

ORACLE®



ORACLE®

Oracle数据库规范

Victor Li

Senior Consultant – Oracle Consulting Service

Oracle数据库规范

- 数据库模型设计规范
- SQL开发规范
- 数据库管理规范
- 数据库性能优化建议

建表的参数设置

- 存储参数 PCTFREE

- 对于经常被UPDATE且被update的列所占字节比较大的表，需要设置较大的PCTFREE，避免造成过多行迁移。

- 对于可能造成热点块的表，PCTFREE参数需要设置较大，尽量减少热点块发生概率。

- 事务参数

- 对并发DML事务很大的表，可以在建表初期设置较大的initrans参数。 initrans值过小可能会引起ITL竞争，导致热点块或者enqueue较为严重。

建表的参数设置

- Initrans使用原则

- 每个initrans在数据块上占用24Bytes空间
- Initrans不可以超过10
- 不可以修改maxtrans值
- 修改initrans后只对新的数据块起作用，如需要对全部数据起作用需要重新组织表
- alter index xxx initrans 8;
- Alter table xxx storage (freelists 8) ;

注意: **ASSM**表空间上无需设置**freelists**

列设计

- 定长字符型列使用**CHAR**类型，不定长字符型列使用**VARCHAR2**类型。
- 固定长度用**char**，相对**varchar2**可以提高查询速度
- 不定长度用**varchar2**,如果用**char**，则会在列末用空格补充，在处理上需要用**trim**等函数去空格，因此容易造成处理的不便和效率问题。
- 日期字段需定义为**DATE**类型。如果定义为**varchar2**或者**char**时需要进行转换，影响效率。需要数据精确到微秒的字段需定义为**TIMESTAMP**类型。
- 列为**null**时，需定义**default**值，避免因为**null**而造成索引不能被用到的情况。
- 使用**NUMBER**型时必须指定长度。由**number**得精度和密度来保障数据的一致性

列设计

- 表中字段的命名长度不应该超过20个字节。
- 记录数达到千万级的表，必须进行分区，分区一般遵循以下原则：
 - 数据具有明显的范围属性，比如日期，大小等，且经常进行范围条件查询的表，采用范围分区
 - 数据具有明显的列表属性，比如地点，省份等，且经常用列表条件查询的表，采用列表分区
 - 数据不具有明显的范围属性或者列表属性，且数据量很大，则可以采用hash分区

临时表

- 对于只对本事务有效的临时表使用**ON COMMIT DELETE ROWS**关键字创建该表。
- 对于只对本会话有效的临时表使用**ON COMMIT PRESERVE ROWS**关键字创建该表。
- 对于临时表空间要求比较大的业务系统，临时表要存储在独立的表空间中，并且临时表空间的数据文件需要放在单独的磁盘上。

索引

- 小表（数据量小于**5000**条记录为标准）不需要创建索引。
- 对于**OLTP**应用，分区表使用分区索引。
- 创建或重建大索引时需指定使用**NOLOGGING**子句，指定并行度，指定较大的排序区，提高执行效率。

EXAMPLE:

```
ALTER SESSION SET WORKAREA_SIZE_POLICY=MANUAL;  
ALTER SESSION SET SORT_AREA_SIZE=2000000000;  
ALTER SESSION SET SORT_AREA_RETAINED_SIZE=2000000000;  
CREATE INDEX XXX NOLOGGING PARALLEL X;  
ALTER INDEX XXX NOPARALLEL;
```

- 将记录差别数最多的列放在索引顺序的最前面。
- 对于**OLTP**应用的业务系统索引数据的重复率不能超过**20%**。
- 进行**order by column desc**排序时，创建**column desc**索引。

