2007 年秋季学期“分布式数据库”课程

实验系统测试方案说明

（学生版）

- beta 3 Last modified: 01:12, 2008.1.6 -

清华大学计算机系软件所数据库研究组

目　录

1. 测试数据

本章介绍实验系统所需的测试数据。

测试数据涉及 4 个全局表，82360 条记录，分成 14 个（只涉及水平和垂直划分，以下简称“基本划分”）或 16 个（混合划分）分片，分配在 4 个逻辑站点上。

本章内容安排如下：第 1.1 节定义 4 个全局表；第 1.2 节介绍相应的数据划分方案，包括基本划分和混合划分；第 1.3 节介绍站点配置和数据分配方案；第 1.4 节定义测试数据文件的格式，并在附录 A 中给出了一个将该格式的数据导入到 Access 数据库的 Java 程序的例子；最后，附录 B 中给出了一个生成本章所述的数据划分的脚本。

1.1. 全局表

本次测试所用的全局表共有四个，分别是 Student、Teacher、Course、Exam：

Student (id int key, name char(25), sex char(1), age int, degree int)
Teacher (id int key, name char(25), title int)
Course (id int key, name char(80), location char(8), credit_hour int, teacher_id int)
Exam (student_id int, course_id int, mark int)

下面分别进行说明：

Student (id int key, name char(25), sex char(1), age int, degree int)

“学生”表，共 15000 条

各列说明：

名称	含义	类型	值
id（主码）	学号	整数	100001 至 115000，无重复
name	姓名	字符串	-
sex	性别	字符串	'M': 男，约占 50% 'F': 女，约占 50%
age	年龄	整数	18 至 28 之间均匀分布
degree	学生类型	整数	1: 本科生，约占 50% 2: 硕士生，约占 33.3% 3: 博士生，约占 16.7%

Teacher (id int key, name char(25), title int)

“教师”表，共 5003 条

各列说明：

名称	含义	类型	值
id（主码）	编号	整数	200001 至 205003，无重复
name	姓名	字符串	-
title	教师类型	整数	1：讲师，约占 40% 2：副教授，约占 40% 3：正教授，约占 20%

Course (id int key, name char(80), location char(6), credit_hour int, teacher_id int)

“课程”表，共 2357 条

各列说明：

名称	含义	类型	值
id（主码）	课号	整数	300001 至 302357，无重复
name	课程名	字符串	-
location	上课地点	字符串	'CB-6'：六教，约占 33.3% 'FIT'：FIT 大楼，约占 33.3% 'ZJ'：紫荆公寓，约占 33.3%
credit_hour	学分	整数	1：约占 20% 2：约占 35% 3：约占 35% 4：约占 10%
teacher_id	教师编号	整数	Teacher.id

Exam (student_id int key, course_id int key, mark int)

“考试成绩”表，共 60000 条

各列说明：

名称	含义	类型	值
student_id	学生学号	整数	Student.id
course_id	课程号	整数	Course.id
mark	考试分数	整数	大致符合均值为 80，方差为 20 的正态分布，并且最大值限制为 100。其中： ● 低于 60 的约占 15% ● 高于 90 的约占 30%； ● 其余的约占 55%

1.2. 数据划分

Student (id int key, name char(25), sex char(1), age int, degree int)

水平划分：

分片名	划分条件
Student.1	id<105000
Student.2	id>=105000 and id<110000
Student.3	id>=110000

Teacher (id int key, name char(25), title int)

水平划分：

分片名	划分条件
Teacher.1	id<201000 and title<>3
Teacher.2	id<201000 and title=3
Teacher.3	id>=201000 and title<>3
Teacher.4	id>=201000 and title=3

Course (id int key, name char(80), location char(6), credit_hour int, teacher_id int)

方案一，垂直划分：

分片名	划分条件
Course.1	(id, name)
Course.2	(id, location, credit_hour, teacher_id)

方案二，混合划分：

Step 1. 垂直划分（同方案一）

分片名	划分条件
Course.1	(id, name)
Course.2	(id, location, credit_hour, teacher_id)

Step 2. 将 Course.2 进一步水平划分如下：

分片名	划分条件
Course.2.1	credit_hour=1
Course.2.2	credit_hour>1 and credit_hour<4
Course.2.3	credit_hour=4

Exam (student_id int key, course_id int key, mark int)

水平划分：

分片名	划分条件
Exam.1	student_id<107000 and course_id<301200
Exam.2	student_id<107000 and course_id>=301200
Exam.3	student_id>=107000 and course_id<301200
Exam.4	student_id>=107000 and course_id>=301200

1.3. 数据分配

站点配置

共 4 个站点：Site 1 ~ Site 4，部署在局域网内的 3 台机器上。

数据分配

基本划分：

站点名	分片名
Site 1	Student.1 Teacher.1 Course.1 Exam.1
Site 2	Student.2 Teacher.2 Course.2 Exam.2
Site 3	Student.3 Teacher.3 Exam.3
Site 4	Teacher.4 Exam.4

混合划分：

站点名	分片名
Site 1	Student.1 Teacher.1 Course.1 Exam.1
Site 2	Student.2 Teacher.2 Course.2.1 Exam.2
Site 3	Student.3 Teacher.3 Course.2.2 Exam.3
Site 4	Teacher.4 Course.2.3 Exam.4

1.4. 测试数据的发布

为了做到平台无关，我们采用纯文本文件来发布 1.1 节所述的数据。该数据文件的格式定义如下：

```
Line 1:    table_name_1  number_of_rows
Line 2:    column_value_1_1  column_value_1_2  …   column_value_1_n1
Line 3:    column_value_2_1  column_value_2_2  …   column_value_2_n1
…
Line N:    table_name_2  number_of_rows
Line N+1:  column_value_1_1  column_value_1_2  …   column_value_1_n2
…
```

其中：

1) “Line 1:”等表示行号，不会出现在文件中；
2) table_name_i 表示第 i 个表的名称，表的出现顺序与 1.1 节中的顺序一致；
3) number_of_rows 表示该表共有多少行；
4) column_value_j_k 表示第 i 个表的第 j 行的第 k 个属性的值，属性出现顺序与 1.1 节中各表的属性顺序一致；
5) 各个属性值之间使用制表符（'\t'）分隔；
6) 对于字符串，用单引号（'\''）将其引起来，如；'xiao ming'；
7) 任何字符串中都不会出现制表符、单引号或双引号。

各组只需编写少量的代码即可解析并加载该格式的数据，附录 A 中给出了一个将该格式的数据导入到 Access 数据库的 Java 程序的例子。

2. 测试流程

本章给出检查实验时的参考测试流程。整个测试分为 6 部分：

1. 初始化数据库
2. 单条数据的插入、删除测试
3. 导入数据
4. 查询测试
5. P2P 测试（任选）
6. 自由展示（任选）

其中，各组可以在自由演示部分尽量充分的展示所做系统的新特性、新功能、或是其它与众不同的设计与实现，测试人员注意仔细记录。

各小组实现的系统必须提供良好的用户界面，要求如下：

1) 可以清晰的显示当前的数据字典。各组必须提供如下信息的显示：
   - 站点信息，
   - 全局表的信息，
   - 统计信息，
   - 分片信息；

2) 可以方便的导入数据，并且显示导入结果。各组必须提供如下信息的显示：
   - 成功或失败，
   - 导入数据总量，
   - 所用时间；
3) 可以方便的输入 SQL 语句（插入语句、删除语句、查询语句），并且可以对用户的不同输入显示相应的结果。各组必须提供如下信息的显示：
   - 执行 SQL 语句所涉及的站点，
   - 插入、删除、查询等命令执行是否成功，
   - 执行时间，
   - 查询语句的返回结果集，包括行、列计数，
   - 优化后的查询树、查询执行计划；
4) 对系统运行时出现的错误和用户输入的错误给出相应的提示。

2.1. 初始化数据库

按第 1 章所述，配置站点并建立各个表（基本划分）。

2.2. 插入与删除

Insert 和 Delete 测试一共 10 条（Insert 和 Delete 各 5 条），依次执行之后数据库应该刚好为空。这里仅分别公布 2 条，正式测试时会加入另 6 条。

1) insert into Student values (190001, 'xiao ming', 'M', 20, 1)
   涉及站点：Site 1
2) insert into Teacher values (290001, 'Santa Claus', 2)
   涉及站点：Site 3
3) delete from Teacher where title=1
   涉及站点：Site 1、Site 3
   删除 Site 1 上的(200001, 'St. Nicholas', 1)
4) delete from Teacher where id>=290000 and title=2,
   涉及站点：Site 3
   删除 Site 3 上的(290001, 'Santa Claus', 2)

2.3. 导入数据

数据将以文本文件的形式发布，格式请参考第 1.4 节。文本文件总大小约为 2MB，导入时间大约为 1 分钟（Java+Access 数据库）。

正式检查时的数据可能会有一些微小变化，但数据分布和数据总量不变。

2.4. 查询测试

基本划分（所有组必测）

查询测试用例共 9 条，这里公布 7 条作为例子，正式测试的时候会加入另 2 条。

1) 查询所有学生的信息
   select * from Student
2) 查询所有课程的名称
   select Course.name from Course
3) 查询所有学分大于 2 且上课地点在六教的课程信息
   select * from Course where credit_hour>2 and location='CB-6'
4) 查询所有考试的课程号和成绩
   select course_id, mark from Exam
5) 查询由正教授讲授的学分大于 2 的课程的课程名、学分、教师姓名

```
select    Course.name, Course.credit_hour, Teacher.name
from      Course, Teacher
where     Course.teacher_id=Teacher.id and
          Course.credit_hour>2 and
          Teacher.title=3
```

6) 查询各个学生的所有考试成绩

```
select    Student.name, Exam.mark
from      Student, Exam
where     Student.id=Exam.student_id
```

7) 查询年龄大于 26 且参加了授课地点不在六教的课程的考试的学生的 id，姓名，考试分数，以及相应的课程名称

```
select    Student.id, Student.name, Exam.mark, Course.name
from      Student, Exam, Course
where     Student.id=Exam.student_id and
          Exam.course_id=Course.id and
          Student.age>26 and
          Course.location<>'CB-6'
```

混合划分（提高要求，任选）

查询测试用例共 2 条，这里公布 1 条作为例子，正式测试的时候会加入另 1 条。

首先，按照第 1 章所述配置各个站点并建立各个表（混合划分），然后执行查询：

1) 查询所有课程的信息
   select * from Course

2.5. P2P 测试（任选）

在 2.4 给出的 9 个基本划分下 SQL 查询中找一个不访问所有站点的，比如查询 2（只需要访问 Site 1）。然后分别测试：

1) Site 1 正常接入，断掉 Site 2，系统应该正常给出结果；
2) 断掉 Site 1，Site 2 正常接入，系统应给出相应的提示。

3. 补充说明

有以下几点需要补充说明：

- 除了本文档，请大家注意仔细阅读期中实验辅导 PPT，注意上面的要求；
- 条件表达式中‘NOT’操作符可以不做；
- 注意“诱导删除”问题：比如删除 Course 表中的数据时，删除语句中的条件不一定只涉及 Course 的主码；
- 每组检查时间约为 1 小时，其中我们的测试约为 50 分钟，各组自由展示的时间为 10 分钟左右；
- 请各小组在检查之前提前配置好所需设备，保证检查人员到达后即可开始检查；
- 由于时间有限，请各小组仔细设计自由展示部分（任选），尽量在最短的时间内最充分的展示自己系统的特别之处；
- 如对测试数据或检查流程有任何意见或建议，或者发现了本文档叙述上的任何错误或前后不一致之处，请尽快我联系：

