国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
一九八六年第三期　总第五十五期　No. 3 1986 Sum.55

分布式数据库操作的形式化描述

张　滨

提　要　这篇论文提出了一种分布式数据库的形式化描述表示，如何将分布式数据库的全局操作变为局部操作。这种形式化描述可以作为描述全局层与局部层操作转换的工具，也可以作为分布式程序语言的起点。从实用化角度来说，也可以作为一般的 D—DBMS 操作部分实现的设计工具。

一、引　　言

分布式数据库是网络技术和数据库技术的统一，即分散与集中的统一。计算机网络能使利用通讯线路相互连接的计算机系统之间的分布数据与程序适应地域分散的情况，而数据库技术是抽象的集中数据的管理方法，它给用户提供一个总的聚合的数据集合，这两种表面上矛盾的方法既是产生分布式数据库思想的基础，也给分布式数据库提供了实现的技术手段，分布式数据库利用网络技术把数据库系统的技术水平提高了一步，使数据库技术更适应于社会、经济、军事组织的需要。

当前的分布数据库多用关系模型，因而可将它视为分布在计算机网络各个节点上的关系的集合。从用户角度来看，它是透明的，即用户可认为这些关系都在一个集中的数据库中；从系统管理者的角度来看，各节点有各自的独立性和控制权，而在更高一层即全局层上又要实现协同管理，这种协同管理要求各局部层之间有灵活的信息流动过程和信息管理。

二、形式化描述的功能

分布式数据库的构成有两种方法：一种是由底向上，即整体分布数据库是由一些不同的、异质的集中数据库构成；另一种是由顶向下的，即将一个集中数据库划分成若干分布的数据库，而这些库都分别存于网络的各不同节点上。

关系型分布式数据库的关系有的存在于一个计算机节点上，称为局部关系；有的分散在计算机多个节点中，称为全局关系。

本文 1986 年 1 月 10 日收到

分布式数据库的操作不仅涉及到操作所发生的各局部节点，而且还涉及到网络中的其他节点。当一个操作启动时，它要查出将要操作的关系在哪些节点上，并且要传播这个操作或这个操作的某些子动作到相应节点的相应关系段上。在这篇文章中，我们提出一种形式化的描述表示这种传播和相应的操作以及它们的逻辑关系。这种形式化描述不仅表示要执行什么操作而且还表示这些操作如何执行，比如是顺序的还是并行的。

这种形式化描述不考虑分布数据库的产生，即假设数据库已经按各种分布标准分布完毕。整个数据库系统对用户来讲也是透明的，即用户不用考虑数据的分布。用户可用关系名、属性名和D—DBMS提供的特别机制去修改和检索数据并可进行并行与复盖控制。

对于各种数据分布标准，每种操作在全局层要求：

- 在分布关系的各段上要进行哪些操作或动作
- 这些动作如何执行，顺序的或并行的
- 总的结果如何由各子结果组成；为了保证整个数据库的一致性，各子动作之间应如何协调

这种形式化描述就可实现以上功能。

分布式数据库操作的形式化描述以下简称 DDBOF (Distributed Database Operation Farmalism)。

DDBOF 不是作为一种新的 DML（数据操纵语言）提出来的，而只是在分布数据库的全局和局部层之间的一种映射描述。它可以作为提供给一般 D—DBMS 的工具，能够作为一种全局操作的综合表示方法，也对分布数据库的并行操作优化执行提供了描述手段。

DDBOF 定义全局操作，指明在各子动作间是顺序或并行地执行，最后用布尔函数表示是否成功。

三、数据分布描述

在分布式数据库中关系划分一般有横向和纵向划分两种，划分后的关系称为段。

1. 横向划分即将关系按元组分段，按限制条件将一个关系分成若干个元组集合。这些限制条件可以是表达式，也可以是序号域。划分之后，如对于各段重新定义关键字，则还必须测试关键字的唯一性；在进行全局修改时，还有必要按原限制条件控制元组的移动。

例 关系 Supplier (SNO, SNAME, TOWN)，设属性 TOWN 域值有 “Paris”, “NewYork” 和 “London”，则可横向划分 Supplier 为：

Supplier1：=Supplier：TOWN= “*Paris*”

Supplier2：=Supplier：TOWN= “*NewYork*”

Supplier3：=Supplier：TOWN= “*London*”