索引

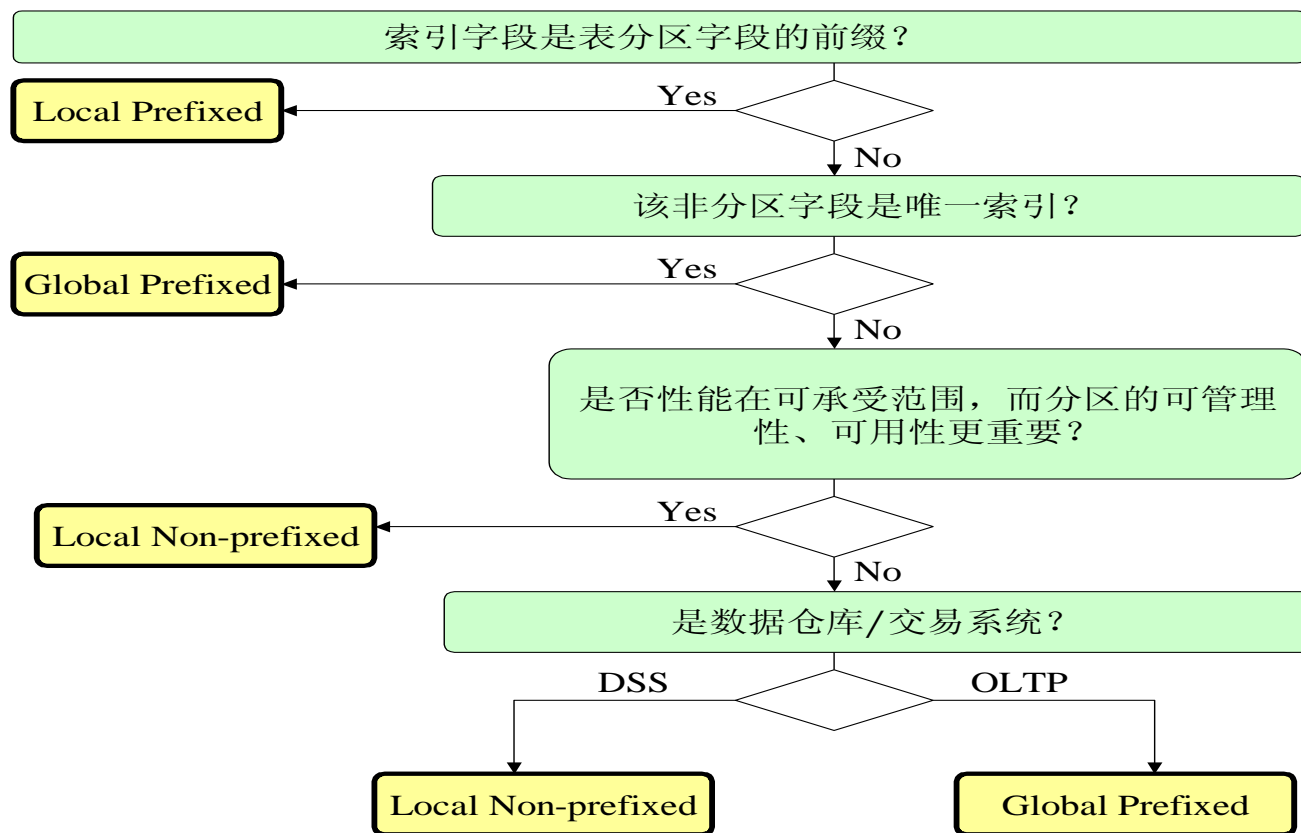
- 频繁使用的**index**需要放入库缓存的**keep**池中。
对于频繁使用的较小的**index**(可以根据**sga buffer**来确定大小的标准)可以直接放入**buffer**的**keep**池，以提高效率。对于较大或者更新比较频繁的表或者索引不建议放在**buffer** 的**keep**池。对于系统经常访问的小表，比如系统配置表等可以直接修改表和索引的属性，以保留在**buffer**池中。
- 在**OLTP**系统中，少用位图索引；对于更新频繁的表，少用位图索引。

索引

- 通过索引查询数据，行数一般不会超过全表的20%，否则，采用全表扫描方式更合适些
- 10,15,20的原则。通过索引查询数据最好是少于表记录数的20%，否则就最好用全表扫描 Tom Kyte建议在20%
- 对索引的使用进行监控，可以发现那些索引有用，那些索引没有用到。没有用到的索引可以直接删除。
- 重建索引的时间最好选择在维护时间窗口，如果一定要在业务处理期间重建索引则使用online选项
- 如果索引的level不高，且不需要挪动索引的存储位置，可以选择 `alter indexcoalesce`
- 当索引的clustering_factor接近表的行数时，可以把表数据按照索引列排序后进行重建。以提高索引查询的效率，接近表的block数是最理想的情况。
- 反向索引不适合于表上含有较多>,>=,<,<=范围查询的系统，否则反而会降低系统的整体性能

