8. 数据库系统性能调优

8.1 概述

- 性能是从数据库诞生起就存在的问题，迄今仍然是数据库系统的关键问题之一。
- 在一定的资源条件下，尽可能满足用户对服务及时性的要求。
- 数据库性能是贯穿数据库整个生存周期的全局课题。
- 响应时间和吞吐率是衡量性能的两个既有关联、又有区别的指标。
- 必须对事务执行的全过程和影响其性能的各种因素有全面的了解和分析，并择其中影响大的执行环节，采取调优措施。
- 数据库系统是一个服务系统，其性能不仅取决于数据库系统本身，而且与它所服务的应用系统以及应用系统与数据库服务器的连接有关。

2

8.2 影响数据库系统性能的因素

- 数据库系统的基础设施
  - 处理机及其内存系统、外存储器系统、操作系统以及与其连接的网络系统。
- DBMS及其配置
  - 应用系统与数据库服务器的连接
  - 存储器系统—RAID
  - SQL的编译执行和解释执行
    - Oracle SQL sharing

3

8.2 影响数据库系统性能的因素

- DBMS及其配置
  - I/O瓶颈
    - 原则上，凡是需要重复访问的数据和程序都可以在内存中保留一段时间—库缓冲区。
    - 数据目录缓冲区
    - I/O缓冲区（I/O buffer cache）
    - 后像缓冲区（redo log buffer cache）和后像文件（redo log file）。Oracle10.2建议初始设置不超过0.5MB。
- 应用系统的规划、设计与开发
  - 数据库性能调优不仅是DBA的事，应用程序设计者也要承担其责任。

4

8.3 系统结构与性能调优

- 以SAN为中心的计算机系统
  - 含有两类设备，即处理机（包括CPU及内存）和存储设备。它们彼此间没有从属关系，可以通过SAN按客户/服务器关系形成一或多个计算机系统。
- 共享缓冲区
  - Oracle“Real Application Clusters”，简称RAC
- GRID
  - *Egenera*虚拟计算机系统介绍
  - Oracle 10g网格体系结构

5

8.4 应用程序设计与性能调优

- 数据库设计
  - 数据模式、索引
  - 设置索引惟一的目的是为了提高查询效率，可是任何索引都有不利于数据库性能的“副作用”。
  - 可以用EXPLAIN PLAN语句查看其执行计划。要结合对应用需求的了解和查询优化器实现的掌握，合理设置索引。

6

8.4 应用程序设计与性能调优

- 网络连接
  - 应合理配置足够的网络资源、简化传输协议消除连接中可能出现的瓶颈
  - 还要在设计应用程序时注意节约网络资源，特别减少连接和对话次数
    - 尽可能不用数据库的人机交互接口
    - 避免用游标逐行传送查询结果。例如Delphi中table控件和query控件之差别

7

8.4 应用程序设计与性能调优

- SQL程序设计
  - 很多DBMS目前已提供了共享SQL程序的手段，关键在于统一认识和严格实施。
  - 对应用程序设计者的主要要求就是参数要用命名一致的赋值变量表示。
- 合理使用事务——事务隔离级别
- 应用程序本身数据结构、算法的优化，窗口显示处理的优化。如处理完一屏即显示而不是处理完所有内容再显示
- ……

8

8.5 SQL调优

8.5.1 SQL调优与查询优化器

- 模透查询优化器的“脾气”，“投其所好”，“避其所忌”
- 用提示语句影响查询优化器拟订执行计划

9

模透查询优化器的“脾气”

- 如有些查询优化器对于连接操作，一般按连接对象在FROM子句中出现的先后次序进行连接
  - 则应小表放在前面，大表放在后面
- 又如在有些查询优化器中，凡是查询条件用OR连接的，就一概不用索引
  - 设法改写，使用UNION ALL

```
SELECT *
FROM STUDENT
WHERE YEAR(BDATE)=1980  OR
        HEIGHT>=1.80;
```

10

模透查询优化器的“脾气”

- 在上面的SQL中，即使在属性HEIGHT和函数YEAR（BDATE）上建有索引，查询优化器也不会用这些索引。改写：

```
SELECT *
FROM STUDENT
WHERE YEAR(BDATE)=1980
UNION ALL
SELECT *
FROM STUDENT
WHERE HEIGHT>=1.80;
```

- 模透查询优化器的“脾气”并不难，可以用EXPLAIN PLAN语句获得查询优化器所拟订的执行计划，还可以用跟踪语句进一步查得计划的详细执行情况。

11

用提示语句影响查询优化器

- 提示是用户影响查询优化器决策的一种手段。
  - 优化目标——例如提示以响应时间或吞吐率为优化目标。
  - 查询语句变换——例如提示将嵌套查询变换成非嵌套查询等。
  - 存取途径选择——例如用某一索引或全表扫描等。
  - 连接次序方法——例如提示连接的先后次序，采用嵌套循环或排序归并法等。
  - 其他提示
- 语法规定：凡是查询优化器不能理解的内容，查询优化器就不予理采，但不影响对其他可理解部分的执行。

12

提示举例一

```
SELECT /*+FIRST_ROW（3） 优化目标是尽快
          返回前3行 FULL（S） 对S全表扫描*/
SNAME，HEIGHT
FROM  STUDENT S
WHERE SEX=‘男’；
```

13

提示举例二

```
SELECT /*+INDEX（EMP EMP_EDNO_IX）*/
ENO，ENAME
FROM  EMP
WHERE EDNO=100；
```

14

提示举例三

```
SELECT /*+ USE_NL_WITH_INDEX（E
          EMP_ENO_IX）*/
D.DND，D.MENO，E.ENAME
FROM DEPT D，EMP E
WHERE D.MENO=E.ENO;
```

15

8.5.2 SQL调优导则

- SQL是非过程语言，同样的功能可有多种不同的表达，这就存在一个如何表达才能提高SQL语句的执行效率问题。
- SQL语句能否有效执行，不但与语句本身有关，还和DBMS及执行环境有关。下面的一些规则在一般情况下是可用的，但不能绝对化。

16

8.5.2 SQL调优导则

1. 尽可能避免排序操作。因此对下列语法成分或操作，必须慎用：
   - DISTINCT
   - 集合操作UNION，INTERSECTION， EXCEPT（MINUS）
   - GROUP BY，ORDER BY子句
   - 排序归并连接法（除非参与连接的表已经按连接属性排序）
   - 建立簇集索引等

17

8.5.2 SQL调优导则

2. 利用查询条件，尽可能早地消除无用元组，以缩小中间结果：
   - 能用WHERE子句表示的查询条件，不要放在HAVING子句中。
   - 查询计算机系所开课程的平均成绩和最高成绩：

```
SELECT C.CNO，AVG（SC.GRADE），MAX
    （SC.GRADE）
FROM SC，COURSE C
WHERE C.CNO=SC.CNO
GROUP BY C.CNO
HAVING C.CNO=‘CS%’;
```

18

8.5.2 SQL调优导则

- 改写后的查询：

SELECT C.CNO，AVG（SCGRADE），MAX（SC.GRADE）
FROM SC，COURSE C
WHERE C.CNO='CS%' AND C.CNO=SC.CNO
GROUP BY C.CNO

- 当心在语句中出现笛卡尔乘积（即无连接条件的连接），慎用外连接、外并操作、非等连接（即连接条件中出现非等号比较符）等操作。
- 在多元连接时，须将选择性高、元组少的表列在FROM子句的前两位，其他表也要按选择性高和元组数少优先原则依次排序，以缩小中间结果。

19

8.5.2 SQL调优导则

3. 大量加载数据时，不要用INSERT语句
   - 应利用加载工具，例如Oracle中的SQL*Loader
4. EXISTS，IN的应用
   - 慎用关联嵌套子查询，在多数情况，IN比EXISTS较为有利。
5. 不滥用视图
   - 视图是按其定义临时生成的中间结果，没有索引之类的存取路径可用。
   - 虽然允许视图的定义中再引用其他视图，但就性能而言，这种多层嵌套的视图不值得采用。如果不是出于安全、保密等的考虑，可将视图定义合并到FROM子句中。

20

8.5.2 SQL调优导则

6. 当心查询优化器选择全表扫描
   - 如果查询条件中出现比较符“<>”或“IS NULL”等，则查询优化器一般选择全表扫描。

- 总之，在编写SQL语句时，设计者都得想一想：查询优化器可能为所设计的语句选用哪样的执行计划，以免出现意外的耗费资源的“大户”，这样做是有益的。

21

8.6 数据库性能调优的实现

- 无论是主动性能调优，还是被动性能调优，都必须拥有充足的运行记录，才能做出准确的诊断。
- 数据库性能按其“病因”的来源可以由粗到细地分为3个粒度：基础设施、数据库实例和会话（session）。

22

基础设施级运行记录

- 操作系统运行记录——了解硬件资源分配和利用情况，其中主要有CPU的总利用率和系统利用率。
- 网络运行记录——主要有网络平均延迟时间、等待网络服务的队列长度等。

23

数据库实例的运行记录

- 应用程序运行记录——应用系统的CPU利用率、库缓冲区匹配命中率、从I/O缓冲区直接读取数据库数据的命中率、连接数据库实例的响应时间和持续时间、对话次数、事务的最长持续时间及平均时间等。
- 等待时间（wait events）——等待事件反映某种资源的紧张或短缺。等待队列过长是形成瓶颈的先兆。等待事件是调优的重要统计数据。

24

会话级运行记录

- 一个会话的全部活动的记录称为对话的历程（session history）。在Oracle 10g中，成功地用采样（sampling）的方法解决了此问题。每次采样可为每个会话录下一个元组。
- 采样数据存于内存中一个环形缓冲区中。称ASH (Active Session History)缓冲。如ASH缓冲满或采样达1小时，则将ASH缓冲中的采样数据转存到磁盘上的Automatic Workload Repository （AWR）中。
- Oracle默认值：ASH一般每秒采样一次，AWR一般将每小时的采样数据加工成一系列的快照和报告，以反映对话活动和数据库负荷的变化。一系列快照构成近期活会话的历程，快照一般保留一周。

25

Oracle自诊断工具ADDM

- 在AWR的基础上，Oracle推出了自诊断工具ADDM（Automatic Database Diagnostic Monitor）：

1. 告警功能
   - 例如表空间（tablespace）存满97%、AWR的快照陈旧、可恢复运行的对话仍被挂起等。
2. 自诊断和监控
   - ADDM定期地分析AWR中的统计值，诊断系统中存在的性能问题，并与其他顾问（advisor）类的工具配合，推荐一或多个解决方案，有时还给出各种方案的量化的预期效益，供DBA比较选用。

26

Oracle自诊断工具ADDM

3. 调优目标定在吞吐率，而不在响应时间
   着重发现和解决影响全局的问题。主要有：
   - 及时发现和消除系统性能的瓶颈；
   - 及时发现资源短缺，并采取应对措施；
   - 及时发现大负荷SQL语句；
   - 监控应用系统不合理地使用Oracle，例如反复登录/退出数据库服务器，与数据库服务器连接/断开频繁，SQL共享率不高，库缓冲区匹配命中率太低等等。
4. 可识别出系统中“无问题领域”（non-problem areas）
5. 诊断-建议-调优往往是个迭代过程

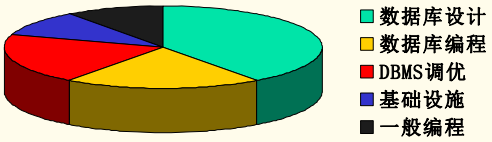
27

总结

- 总之，数据库性能是贯穿数据库整个生存周期的全局性课题。
- 根据具体的软硬件环境和应用需求调整所用DBMS的各种参数及配置固然重要，但只是保证性能的一个方面。
- 必须从整个信息系统的规划设计、开发实现、部署实施及使用维护全过程考虑，系统设计人员、开发人员、DBA等要协调配合，才能保证系统的性能满足应用需求。