其中 TOWN= “*Paris*” 这种形式为关系限制条件，是关系的选择操作。

2. 纵向划分是将关系按属性进行分段。一般来说，每段都应包含划分之前的关键字属性，扩展这个条件，我们可以在划分之后重新定义一个关键字属性，值得注意的是

可能要压缩某段，使它的关键字属性值是唯一的。

例 关系PART (PNO, PNAME, COLOR, PRICE) 可纵向划分为：

PART1：=(PNO, PNAME, COLOR)PART

PART2：=(PNO, PRICE)PART

其中 PNO 为原关系关键字，这种形式表示关系的投影操作。表示成一般表达式

PART：=PART1 * PART2

其中 * 为自然连接。

关系 PART 也可以划分为

PART1：=(PNO,PRICE)PART

PART2：=(PNAME,COLOR)PART

其中 PART2 段关键字定为 PNAME，则必须将 PNAME 中重复的值删除（假定 PNAME 和 COLOR 是一致的，即货主的货颜色都相同），这样 PART2 才可独立成为一关系段。

3. 横、纵向划分可以组合，即对已经横向划分的段可再纵向划分，反之亦然。

4. 另一种特殊的分法就是按邻关系的横向划分法。即关系R和S有某一共同属性X，若S按其他的属性横向划分，而这划分又必将X属性分成若干域值，则R按X属性所被分的域值进行横向划分。当然有个条件即X属性的相同值必须对应于S的划分属性域中同一值。

例 关系 Customer (CNO, CNAME, SNO)

Supplier (SNO, TOWN)

关系 Supplier 按TOWN 横向划分，则 SNO 必被分成几个值域段（假定 SNO 为关键字属性）。我们将 Customer 按这几个 SNO 值域段横向划分。

总之，我们可描述关系划分如下：

G(K,X)：K为关键字属性，X为非关键字属性

G(K,X)：$=G_1$(K,X1)*G_2(K,X2)*…*G_n(K,X_n)

G_i(K,Xi)：=UNION((K,Xi)Tj)

Tj：——Lj： 一个局部关系

——Lj：Wj 一个有限制Wj的局部关系

——Lj*Vj 邻关系横向划分（诱导划分）

其中 UNION 为横向组合。

另外，冗余性是分布数据库的特点，数据分布于各个节点，关系的分布类型包括局部的、全局的、全部重复的、部分重复的等等。

- 局部关系：（L-Relation）即整个关系存在于一个节点。
- 全局关系：（G-Relation）即关系的各段分存于不同节点。
- 完全重复关系：同一关系产生多个副本，分存于不同节点。
- 部分重复关系：关系的属性或某些元组在整个系统中有重复。如纵向划分中的关键字属性；横向划分中某些限制条件定义相容的元组。

若把子关系也看成关系的话，上面的定义亦成立。

四、关系数据库的基本操作

关系数据库有四种基本操作，即检索、插入、修改、删除。

1—GET(R)：检索R的全部元组。R为关系名或关系表达式。

2—INSERT(R,t)：将元组 t 插入到关系R中。

3—DELETE(R,t)：从关系R中删除元组 t .

4—MODIFY(R,t,t′)：用 t′ 元组代替 t 元组，假定这种修改只涉及到非关键字属性。

对这几种操作定义两种判定结果：一个称为实际结果，即得到一个元组或一组元组或元组的操作变化状况，用R—Result表示；另一个称为布尔结果，表示操作成功与否，用B—Result表示。

对各种操作，两种结果应如下：

操 作	R-Result	B-Result
GET (R)	T	True if T=R
INSERT (R,t)	RUt	True if $t\in R$
DELETE (R,t)	R—t	True if $t\notin R$
MODIFY (R,t,t′)	(R-t)Ut′	True if $t\notin R$ and $t'\in R$

T表示检索出的元组或元组集。

B—Result中的R表示实际结果。

下面给出两个谓词操作：

—KEY(R,t)：给定元组 t 及 t 在关键字属性 k 上的投影，如结果对R来说是关键字匹配的，则 KEY(R,t) 为真。

—PRED(R,t)：表示关系R按一断言分段，若元组 t 满足这个断言分段，则 PRED (R,t) 为真。

五、DDBOF 说明

下面的形式化描述中用 OP 表操作，OP表示其B—Result，而 NOT OP表示B-Result之反。

定义几种表达式：

1) 简单布尔表达式：如OP1. AND. OP2. OR. OP3

2) 条件表达式 (C—expr)：其中 expr 表示操作表达式。如 expr1：(expr2/expr3)

其意义可用如下一段程序表示：

```
Perform (expr 1)
  if B—Result (expr 1)
    then Perform (expr 2)
```

B—Result: =B—Result (expr 2)
else Perform (expr 3)
B—Result: =B—Result (expr 3)
end if

如表达式为 expr1: (expr 2) 则

Perform (expr 1)
if B—Result (expr 1)
then perform (expr 2)
B—Result: =B—Result (expr 2)
else B—Result: =B—Result (expr 1)
end if

3) 并行操作表达式 (P—expr): 设a、b 表 expr

P—OR (a. OR. b)
P—AND (a. AND. b)
P—XOR (a. AND..NOT. b.OR.b. AND..NOT.a)

P—NOR, P—NAND, P—NXOR 为以上表达式之反, 如 P—NOR(.NOT.(a.OR.b))

定义 P—@ (expr1, expr2) 的含义为

Perform in Parallel expr 1 and expr 2
B—Result: =B—Result(expr1)@B—Result(expr 2)

其中@可以为 'OR', 'AND', 'XOR', 'NOR', 'NAND', 'NXOR'.

对 n 元操作, 可定义为

P—@ (expri) for i=1,2,…,n
P—@ (expr1,expr2,…,expr n)

expr i 可以是: i) T或F
ii) 操作 Get, Insert, Delete, Modify, Key, Pred 等
iii) 一简单、条件或并行表达式

4) 控制表达式: 在全局关系上的操作传播可用它来表示, 形式为

OP: =expr⇒RC-OP

其意义为

Perform (expr)
if B—Result (expr) then
Perform RC-OP end if
B-Result: =B—Result (expr)

其中 OP 是在全局关系上的操作, 如 Get, Insert, Delete, Modify 等, expr 可以是各种表达式, RC—OP 是组合关系操作 (如UNION, JOIN 等) 或交付操作 (COMMIT)。

DDBOF 即利用这些表达式来表示分布数据库的全局操作与局部操作的逻辑联系, 表示它们之间的转换以及各种操作的逻辑执行。

下面给出 DDBOF 的描述：

FORMALISM

说明：{ }表示选择项，[ ]表示可省缺项，( ) 表示参数项。

• 传播 (Propagation)

DML－Operation：＝expr⇒RC－OP

• 表达式 (Expr)

{Term
Term Bool－OP expr
.NOT.(expr)
C－expr
P－expr}

• 布尔操作符 (Bool－OP)

{.AND.
.OR.
.XOR.}

• 项 (Term)

{DML－Operation
True
False}

• 条件表达式 (C—expr)

expr：(expr[/expr])

• 数据操纵语言操作 (DML—Operation)

{{KEY
PRED
INSERT
DELETE} (Relname，tuple)
MODIFY (Relname,tuple1,tuple 2)
GET (Relname)}

• 并行操作表达示 (P—expr)

Parallel—OP (P—expr—List)

• 并行操作符 (Parallel—OP)

{P－AND
P－OR
P－XOR
P－NAND
P－NOR
P－NXOR}

• 传播的交付 (RC－OP)

{{UNION
JOIN} Sub－expr
COMMIT}

• 并行操作表达式列表 (P－expr－List)

{expr,P－expr－List
Sub－expr}

• 子表达式 (Sub－expr)

{expr i
L－relation i}

注：L-relation 为局部关系或段。

上面只给出了一些基本定义，还可以扩展到诸如关系的移动，关系查找，关系一致性和完整性检验等操作。进一步要做的工作是将这形式化描述具体化，对一些限制条件，关系分配地点，关键字属性等辅助参数给出定义及表示。

六、示　　例

1. 在一个全局关系上检索

G：=UNION (L1,L2,…,Ln) 则 Get(G) 为

UNION (Get(L1), Get(L2),…, Get(Ln))

又各 Get(Li) 可并行实现，则：

Get(G)：=P－AND(Get(Li))⇒UNION(Get(Li))

2. 全局关系的操作传播

对一个全局关系的操作要转化为对这关系所在的各节点上子关系的操作集。要注意的是在一个节点上检索一个全局关系并修改之后可能要重新分布这个全局关系。

G—rel 由 Li (i=1,…,n) 子关系组成

G－Update(G)：=f(Li)⇒COMMIT

用 Update 表示各类有关的操作，而 f 表示有关的简单、条件、并行表达式。

3. 对前面定义的 Supplier 全局关系插入元组 t .