分区索引

- 通过下图定义的规则，可确定分区索引类型



分区索引

- **Hash Global分区索引介绍**
- HASH-Partitioned Global索引是Oracle 10g开始提供的新特性。而在以前的版本中，Oracle只支持
- Range-Partitioned Global索引。HASH-Partitioned Global索引的好处如下：
- 比Range-Partitioned Global索引易于实施。HASH-Partitioned Global索引是根据索引字段值，通过Oracle内部的HASH算法自动均匀散列到定义的分区中。而Range-Partitioned Global索引需要根据索引字段值的范围进行分区，因此实施和维护的难度都大。
- HASH-Partitioned Global索引适合于在并发量、吞吐量很大的交易系统（OLTP）中，对某些字段的访问冲突。尤其是sequence字段值。
- HASH-Partitioned Global索引适合于大批量的数据查询。HASH-Partitioned Global索引不仅可以提供分区之间的并行查询，而且在分区内也可进行并行查询的处理。
- 建立分区索引必须指定表空间，并且指定的表空间要与数据表空间分开，这样便于管理，同时尽可能分开索引和数据的IO访问，提高效率

索引

- 索引数量
 - 对于OLTP应用的业务系统，单个表上索引的个数不超过5个。
 - 对于OLTP系统来说，索引数量太多会严重影响DML操作
- 索引类型选择 **btree or bitmap**
 - B-TREE索引，一般情况下选择普通的b-tree索引
 - 位图索引

对于列的distinct value很少，且数据量较大，且DML操作很少，这种情况可以选择使用位图索引。比如性别，distinct value只有男和女且DML操作也会很少，此时选择位图索引，效率可能会更高。

复合索引

当需要创建复合索引时，需要慎重考虑列在索引中的顺序，主要影响的是**INDEX SKIP SCAN**

- 当查询条件中有第一列时，不需要考虑列的分布情况
- 当查询条件中没有第一列时，需要考虑列的分布情况。尽量把**distinct value**不多的列放在复合索引的第一列
- 如果复合索引所包含的字段超过**3** 个，那么仔细考虑其必要性，考虑减少复合的字段；
- 不管是单列索引还是复合索引，当列允许为空，且实际有空值时，在查询中可能会不走索引扫描，因为**null**值不在索引条目中。因此尽可能选择**not null**的列做为索引列，如果不能避免时，可以采用列为空时去一个默认值的方法解决。

索引

- 函数索引
 - 尽量不采用函数索引，函数索引会在insert以及select时多一次函数计算的消耗。
- 递减索引
 - 进行order by column desc排序时，综合考虑，创建column desc索引，不再额外进行排序操作。

视图

- 物化视图的刷新闻隔时间最小为3分钟。
- 物化视图的基表必须创建主键

存储过程、函数和包

- 存储过程、函数和包中不允许频繁的使用DDL语句。
- 存储过程、函数和包必须有相应的出错处理功能。
- 存储过程、函数和包中变量在引用表字段的时候，需使用`%rowtype`类型。
 - `rowtype`可以方便，容易的定义字段类型，当表结构或者字段类型发生变化是不再修改程序
 - `rowtype`在游标使用中 可以整行的接受数据定义，而不需要对一行数据的每个字段进行分别定义

Directory

- 目录设置要求与Oracle系统用户目录分开。
- **Directory**使用权限只赋予需要使用的数据库用户。
- 对应的操作系统目录必须对oracle操作系统用户开放读写权限。
- 定期清理和备份**Directory**对应的操作系统目录。

