分布式多媒体数据库应用

李祥和 芦康俊 (解放军信息工程学院)

摘要:本文从实践出发,探讨分布式多媒体数据库应用技术,旨在尽可能多地挖掘分布式数据库系统 *SYBASE* 在多媒体方面的潜力和优势。

一、多媒体数据库现状

自从 1984 年美国 APPLE 公司推出世界上第一台具有多媒体特性的计算机—MACINTOSH 计算机以来,多媒体这一术语频繁出现在计算机行业,多媒体计算机系统也越来越多地渗透进工业、商业、艺术、教育、家庭等各领域。

把多媒体技术与数据库和网络技术结合起来构成分布式多媒体数据库系统,具有非常广阔的发展前途。就多媒体技术本身来说,涉及到两个问题:庞大的数据冗余量和对数值、图象、动画、声音、视频等信息的一体化管理的复杂程度。而这两个问题恰恰是数据库管理系统的拿手好戏。尤其是一些大型数据库管理系统(如 SYBASE),其主要优点就是可以对庞大数据量进行复杂的管理。而且网络级的多媒体管理可把多媒体应用扩充出更强的可交互性。同时,三者的结合即网络级的多媒体数据库可以使当今的数据库应用具有多媒体的特性。这样的系统结构就是全新的系统集成,又可充分发挥三项技术的长处,相得益彰。

一般地说,能够管理数值、文字、表格、图形、图象、声音等多种媒体的数据库称为多媒体数据库。由于多媒体数据库处理的对象比普通的 DBMS 所处理的对象要复杂的多,因此在实现上应采取不同的措施。总的来说,多媒体数据库的管理方式主要有三种:其一,基于关系模型,但需加以扩充,使之支持多媒体数据类型;其二,基于面向对象的模型来实现对多媒体的描述及操纵;其三,基于超文本模型。

近年来世界上几家大的关系数据库厂家都将原有产

品加以扩充，使之支持多媒体数据库类型，这种做法的好处在于：能够保护用户现有投资，使现有的基于关系数据库的应用系统可以通过一种渐进的方式逐渐演变到多媒体数据库；在格式化和非格化数据之间有一个合理的折中，对于办公自动化、管理信息系统这样一类的典型应用来说，数值、文字等往往是主要的，这部分数据用关系型数据库管理一般更为有效，而图形、图象、声音等非结构化数据通常不作主要成分，完全采用面向对象或超文本形式来存取会大大增加整个系统的复杂度，总的存取效率也会受到影响。这也是我们在开发应用过程中选用SYBASE的一些原因。

二、SYBASE多媒体数据管理

SYBASE数据库系统是目前发展最迅速的关系型数据库产品之一，被誉为数据库产品的后起之秀，SYBASE不但具有高可靠性、数据完整性和联机事务处理功能，而且还具有很强的分布式查询和分布式更新处理功能，它支持多媒体类型并能控制多媒体数据资源，支持多种网络环境，可在几种标准的操作系统下工作，具有令人满意的数据库备份和恢复能力，提供了大量的易于使用的开发工具，其客户／服务器的体系结构，把用户界面与事务控制和数据管理功能明确分开，满足九十年代分布式数据库系统的迫切要求。因此它是目前较理想的一个支持分布式多媒体应用的数据库系统. 1994美国世界杯足球赛组织者采用了SYBASE，其中包括有多媒体事物管理应用(EMA)，该系统为国际和国内的7000名记者提供以往比赛信息、运动员和运动队的文档、比赛统计、标准、结果。这项服务包括有国际足联的官方文档，这是全球性的数据库，其中有自1930年以来世界杯比赛参赛的2万运动员所有比赛的统计信息。

SYBASE SQL SERVER除一般的数字数据和字符数据外还支持多种数据类型，特别是binary,image,text数据类型，text类型的最大长度可达2G个字节，打破了传统的字符数据长度不能超过255个字节(127个汉字)的限定，这为图书资料，档案型数据库管理提供了方便。binary和image(image数据类型的长度最大也可达2G个字节)，为实现多媒体数据管理打下基础。当图象、图形、声音、动画等经各自的电路卡和接口；压缩并转换成数字文件，再利用DB-Library提供的库函数写程序，将此类数据文件装入SYBASE数据中，由SYBASE SQL SETNER进行管理，必要时执行这些程序，使SYBASE数据库中的数据取出，经解压缩处理形成相应的文件输出，再经相关的电路卡处理后播放或显示。其结构如图1所示：

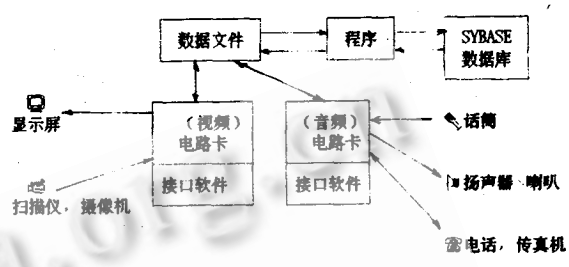

图 1

由于多媒体数据类型MDT(如text,binary,image)与传统的数据类型有较大差别，因此SYBASE在MDT的存储上也做了特殊的处理。专门开辟了多媒体数据区，用于存储多媒体数据。如果一个记录中包含有一个或多个MDT的域，则数据库表中该记录MDT域的值不是多媒体数据本身，而是有关该字段的存储结构及状态信息，如起始地址、终止地址、读写指针及当前读写状态等。而多媒体数据存放在专门的多媒体数据区。

下面以房产公司房源表为例叙述SYBASE对text，和image类型数据的存取方法。设fyb房源表的格式如下：

column	Datatype	size	NullsAllowed:
fy_id	char	12	NO
fy_pic	image		yes

.....