28

总结



29

谢谢大家！

Oracle SQL sharing

- 管理SQL程序库缓冲区，实现SQL共享。
  - Oracle将SQL程序（不包括其可执行代码）看作字符串，用散列函数将其变换成一个定长的散列值，作为搜索和比较的手段。
  - 单纯字符串匹配存在不足，需要SQL语句规范化。例如：

```
SELECT *
FROM STUDENT
WHERE SNO=0708123;
```

由于SNO取值不同而不能共享可执行代码。

31

Oracle SQL sharing

- Oracle对SQL共享提供了3种设置：
  - EXACT
  - SIMILAR
  - FORCE

在Oracle的库缓冲区中，每个SQL程序需对外提供两个不同的散列函数值。Oracle推荐EXACT为默认设置

- 库缓冲区匹配命中率（match ratio）是衡量库缓冲区利用效果的依据，如果这个指标太低，库缓冲区不够大也是可能的原因之一，可作为决定扩充库缓冲区的参考。

32

*Egenera*虚拟计算机系统

- 采用虚拟技术，实现了传统IBM、HP等大型主机的系统功能。真正实现了将硬件、软件、网络、服务集约化在一个系统平台上。
- 可在一个整机系统中实现多个逻辑服务器以完成信息中心所有的应用服务功能，逻辑服务器可动态分配系统资源。更有效地配合和利用SAN技术。被誉为下一代数据处理中心的系统平台。
- 逻辑服务器可以安装不同版本的操作系统，和不同类型的操作系统（LINUX、WINDOWS），而传统IBM、HP等大型主机只能运行单一的专有操作系统。
- 虚拟技术极大地提高了整个信息平台的使用效率，大大降低了传统离散系统设备的冗余度，提高处理器利用率达50%以上。
- 内置网络体系。逻辑系统间为点对点交换，速率为1Gbps，并具有网络平衡、网络备份等特点，便于和方便地组织信息中心逻辑服务器之间的信息交换。另外，该系统有4/8个千兆网卡用于与外界相连进行数据交换

33

集约化、简单、虚拟

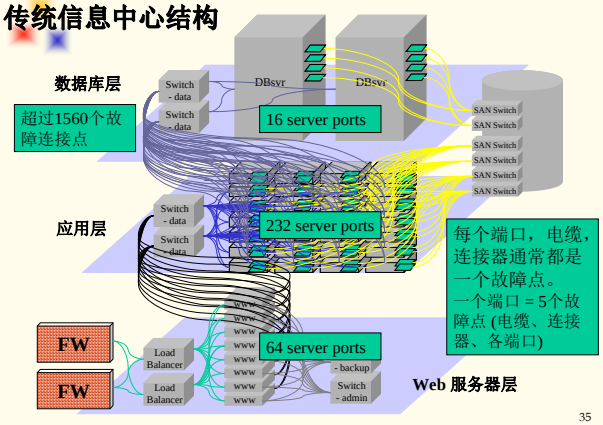
第1层:
Zeus **万维网服务器**
第2层:
应用服务器
第3层:
Oracle 9*i* RAC **数据服务器**

OA应用系统

作复原或扩展备份

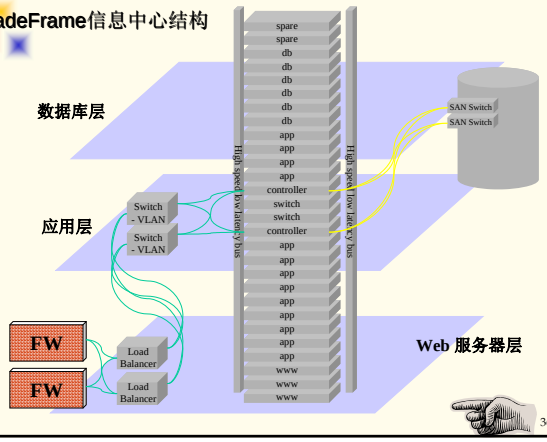
34

传统信息中心结构



35

BladeFrame信息中心结构



36