同义词

- 对于只读用户，必须创建与表相同名字的别名。
- 别名的访问顺序：**public**别名 -> **private**别名 -> 与表同名的对象。

Sequence

- **sequence**要求不连续时，需要指定**cache**和**noorder**，特别是对于**RAC**系统而言，**sequence** 的**nocache**属性可能会带来较大的性能影响。
- 对于**RAC**系统，**sys.audses\$,sys.idgen1\$**两个**sequence**的**cache**值调整为**10000**或者更高，特别是短连接且**logon**频率高的系统更需要调整

LOB对象

Oracle对大对象的操作机制象普通数据一样，因此大对象需要被分解成多个小的数据片段才能完成这种数据操作机制，这种小的数据片段就是CHUNK。

- 使用CLOB/BLOB
- 不可使用raw/raw long
- 分开存储, uniform size tbs
- 使用16K CHUNK SIZE
- 升级到11g的SecureFiles LOB

数据库安全设计

数据库用户安全设计原则

- 数据库用户权限授权按照最小分配原则。
- 数据库用户要分为管理、应用、维护、备份四类用户。
 -
- 不允许使用sys和system用户建立数据库应用对象。
- 禁止grant dba to user。

数据库安全设计

序号	用户类型	描述
1	DATA OWNER	该类用户拥有数据库业务schema 对象，特别是tables及package和Procedure以及其他对象。不对应用程序开放。只有通过对象授权和系统授权，Transaction类型用户才可访问DATA OWNER类型用户；表空间的使用，要通过空间授权配额，才可访问。
2	Transaction	该类型用户拥有数据库最小权限。只有通过角色ROLE对DATA OWNER中的对象进行授权才可访问DATA OWNER中的对象（如GRANT SELECT,INSERT,UPDATE,DELETE ON XXX TO ROLE_1; GRANT TO ROLE_1 TO Transaction等）。一般用于应用程序直接连接数据库进行数据更新操作。
3	READ ONLY	该类型用户拥有且只拥有对DATA OWNER的表或视图有SELECT的权限，并且必须不能有for update的子语句。必须通过角色授权才可以访问。一般用于应用数据查询。
4	Monitor	该类型用户一般用于监控数据库性能，或者是第三方工具使用。作为监控软件，如QUEST监控软件、patrol等；

数据库用户角色及权限设计规范

- 删除或锁定数据库测试用户 **scott**。
- 修改可用用户的默认密码。
- 对查询用户只能开放查询权限。
- 对新建用户初次登录数据库强制修改密码。
- 开发环境，测试环境，生产环境中相同用户的权限设置必须完全一致。

数据库用户角色及权限设计规范

序号	角色类型	描述	
1	DATA OWNER	R_<X>_DB: X为长度为4-8个字符的业务功能简称。	
.2	Transaction	R_<X>_T: X为长度为4-8个字符的业务功能简称。	
3	Read Only	R_<X>_Q: X为长度为4-8个字符的业务功能简称。	
4	Monitor	应按照第三方厂商提供的方式命名。	

Oracle数据库规范

- 数据库模型设计规范
- **SQL开发规范**
- 数据库管理规范
- 数据库性能优化建议

SQL开发规范

- **Select**语句中不可以用*, 必须**select**字段列表, 以节省内存, 提高效率
- 避免频繁**commit**, 尤其是把**commit** 写在循环体中每次循环都进行**commit**。避免在一个事务中出现2此**commit**的现象。如果执行一半执行一次**commit**, 执行完另外一半又执行一次**commit**
- **like** 子句尽量前端匹配, 如写成 '**like string%**', 不要写成 '**%string%**'。
- 批量**insert** 大数据量时可以采用**append**和**nologging**方式, 提高处理速度
- **exp**时可以采用**direct=y,index=no**的方式提高处理效率, **imp**时可以指定较大的**buffer**。如果是**oracle 10g**以上则可以使用**expdp**和**impdp**来提高处理的速度
- **exp**和**imp**时, 如果系统中存在主外键约束, 在**imp**时可以设置**constraints=n**,避免在**imp**时因为主外键而报错。

SQL开发规范

- 在使用for update子句时一定注意限制条件，避免锁定全表或者不需要被锁定的行记录。readonly 的情况不能用for update。
- PLSQL开发中一定注意解析的问题。比如
execute immediate 'select from table where xx=' using '...'而不要直接写成execute immediate '...'
- 尽量少用not exist/not in等否则写法。如果一定要用时，尽量选择not exist，not in可能用不到index,not exist效率更高，速度更快。
 -
- exist和in的选择一般遵循以下原则
 - 当查询的数据较少或者字段值比较时用in比较好，比如select * from t where t.id in (...);
 - 当查询的值比较多或者是子查询时用exist比较好，比如select * from t where exist(select)
 - exist中的子查询用常量是速度比select *快,比如select * from t where exist (select 1 from ...)