SYBASE可以用两种方法取字段fy_pic的值。

1.select语句

其所选的数据长度依赖于全局变量textsize。

首先用set语句定义欲得到值的长度。

```
set textsize 256 或 set textsize 0 (定义 textsize 缺省值 32K)
```

然后再用select语句

```
select fy_pic from fyb where fy_id="A_001_1234"
```

2.readtext语句

该语句需要给出读写指针，偏移量及长度等信息。

```
declare @Val Varbinary(16)
Select @ Val=text ptr(fy_pic) from fyb where
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn

```
fy_b="A_001_1234"
    readtext fyb.fy_pic @Val 0 256
```

SYBASE 插入修改多媒体数据也与普通型数据不同,仍以 fyb 房源表为例说明对 fy_pic 字段的修改过程,首先建立 fy_pic 列的文件指针:

```
update fyb Set fy_pic=Null where fy_rid="A_001_1234"
```

然后在相应行的 fy pic 列上写实际数据:

```
declare @Val Varbinary(16)
Select Val=textptr(fy_pic) from fyb where fy_id="A_001_1234"
writetext fyb.fy_pic Val"* * * * *,....* * * *"
```

三、网络环境下 SYBASE 对多媒体数据的操作

在网络环境下,SYBASE 的一个应用程序对数据库表中含有多媒体数据(text,brinary 及 image 型)的记录进行操作的步骤。(设 fyb 库表存放在名为 xsb 的数据库中)

```
#include <stdio.h>
#include <sybfront.h>
#include <sybdb.h>
#include "sybdbex.h"
#define IMAGE_LENGTH ((long)301)
#define FY_ID ((char *)"A_001_1234")
main()
{
DBPROCESS *q_dbproc, *u_dbproc;
LOGINREC *login;
char date[IMAGE_LENGTH];
RETCODE result_code;
BYTE *ret_p ;          /*存取图象指针*/
DBBINARY *txptr;       /*正文指针*/
DBBINARY *ty_ptr;      /*时间戳指针*/
if (dbinit()==FAIL) exit(ERREXIT); /*初始化 DB-library*/
login=dblogin(); /*初始化 LOGINREC 结构*/
DBSETLUSER(Login,USER);
DBSETLPWD (Login,PASSWORD);
q_dbproc=dbopen(login,Null);/*打开到 SQL Server 的联结*/
u_dbproc=dbopen(login,Null);
dbuse(q_dbproc,"xsb"); /*置当前库*/
dbuse(u_dbproc,"xsb");
getimage(data); /* 取图象,产生某些随机信息存入 data 中,具体过程略*/
/* step 1:插入数据 */
dbfcmd(q_dbproc,"insert fyb (fy_id) values ('%s')",FY_ID);
dbsqlexec(q_dbproc);
while(dbresults(q_dbproc)!=NO_MORE_RESULTS) continue;
/* step 2:更新新插入的记录*/
dbfcmd(q_dbproc,"update fyb set fy_pic=mull where fy_id= '%s'",FY-ID);
dbsqlexec(q_dbproc);
while(dbresults(q_dbproc)!=NO_MORE_RESULTS) continue;
/* step 3:查找新插入的记录中图象属性列 */
dbfcmd(q_dbproc,"select fy_pic from fyb where fy_id='%s'",FY_ID);
while((result_code=dbresults(q_dbproc))!=NO_MORE_RESULTS)
    if(result_code=SUCCEED)
        dbcanquery (q_dbproc); /*查询目的是得到正文指针*/
    /* setp 4:取正文指针,正文时间戳指针*/
txptr=dbtxptr(q_dbproc,1);
ts_ptr=dbtimestamp(q_dbproc,q);
    /* step 5: 写图象数据*/
dbwritetext(u_dbproc,"fyb.fy_pic",txptr,DBTXPLEN,ts_ptr,TRUE,IMAGE_LENGTH,data);
    /*下面我们把刚插入的图象数据取出*/
dbfcmd(q_dbproc,"select fy_pic where fy_id='%s'",FY_ID);
dbsqlexec(q_dbproc);
while((result_code=dbresuts(q_dbproc))!=NO_MORE_RESULTS)
    if(result_code=SUCCEED)
        white(dbnextrow(q_dbproc,1)!=NO_MORE_RESULTS)
            ret_p=dbdata(q_dbproc,1);
    /*从数据库中删除新插入的记录*/
dbfcmd(u_dbproc,"delete fyb where fy_id='%s'",FY_ID);
dbsqlexec(u_dbproc);
while(dbresults(u_dbproc)!=NO_MORE_RESULTS)
        dbcanquery(u_dbproc);
dbexit();
}
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn