 由于工作需要，笔者在HP UX, Soralis 上面设置了Oracle Intermedia来实现全文检索。目前已经投入实际使用。设置过程中有许多问题和经验，拿来和大家交流。本文依据的是Oracle 8.1.6 和8.1.7两个版本,不能保证适用于其他版本。 目前全文检索功能几乎所有主流数据库都支持。此前笔者曾在sql server 2000上实现，感觉非常简单，方便，但创建全文检索索引的时间比较长，通常要十几个小时。Oracle 的全文检索建立和维护索引都要快得多，笔者的65万记录的一个表建立索引只需要20分钟，同步一次只需要1分钟。但设置就要复杂得多。 一．设置过程 1．首先，检查你的数据库是否安装了intermedia 这可以通过检查是否有ctxsys用户和ctxapp角色(role). 如果没有这个用户和角色，意味着你的数据库创建时未安装intermedia功能。你必须修改数据库以安装这项功能。 修改过程： 运行$ORACLE_HOME/bin/dbassist, 选择modify database, 然后在选择数据库功能时将j server 和 intermedia 都选上（安装intermedia必须同时安装jserver).强烈建议你在做这个改动前先备份整个数据库。 2．设置extproc Oracle 是通过所谓的‘外部调用功能’(external procedure)来实现intermedia的，因此正确地设置extproc是关键一步。 首先要配置listener 使它能监听intermedia 调用的请求。你可以通过运行$ORACLE_HOME/bin/netassit 来进行配置，也可以手工修改配置文件：$ORACLE_HOME/network/admin/listener.ora ,然

后重新启动listener。下面以一个例子来讲述如何手工修改配置文件。 打开listener.ora文件，在修改前，通常有如下内容（假定使用缺省listener）: LISTENER = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = MYDATABASE)(PORT = 1521))) SID_LIST_LISTENER = (SID_DESC = (GLOBAL_DBNAME = mydatabase.world) (ORACLE_HOME = /u01/app/oracle/product/8.1.6) (SID_NAME = mydatabase)) 这个listener还没有配置extproc, 因此，需要为它增加对extproc的监听，办法就是分别增加description 和 sid_desc. 修改后的listner.ora 如下： LISTENER = (DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = MYDATABASE)(PORT = 1521))) (DESCRIPTION = (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC)))) SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = mydatabase.world) (ORACLE_HOME = /u01/app/oracle/product/8.1.6) (SID_NAME = mydatabase)) (SID_DESC = (PROGRAM = extproc) (SID_NAME = PLSExtProc) (ORACLE_HOME = /u01/app/oracle/product/8.1.6))) 注意上面的host, global_dbname,sid_name,oracle_home应填写你的数据库的实际值，但program一项必须填写extproc. 其次，要配置服务器端的tnsnames.ora文件。该文件的位置在$ORACLE_HOME/network/admin下面。同样可以通过运行netasst来进行配置。 在tnsnames.ora文件中需要增加如下一项： EXTPROC_CONNECTION_DATA,EXTPROC_CONNECTIO

N_DATA.WORLD = (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))) (CONNECT_DATA = (SID = PLSExtProc))) 注意其中，KEY和SID必须与listener.ora中的key 和sid_name对应相同。 修改完成后，重新启动listener (先用lsnrctl stop, 然后 lsnrctl start), 然后，使用tnsping 来测试一下是否配置正确： tnsping extproc_connection_data 或者 tnsping extproc_connection_data.world,如果配置正确，会显示： Attempting to contact (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)) OK（140毫秒） 否则请检查你的上述两个文件，并注意，在修改后一定要重新启动listener，但并不需要重新启动数据库。 3．设置词法分析器(lexer) Oracle实现全文检索，其机制其实很简单。即通过Oracle专利的词法分析器(lexer),将文章中所有的表意单元（Oracle 称为 term） 找出来，记录在一组 以 dr$开头的表中，同时记下该term出现的位置、次数、hash 值等信息。检索时，Oracle 从这组表中查找相应的 term，并计算其出现频率，根据某个算法来计算每个文档的得分（score）,即所谓的‘匹配率’。而lexer则是该机制的核心，它决定了全文检索的效率。Oracle 针对不同的语言提供了不同的 lexer, 而我们通常能用到其中的三个： basic_lexer: 针对英语。它能根据空格和标点来将英语单词从句子中分离，还能自动将一些出现频率过高已经失去检索意义的单词作为‘垃圾’处理，如if , is 等，具有较高的处理效率。但该lexer应用于汉语则有很多问题，由于它只认空格和标点，而汉语的一句话中通常不会有空格，因此，它会把整句话作为一个term,事实上失去检索能力