SQL开发规范

- 尽量不用select嵌套写法，如select ...from tb1,(select...tb2)...尽可能把select的嵌套转化为连接方式，如select From tb1,tb2 ...
- 尽量少用is null/ is not null等null 的处理。
- 对于大表查询中的列项应尽量避免进行诸如to_char()、to_date()、to_numbre()等转换。
- 尽量避免进行全表扫描，限制条件尽可能多，以便更快搜索到要查询的数据
- 在做全表扫描时，可以先修改session的db_file_multiple_read_count值再进行全表扫描以提高执行效率。
alter session set db_file_multiple_read_count=32

SQL开发规范

- 创建database 对象的时候，对象名不能用“”引上，否则可能会带来找不到对象的问题。
- hash join时注意表连接顺序。且需要注意选择合适的hash key，选择distinct value很多的字段来作为hash key，如果重复值太多，在进行hash探测的时候效率将会很低。
- hash join较多的系统，需要设置较大的pga_aggregate_target参数，以便进行最优的hash join
- 收集统计信息时，尽量在维护时间段进行，否则会造成大面积SQL重新解析，可能导致较为严重的问题。
- 在parallel查询的使用过程中，尽量指定hint来使用而不是在表上加degree参数，这样会导致对该表的所有查询都将偏向于走全表扫描，同时可能会带来内存的大量消耗。对于并行度的问题，不繁忙的系统可以设置并行度为CPU的个数，对于繁忙的OLTP系统，最好不要使用并行查询。
- 在SQL开发的过程中，SQL的字母大小写采用一致规范，避免因为大小写的问题造成SQL的多次解析。

SQL开发规范

- in 替换or，比如select * from t where id=1 or id=2 or id=3,select * from t where id in (1,2,3)
- union 与or，union有时比or更有效，or有时比union更有效，综合union 替换or或者or 替换union后是否能用到索引，以及条件的选择性来综合判断和评估。
- >=替换>，比如select count(1) from t where id>=3 而不用select count(1) from t where id>2
- order by在对索引列进行排序时需要注意
 - order by 后面的字段顺序，应该和索引列(复合索引)的顺序 保持一致。
 - order by后面的字段如果为null，则不能用到索引
select id from t order by id. 索引在id列上，如果id列为null，则该语句要用全表扫描再排序，如果为not null，则直接用索引扫描即可。
如果id列为null，则可以通过用where和order by置换的方式，比如
select id from t order by id 全表扫描再排序
select id from t where id>0 直接扫描索引

SQL开发规范

- 避免数据类型转换，如果 id列为number型,select * from t where id='1'则会发生数据转换id=to_number('1'),如果类似转换发生在索引列上将导致索引不能被用到。
- group by , order by 尽量放到SQL的最末尾，这样可以把不需要返回的行过滤掉，减少需要进行group by order by 操作的集。比如select name from t group by name having name='xxxx' 效率肯定不如select name from t where name='xxxx' group by name;
- !=操作符
除非必须必要，否则坚决避免使用!=操作符，因为使用!=会使ORACLE无法使用索引，Oracle内核会假定查询会返回表中的多数记录，因此就会走FULL TABLE SCAN。

如何评估SQL语句的性能

- 分析与SQL执行相关的配置
 - 参数: `optimizer_index_cost_adj`
 - 参数: `optimizer_mode`
 - 参数: `sql_trace`
 - 参数: `db_file_multiblock_read_count`
- 分析SQL执行计划
 - 查看SQL执行计划有三种方式：
 - 设置`set autotrace:`
 - 通过`explain plan for`
 - 通过`tkprof`工具

通过SQL TRACE分析SQL的执行

SQL_TRACE是Oracle提供的用于进行SQL跟踪的手段，是强有力的辅助诊断工具.在日常的数据库问题诊断和解决中，SQL_TRACE是非常常用的方法。

- SQL_TRACE使用

SQL_TRACE可以作为初始化参数在全局启用，也可以通过命令行方式在具体session启用。

- 10046使用

10046事件是Oracle提供的内部事件，是对SQL_TRACE的增强.

