对数据库共享链装置IQS进行存储,确保可以重复利用,不需要反复的解析,可以在利用的时候减少利用率。数据库分享可以让使用者在使用数据库系统的时候,协助使用者对句子进行解析,可以更好地减少能量

的消耗,实现更快更好的运作。在Oracle数据库工作期间,可以提升使用者的利用效率,减少使用者查找数据的时间。通过对共享链的优化和调节,可以减轻网络中的恶意使用概率,为更多的应用提供支持,从而达到最优的共享连接。(3)日志缓冲系统的优化。利用日志缓存技术对数据库中的数据进行有效的存储,并且可以对被更改的数据进行实时存储。日志缓冲链,若已达到某一存储容量,则将未存储的日志存放于等候链,若不能将其高效地存储,则将出现故障。在记录中存储的数据出错次数越多,说明缓存链的存储容量就越少,这将会对数据库的整体运行产生负面的作用。对记录缓存链接的调节和优化,就是要对某些资料做适当的修改和调节,使整个数据库得到持续的改进。

5.2 网络I/O优化策略。(1)索引INDEX优化。索引是数据库中最主要的结构,它可以对数据库进行最优处理。在ROWID中引入了一个新的指标,利用该指标可以迅速地查找所需的信息,从而提高了查询速度。而在数据库中使用索引,则是为了提高使用者的查询效率,协助使用者寻找所需的材料,以提升使用者使用网路的效能。比较成熟的系统通常都带有一个指标,可以让使用者缩短工作的周期,提升工作的效率。不过,索引并非愈多愈好,过多的索引有时候反而会影响资料库的效能。(2)Oracle数据库系统的分区技术。Oracle系统拥有比较完备的数据库,可以进行各种管理。它可以把各种资料分别存放在一个硬盘里,这样就大大提高了查询的速度。它可以有效地协助使用者在进行数据的查询过程中缩短所需的时间,并且可以针对各种存储盘使用FTS来进行资料搜索,这可以让使用者缩短寻找资料的时间,并提升使用者找到资料的速度。

5.3 CPU性能优化。通过对CPU进行最优配置,可以确保各种资源高效运行,实现高效运行和运行品质。通过对CPU处理器的数据进行加工与检索,为Oracle的应用提供了广阔的发展空间。当前,随着各行各业对CPU的重视和对系统的应用的重视,可以在数据库服务装置中将CPU分开,并且可以在不同的业务装置之间传递CPU的请求,并且可以通过PQO方式执行数据的查询,实现超过两个CPU对SQL语言的要求进行处理,并对位于不同硬盘中的数据进行处理,并在执行过程中对数据进行分析与处理。

5.4 SQL语言优化。SQL语言优化系统在进行优化时,首先要完成对市场中的数据建立某种索引,从而完成对数据的检索。当执行数值符号的查询时,不要使用通配符的方法来进行查询,当数据库运行时,尽可能避免整个搜索。此外,当有目的地进行资料检索时,可以把一个统一的体系放在一个统一的模组内,而当资料检索时,则要构造出比较完备的连结状况。

总之,Oracle数据库系统相对来说比较复杂,优化难度系数相对较大。在对Oracle数据库系统进行优化时,要根据储存磁盘性能、语言优化等多个方面进行调整,帮助Oracle数据库系统达到相对完善的运行效果,不断地提高用户的使用便利性水平。Oracle数据库系统的进一步优化与调整具有极为重要的意义,能够帮助Oracle数据库系统实现性能优化,提升用户使用效率。

参考文献

- [1]李平.Oracle10g数据库系统性能优化与调整.2023.
- [2]王福哲.基于Oracle数据库系统的优化与性能调整研究.2023.