haowu06@mails.tsinghua.edu.cn 吴昊 13699184416

附录

A. 将文本数据导入到 Access 数据库示例

```
public class Loader
{
  public static void load()
  {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    String db = "jdbc:odbc:Driver={MicroSoft Access Driver (*.mdb)};DBQ=dataset.mdb";
    Connection conn = DriverManager.getConnection(db, "", "");
    Statement stmt = conn.createStatement();
    BufferedReader br = new BufferedReader(new FileReader("dataset.txt"));
    while(true) {
      String line = br.readLine();
      if(line==null)
        break;
      String[] fields = line.split("\t");
```

```
      String tableName = fields[0];
      int n_rows = Integer.parseInt(fields[1]);
      for(int i = 0; i<n_rows; i++) {
        line = br.readLine();
        fields = line.split("\t");
        String q = "insert into "+tableName+" values (";
        for(int j = 0; j<fields.length; j++) {
          q += fields[j];
          if(j<fields.length-1)
            q += ",";
        }
        q += ")";
        stmt.executeUpdate(q);
      }
    }
    stmt.close();
    conn.close();
  }
}
```

B. 数据划分定义脚本

B1. 基本划分

```
define site S1 127.0.0.1:2001
define site S2 127.0.0.1:2002
define site S3 127.0.0.1:2003
define site S4 127.0.0.1:2004
//
create table Student (id int key, name char(25), sex char(1), age int, degree int)
create table Teacher (id int key, name char(25), title int)
create table Course (id int key, name char(80), location char(8), credit_hour int, teacher_id
    int)
create table Exam (student_id int, course_id int, mark int)
//
fragment Student horizontally into id<105000, id>=105000 and id<110000, id>=110000
fragment Teacher horizontally into id<201000 and title<>3, id<201000 and title=3,
    id>=201000 and title<>3, id>=201000 and title=3
fragment Course vertically into (id, name), (id, location, credit_hour, teacher_id)
fragment Exam horizontally into student_id<107000 and course_id<301200, student_id<107000
    and course_id>=301200, student_id>=107000 and course_id<301200, student_id>=107000
    and course_id>=301200
//
allocate Student.1 to S1
allocate Student.2 to S2
allocate Student.3 to S3
allocate Teacher.1 to S1
allocate Teacher.2 to S2
allocate Teacher.3 to S3
allocate Teacher.4 to S4
allocate Course.1 to S1
allocate Course.2 to S2
allocate Exam.1 to S1
allocate Exam.2 to S2
allocate Exam.3 to S3
```

```
allocate Exam.4 to S4
```

B2. 混合划分

```
define site S1 127.0.0.1:2001
define site S2 127.0.0.1:2002
define site S3 127.0.0.1:2003
define site S4 127.0.0.1:2004
//
create table Student (id int key, name char(25), sex char(1), age int, degree int)
create table Teacher (id int key, name char(25), title int)
create table Course (id int key, name char(80), location char(8), credit_hour int, teacher_id
    int)
create table Exam (student_id int, course_id int, mark int)
//
fragment Student horizontally into id<105000, id>=105000 and id<110000, id>=110000
fragment Teacher horizontally into id<201000 and title<>3, id<201000 and title=3, id>=201000
    and title<>3, id>=201000 and title=3
fragment Course vertically into (id, name), (id, location, credit_hour, teacher_id)
fragment Course.2 horizontally into credit_hour=1, credit_hour>1 and credit_hour<4,
    credit_hour=4
fragment Exam horizontally into student_id<107000 and course_id<301200, student_id<107000
    and course_id>=301200, student_id>=107000 and course_id<301200, student_id>=107000
    and course_id>=301200
//
allocate Student.1 to S1
allocate Student.2 to S2
allocate Student.3 to S3
allocate Teacher.1 to S1
allocate Teacher.2 to S2
allocate Teacher.3 to S3
allocate Teacher.4 to S4
allocate Course.1 to S1
allocate Course.2.1 to S2
allocate Course.2.2 to S3
allocate Course.2.3 to S4
allocate Exam.1 to S1
allocate Exam.2 to S2
allocate Exam.3 to S3
allocate Exam.4 to S4
```