10046事件可以设置以下四个级别:

- 1 - 启用标准的SQL_TRACE功能,等价于sql_trace
- 4 - Level 1 加上绑定值(bind values)
- 8 - Level 1 + 等待事件跟踪
- 12 - Level 1 + Level 4 + Level 8

- 分析

分析sql在执行过程中发生的解析次数，逻辑读次数，变量绑定，所花时间。改变SQL写法，变换查询条件等因素，再进行trace分析解析次数，逻辑读次数，变量绑定等指标，最后判断最优化的SQL。

PL/SQL控制结构编码

PL/SQL提供了三种控制程序流程的语句：

- 条件控制语句 **IF**：使用**ELSEIF** 对相互排斥的条件进行判断。
- 循环语句 **FOR** 、 **WHILE** 、 **LOOP**：不使用**EXIT**和**RETURN**语句退出循环，在一个**LOOP**循环中只能有一个出口；将静态的和固定不变的表达（代码）移到循环体外。
- 跳转语句**GOTO**：不允许使用goto语句。

异常处理

- 过程、函数必须在过程体中作异常捕获，并将异常信息通过 `os_Msg` 变量输出。
- `-1 ~ -19999` 的异常为 Oracle 定义的异常代码。
- 过程、函数必须包含两个输出参数，`on_Flag (number)` 及 `os_Msg (varchar2)`，分别用于标识过程的执行状态及过程提示信息（包括异常情况下的异常信息）。其中 `on_Flag` 有三种取值情况，`0` 表示过程执行成功但无提示信息；大于 `0` 表示过程执行成功但有提示信息；小于 `1` 表示过程执行失败且有提示信息。
- 找不到数据“`NO_DATA_FOUND`”时的异常代码为 `100`，异常处理过程中必须进行特殊判断。
- “`WHEN OTHERS`”必须放置在异常处理代码的最后面作为缺省处理器处理没有显式处理的异常。

Oracle数据库规范

- 数据库模型设计规范
- **SQL**开发规范
- **数据库管理规范**
- 数据库性能优化建议

数据库安装规范

软件。	操作系统版本。	对于经过认证的操作系统版本的要求。
	操作系统补丁。	对于不同版本的操作系统的补丁的要求。
	操作系统内核参数。	对于需要调整的内核参数的要求。
	数据库补丁集版本。	当前的补丁集版本是 10.2.0.5。
	数据库独立补丁。	基于当前补丁集 10.2.0.5 推荐安装的独立补丁。
	CRS 补丁版本。	当前的补丁集版本是 10.2.0.5。
	Oracle 推荐补丁。	10gR2 必须安装当前 Oracle 推荐补丁 10.2.0.5.1。 11gR1 必须安装当前 Oracle 推荐补丁 11.1.0.7.5。 11gR2 必须安装当前 Oracle 推荐补丁 11.2.0.1.3。

- Metalink ID : **756671.1 Oracle Recommended Patches**

数据库安装规范 - AIX

11gR2/11gR1/10g/9i RAC

修改/etc/rc.net文件，加上如下行：

```
if [ -f /usr/sbin/no ] ; then
```

```
/usr/sbin/no -p -o udp_sendspace=1048576
```

```
/usr/sbin/no -p -o udp_recvspace=1048576
```

```
/usr/sbin/no -p -o tcp_sendspace=1048576
```

```
/usr/sbin/no -p -o tcp_recvspace=1048576
```

```
/usr/sbin/no -p -o rfc1323=1
```

```
/usr/sbin/no -p -o sb_max=4194304
```

```
/usr/sbin/no -r -o ipqmaxlen=512
```

```
fi
```