。以‘ 中国人民站起来了 ’这句话为例，basic_lexer 分析的结果只有一个term ,就是‘ 中国人民站起来了 ’。此时若检索‘ 中国 ’，将检索不到内容。 chinese_vgram_lexer: 专门的汉语分析器，支持所有汉字字符集。该分析器按字为单元来分析汉语句子。‘ 中国人民站起来了 ’这句话，会被它分析成如下几个term:‘ 中 ’，‘ 中国 ’，‘ 国人 ’，‘ 人民 ’，‘ 民站 ’，‘ 站起 ’，起来 ’，‘ 来了 ’，‘ 了 ’。可以看出，这种分析方法，实现算法很简单，并且能实现‘ 一网打尽 ’，但效率则是差强人意。 chinese_lexer: 这是一个新的汉语分析器，只支持utf8字符集。上面已经看到，chinese vgram lexer 这个分析器由于不认识常用的汉语词汇，因此分析的单元非常机械，像上面的‘ 民站 ’，‘ 站起 ’在汉语中根本不会单独出现，因此这种term是没有意义的，反而影响效率。chinese_lexer的最大改进就是该分析器 能认识大部分常用汉语词汇，因此能更有效率地分析句子，像以上两个愚蠢的单元将不会再出现，极大 提高了效率。但是它只支持 utf8, 如果你的数据库是zhs16gbk字符集，则只能使用笨笨的那个Chinese vgram lexer. 如果不做任何设置，Oracle 缺省使用basic_lexer这个分析器。要指定使用哪一个lexer, 可以这样操作： 第一．在ctxsys用户下建立一个preference: begin ctx_ddl.create_preference(my_lexer,chinese_vgram_lexer). end. 第二．在建立intermedia索引时，指明所用的lexer: create index myindex on mytable(mycolumn) indextype is ctxsys.context parameters(lexer my_lexer). 这样建立的全文检索索引，就会使用chinese_vgram_lexer作为分析器。 4．使用job定时同步和优化 在intermedia索引建好后，如果表中的数据发生变化，比如

增加或修改了记录，怎么办？由于对表所发生的任何dml语句，都不会自动修改索引，因此，必须定时同步(sync)和优化(optimize)索引，以正确反映数据的变化。 在索引建好后，我们可以在该用户下查到Oracle自动产生了以下几个表：（假设索引名为myindex）：DR$myindex$I，DR$myindex$K，DR$myindex$R，DR$myindex$N 其中以I表最重要，可以查询一下该表，看看有什么内容： 0select token_text, token_count from DRI_RSK1I where rownum这里就不列出查询接过了。可以看到，该表中保存的其实就是Oracle 分析你的文档后，生成的term记录在这里，包括term出现的位置、次数、hash值等。当文档的内容改变后，可以想见这个I表的内容也应该相应改变，才能保证Oracle在做全文检索时正确检索到内容（因为所谓全文检索，其实核心就是查询这个表）。那么如何维护该表的内容呢？总不能每次数据改变都重新建立索引吧！这就用到sync 和 optimize了。 同步（sync）:将新的term 保存到I表；优化（optimize）:清除I表的垃圾，主要是将已经被删除的term从I表删除。 Oracle提供了一个所谓的ctx server来做这个同步和优化的工作，只需要在后台运行这个进程，它会监视数据的变化，及时进行同步。但笔者使用ctxserver碰到了许多问题，Oracle 北京的support也建议不使用，而是用以下的两个job来完成（该job要建在和表同一个用户下）： -- sync: VARIABLE jobno number. BEGIN DBMS_JOB.SUBMIT(:jobno,ctx_ddl.sync_index(myindex)., SYSDATE, SYSDATE (1/24/4)). commit. END. -- optimizer VARIABLE jobno number. BEGIN DBMS_JOB.SUBMIT(:jobno,ctx_ddl.optimize_index(myindex,FU

LL)., SYSDATE, SYSDATE 1). commit. END. 其中， 第一个job的SYSDATE (1/24/4)是指每隔15分钟同步一次，第二个job的SYSDATE 1是每隔1天做一次全优化。具体的时间间隔，你可以根据自己的应用的需要而定。至此，你的全文检索功能已设置完成。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com