INSERT (Supplier,t)：=

P－XOR(PRED(Supi,t)：(INSERT(Supi,t)))⇒COMMIT

其中 i=1,2,3

4. 对全局关系G修改，用 t′ 代替 t

MODIFY(G,t,t′)：=	解　　释
P—OR(PRED(Li,t′)：	(1) t′ 符合某段限制条件
(KEY(Li,t)：	(2) 若 t 存在于 Li 中
(MODIFY(Li,t,t′)	(3) 则修改之
/INSERT(Li,t′))	(4) 否则插入 t′
/KEY(Li,t)：	(5) 如 t′ 不合限制条件而 t 存在于 Li 中
(DELETE(Li,t))))	(6) 删除 t 元组
⇒COMMIT	(7) P—OR 为真则交付此操作并保证执行

5. 对于横向分段的全局关系的插入操作

i) 在操作传播之前检验其关键字的唯一性

INSERT(G,t)：=P—NOR	解　　释
(KEY(Li,t)：	(1) 如关键字值不存于任一段
(P—XOR(PRED(Li,t)：	(2) t 满足一子关系限制条件
(INSERT(Li,t))))	(3) 插入此段
⇒COMMIT	(4) 交付

ii）在操作传播过程中检验关键字的唯一性

```
INSERT(G,t)：=                          解    释
P—XOR(PRED(Li,t)：                (1) t元组符合限制条件
(KEY(Li,t)：                      (2) t关键字值存于Li中
(False/INSERT(Li,t))              (3) 若不存在则插入
/KEY(Li,t)：(False))))            (4) t不合限制条件，且关键字存在，不插
                                      入，结果为假
⇒COMMIT                           (5) P—XOR为真则交付
```

6. 对于纵向划分的全局关系的插入操作

i）检查各段的关键字值之后插入

```
INSERT(G,t)：=P—AND(KEY(Li,t)：
                (False/INSERT(Li,ti)))
               ⇒COMMIT
```

ii）若各段关键字属性相同，只检查一个段

```
INSERT(G,t)：=KEY(Li,t1)：(False/P—AND(INSERT(Li,ti)))
     ⇒COMMIT
```

iii）例子：给定两局部子关系

EMP (END, ENAME, SAL, DNO)；关键字为END

DEPT (DNO, MGR)；关键字为DNO

G—relation定义为：

ED(END,ENAME,SAL,DNO,MGR)：=EMP*DEPT

关键字为ENO，且假定DEPT中DNO域值与EMP中域值相同

(DNO)EMP=(DNO)DEPT

要求在ED中插入一个元组t。

A）检查t的END值是否已存在于ED中，若真，则操作结果为假。

B）若t的END值不存在于ED中，则

a） 如DEPT和EMP关系是层次的，则t中DND值必须存在于DEPT中，否则结果为假。

```
INSERT(ED,t)：=KEY(EMP,t)：(False/KEY(DEPT,t)
  ：INSERT(EMP,t)))⇒COMMIT
```

b） 如t元组DNO值不在DEPT中，则需插入t到EMP和DEPT中

```
INSERT(ED,t)：=KEY(EMP,t)：(False/INSERT(EMP,t)
.AND. KEY(DEPT,t)：(True/INSERT(DEPT,t)))
 ⇒COMMIT
```

引进并行操作

```
INSERT(ED,t)：=KEY(EMP,t)：(False/
       P—AND(INSERT(EMP,t),
       KEY(DEPT,t)：(True/INSERT(DEPT,t))))
```

⇒COMMIT

7. 按邻关系划分的关系元组的插入

假定全局关系G是按邻关系V划分的，且假定LGi，LVi为全局关系G和V的各子关系，假定G的属性集包括V的关键字属性，为了决定t插入G的哪一子关系，要检验t中包含V的关键字属性的值。

```
INSERT(G,t): =P—XOR(KEY(LVi,t):
                    (KEY(LGi,t):
                    (False/INSERT(LGi,t))))
                    ⇒COMMIT
```

七、总　　结

文章中定义了几种数据分布的方法，如横向的或纵向的，全局的和部分重复的等等。对分布数据库的全局操作可认为是对每个子部分操作的结果组合，这种形式化描述方法可用来描述全局操作如何变为局部操作的集合并如何实现它们。

DDBOF可作为一种分布式程序语言的起点，也可作为一实现D—DBMS的设计工具。

李金汉老师为本文提出许多宝贵意见，在此致谢。

参 考 文 献

[1] Stefano. Ceri & Giuseppe Pelagatti, Distributed Database Principles & Systems, Mc Graw-Hill Book Company, Newyork, 1984.

[2] Michel Adiba & Juan Mannel Andrade, Update Consistency and Parallelism in Distributed Database, The 2ND International Conference on Distributed Computing System, Paris, 1982.

[3] C. J. Data, An Introduction to Database Systems, Volume II. Addison—Wesley Publishing Company, Inc, 1983.

The Formalism of Distributed Database Operations

Zhang Bin

Abstract

The thesis suggests a kind of formalism of distributed database operations to present how to change the general operations into the local ones. This formalism can be used as a tool to describe the transformation between general level operations and local operations, and can also be used as the beginning of the distributed program language or a design tool to realize the operational part of a general D—DBMS.