数据库安装规范 - HPUX

11gR2/11gR1/10g/9i RAC

使用命令修改网络配置参数:

```
ndd -set /dev/tcp tcp_conn_request_max 10240
```

```
ndd -set /dev/tcp tcp_syn_rcvd_max 1024
```

```
ndd -set /dev/tcp tcp_time_wait_interval 1000
```

```
ndd -set /dev/tcp tcp_xmit_hiwat_def 65536
```

```
ndd -set /dev/tcp tcp_rcv_hiwat_def 65536
```

```
ndd -set /dev/tcp tcp_xmit_lowat_def 24576
```

```
ndd -set /dev/tcp tcp_fin_wait_2_timeout 60000
```

```
ndd -set /dev/tcp tcp_keepalive_interval 60000
```

```
ndd -set /dev/tcp tcp_ip_abort_interval 60000
```

```
ndd -set /dev/tcp tcp_keepalive_detached_interval 60000
```

```
ndd -set /dev/tcp tcp_rexmit_interval_initial 2000
```

```
ndd -set /dev/tcp tcp_rexmit_interval_min 1500
```

```
ndd -set /dev/tcp tcp_rexmit_interval_max 10000
```

```
ndd -set /dev/sockets socket_caching_tcp 1024
```

或者修改文件/etc/rc.config.d/nddconf

数据库安装规范 - 异步I/O

- AIX系统, oracle使用文件系统
 - AIO的minservers设置为10
 - AIO的maxservers设置为盘的10倍再除以CPU数, 再观察高负载时aio的进程数与maxservers进行比较, 再进行调整
- HPUX系统
 - 设置max_async_ports为oracle初始化参数processes+oracle后台进程的数量
 - 设置aio_max_ops为2048

数据库字符集

数据库字符集的确定非常重要，如果选择不当，会给业务数据的保存带来麻烦。如在现实中，有的汉字保存到数据库时发生乱码，从而使客户的信息不能正常保存和显示。数据库字符集在系统设计开发阶段就应当确定。在数据库系统上线后，再更改数据库字符集，代价会非常大。因为不同的字符集设置之间，存在转换操作，如果不兼容，只能逐条转换。如果业务数据中确定有且只有中文字符，则数据库的字符集一般倾向于选择：

- **ZHS16GBK**：此字符包含了大部需要的汉字字符

如果不确定业务数据将来是否在多个国家使用，则倾向于选择：

- **AL32UTF8**：Oracle推荐使用，此字符集支持大部份现在全世界使用的字符，**AL32UTF8**是双字节字符集，因此在保存非英语文字数据时需要更多的空间，在设计表结构与长度时需要注意此点

数据库参数

- LOCK_SGA

建议设置数据库参数lock_sga将目前数据库SGA索住在物理内存中。同时，还需要修改操作系统参数以配合此参数的使用：

AIX: v_pinshm从缺省的0设置为1。

`/usr/sbin/vmo -r -o v_pinshm=1`

设置可锁住内存的最大值：

`/usr/sbin/vmo -r -o maxpin%=percent_of_real_memory`

$\text{Percent_of_real_memory} = ((\text{size of SGA} / \text{size of physical memory}) * 100) + 3$

HPUX: 修改数据库hpux_sched_noage=178。

- MAX_DUMP_FILE_SIZE

建议不要使用默认值（UNLIMITED），最大设置20M。

数据库参数

- PARALLEL_相关的参数
 - PARALLEL_MIN_SERVER
 - PARALLEL_MAX_SERVER
 - PARALLEL_MIN_PERCENT
 - PARALLEL_AUTOMATIC_TUNING
 - PARALLEL_EXECUTION_MESSAGE_SIZE
 - PARALLEL_ADAPTIVE_MULTI_USER

以上参数在数据库进行进行操作的时候使用，并发技术是ORACLE的非常复杂的技术，并且是高效的技术。

根据实际应用经验，可以设置以下参数：

- PARALLEL_MIN_SERVERS = CPU的个数
- PARALLEL_MAX_SERVERS = 2倍于CPU的个数
- PARALLEL_EXECUTION_MESSAGE_SIZE = 16384

统计信息检查

- 小表，小于1000万行，完全收集

```
exec dbms_stats.gather_table_stats(ownname=>'user_name',-  
  tabname=>'table_name',-  
  estimate_percent => 100,-  
  cascade=>true,-  
  method_opt=>'for all indexed columns size auto');
```

- 大表，大于1000万行，如果维护窗口时间，以及主机资源允许的情况下，完全收集。如果不允许采样比例采用30%

```
exec dbms_stats.gather_table_stats(ownname=>'user_name',-  
  tabname=>'table_name',-  
  estimate_percent => 30,-  
  cascade=>true,-  
  method_opt=>'for all columns size skewonly');
```

Oracle数据库规范

- 数据库模型设计规范
- **SQL**开发规范
- 数据库管理规范
- 数据库性能优化建议

性能检查

- 操作系统性能检查
 - CPU检查：
 - OLTP系统，通常CPU总使用率应该在**50%**（（根据该系统历史CPU情况，可以适当下调）以下，**峰值80%**以下；
 - DSS/DW系统，通常均值在70%以下，峰值在95%以下。
 - IO检查：
 - 通常CPU利用率中WIO%应该在10%以内，持续超过30%，必须立即检查：
 - 磁盘情况
 - 应用情况（不良SQL）
 - 内存检查：
 - 内存空闲率，通常操作系统内存空闲大小（free memory）应该在**5%**左右。

性能诊断工具

- Learn to solve performance problems, using diagnostic tools to such as:
 - OS tools and utilities
 - Vmstat, truss and strace, ps, iostat, netstat
 - Common database tools
 - Statspack, ASH reports, 10046 and TKProf and oradebug
 - Specialized tools
 - OS Watcher [301137.1]

10046 event dbms_system

- 使用dbms_system
 - Enable trace at level 12 for session 1234 with serial# 56789
`execute dbms_system.set_ev(1234, 56789, 10046, 12, "');`
 - To turn off the tracing:
`execute dbms_system.set_ev(1234, 56789, 10046, 0, "');`
- User_background_dump找到trc文件

使用topas/top

```
top - 12:55:06 up 18 days, 21:41,  2 users,  load average: 9.09, 6.58, 3.27
Tasks:  88 total,  12 running,  76 sleeping,    0 stopped,    0 zombie
Cpu(s): 32.5% us,  4.7% sy,  0.0% ni, 62.6% id,  0.1% wa,  0.0% hi,  0.0% si
Mem:    775780k total,  151216k used,   624564k free,   58588k buffers
Swap:  1534196k total,   22548k used,  1511648k free,   37156k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3441	oracle	16	0	2292	296	240	R	18.3	0.0	2:25.78	eatcpu
3440	oracle	16	0	2204	300	240	R	12.9	0.0	2:39.98	eatcpu
3442	root	17	0	2196	300	240	R	4.6	0.0	1:02.55	eatcpu
3439	root	17	0	1996	296	240	R	3.8	0.0	0:56.31	eatcpu
5319	root	16	0	3552	840	660	R	0.8	0.1	0:00.02	top
1	root	16	0	1748	536	476	S	0.0	0.1	0:01.00	init
2	root	34	19	0	0	0	S	0.0	0.0	0:05.62	ksoftirqd/0
3	root	5	-10	0	0	0	S	0.0	0.0	0:00.21	events/0
4	root	5	-10	0	0	0	S	0.0	0.0	0:00.01	khelper
5	root	5	-10	0	0	0	S	0.0	0.0	0:00.00	kblockd/0

10046 Oradebug

- 使用top或topas找到最高CPU的操作系统进程号
- 使用SQL*Plus以sysdba连接数据库
- SQL>alter system set timed_statistics=true;
- SQL>oradebug setospid 2269310;
- SQL>Oradebug unlimit;
- SQL>Oradebug event 10046 trace name context forever ,level 12;
- SQL>Oradebug event 10046 trace name context off;
- SQL>oradebug TRACEFILE_NAME

使用Tkprof 格式化10046跟踪文件

- Tkprof \$HOME/bdump/file_name.trc output.txt
sys=no explain=user/pwd sort=fchcpu





ORACLE IS THE INFORMATION COMPANY