分布式系统数据一致性处理的研究

Research on Consistency Processing of Data of Distributed System

成汝震[1] 尚志恩[1] 刘宏忠[2] 张运凯[2]

(河北师范大学教信学院计算机系[1] 网络中心[2] 石家庄050016)

Abstract Through structure and analyse of data object attribute, this paper discusses the scheme on consistency processing of distributed bot multimedia data of structured and common data. The scheme provides a new way for solving difficult problem of distributed data on consistency processing.

Keywords Distributed system, Distributed data, Data consistency, Data-oriented

1 引言

分布式系统中的数据是一组结构化数据的集合，这些数据在逻辑上属于同一系统，在物理上分布在计算机网络不同的节点上，换句话说，在物理节点上的这一组数据是以数据库为单位的[1]。在这里所讨论的数据包括已结构化的多媒体数据，数据库指的是多媒体数据库。多媒体数据指的是除了包括普通数据之外，还包括声音、图像等其他形式的数据。声音、图像等多媒体数据是无结构的数据，一般在多媒体数据库中对它进行了结构化处理，所谓的结构化处理指的是如何使它们与有结构的数据共存，如建立指针等，详细参看Adjeroh 和 Nwosu[2]及 Pazandak 和 Srivastava[3]对多媒体数据库系统研究的若干问题。

数据的分布处理，历来是数据库系统设计中最为复杂的问题。对于传统数据的分布处理，已有多种系统加以研究，如 PORL、ADA-DDM、SIRIUS-DELTA、MULTIBASE 和 Distributed INGRES 等。我们在这里讨论的是包含了已结构化的多媒体数据的分布处理一致性的问题，它与传统数据的分布处理和集中式多媒体数据的处理是有所不同的，许多 MDBS 只注重对多媒体数据的处理，而忽视对数据的管理[4]。在这里我们采取的方法是以构造数据对象属性为基础，用数据对象属性的改变来描述对数据的处理，这样就可以不用把多媒体数据与普通数据分开讨论，即可以统一讨论分布数据一致性处理的问题了。

2 数据对象的构造与分类

2.1 必要的约定

为了讨论的方便，设 NODE$_i$ 是管理体系中具有独立管理功能的一个单元(如具有独立财务核算的管理系统或档案管理系统等，可以是局域网)；A_i 为 NODE$_i$ 的原始数据集；F_i 为在 A_i 上的某数据处理方案(如某些代数的、统计的或辅助决策的运算等)，显然，在原始数据集 A_i 上根据管理要求不同，有着不同的数据处理方案 F_i，无论怎样都不会影响我们的讨论；B_i 为在 NODE$_i$ 上对 A_i 经过 F_i 处理后得到的数据集，称之为信息集，即：

$$F_i(A_i)=B_i(i=1,2,\cdots,n)$$

一般地，$A=\sum A_i$ 为 NODE 上的原始数据集，设 F 为在 A 上的数据处理方案，且 $F(A)=B$，一般地，以下关系不成立，即：

$$F\neq\sum F_i \text{ 且 } B\neq\sum B_i$$

NODE 也是管理体系中具有独立管理功能的一个单元(如可以从管理职能上去理解 NODE，此时 NODE$_i$ 隶属于 NODE)，NODE 与 NODE$_i$ 之间的关系如图1所示。

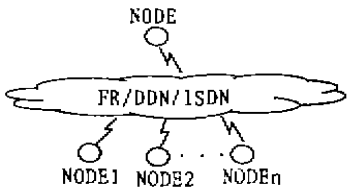

图1 NODE 与 NODE$_i$ 关系图

由图可看出，NODE 与 NODE$_i$ 之间的信息交换构成了基于网络的分布式系统，在这里 $A_i(i=1,\cdots,n)$ 是分布的，因此对管理系统而言，数据分布一致性问题是整个系统能否处理成功的关键。

2.2 数据对象的构造

对任意一个有意义的原始数据 $x\in A_i$，把它构造为一个数据对象 a_{ij}，设 $a_{ij}\in A_i(i=1,2,\cdots,n;j=1,2,\cdots,m)$ 表示 A_i 中的任一数据对象，它可由以下五部分组成：

$$a_{ij}=a_{ij}^1+a_{ij}^2+a_{ij}^3+a_{ij}^4+a_{ij}^5$$

a_{ij}^1：关键字属性，它由编码集组成，由它唯一确定该数据对象；a_{ij}^2：操作属性，它对该数据对象实施增加、删

除、修改等操作加以描述；a_{ij}^2：传递属性，描述该数据对象传递信息时的状态（待传递、已传递）；a_{ij}^4：量值属性，表示该数据对象的量值（如原始数据 $x=a_{ij}^4$）；a_{ij}^5：处理属性，对该数据对象进行处理时的状态（待处理、处理毕）。

2.3 分布式系统数据对象的分类

根据数据在 NODE 与 NODE$_i$ 之间的传递方式，分布式系统数据对象可分为两类：

1. 仅进行单向传递的数据对象（即：数据传递由 NODE 指向 NODE$_i$，或由 NODE$_i$ 指向 NODE），我们称之为第一类数据对象 C_1。

2. 进行双向传递的数据对象（即：数据传递由 NODE 指向 NODE$_i$，同时也可由 NODE$_i$ 指向 NODE），我们称之为第二类数据对象 C_2。

显然，对 C_2 处理过程中的最大问题，是所谓数据的交叉更新，亦即在 NODE 和 NODE$_i$ 上存在着对同一数据对象同时做更新处理的事件。

3 一致性处理方案

3.1 对数据对象的操作描述

设 Add 表示对数据库的增加操作，Edit 表示对数据库的修改操作，Del 表示对数据库的删除操作，即：

$$Add(a_{ij})\in A_i+\{a_{ij}\}$$

$$Edit(a_{ij})=a_{xy}\in A_i$$

$$Del(a_{ij})\in A_i-\{a_{ij}\}$$

分别表示在 A_i 里增加一条记录、修改一条记录和删除一条记录。Inverse 表示取相反数（记录）操作，设 $a_{ij}=a_{ij}^1+a_{ij}^2+a_{ij}^3+a_{ij}^4+a_{ij}^5$，则

$$Inverse(a_{ij})=a_{ik}=a_{ij}^1+a_{ij}^2+a_{ij}^3+(-a_{ij}^4)+a_{ij}'$$

或 $$Inverse(a_{ij})=a_{ik}\in A_i+\{a_{ik}\}-\{a_{ij}\}$$

表示在 A_i 里增加一条与 a_{ij} 数据对象的可计量化部分取值相反后得到的数据对象 a_{ik}，同时去掉数据对象 a_{ij}。

3.2 数据对象传递操作描述

由于分布数据（包含多媒体数据）传递时需要在网上进行，对于网络的带宽、传输协议和存储设备性能等，这些条件虽然影响数据的传递，但不是主要的。对于多媒体数据来说，分布式多媒体数据传递的关键是多媒体时序同步问题[1]，我们在处理数据传递时，采取的原则是，锁定相关的字段（因为无结构的数据已作了结构化的处理），如果其他进程访问时则等待，等到处理完毕，解锁后再允许其他进程访问。我们在数据对象属性中设计了"待传递"和"已传递"的属性，在方案讨论过程中，用这两个属性描述数据的传递操作。为了讨论方便，永远假设数据传递是成功的。

3.3 一致性处理方案

无论数据是从 NODE$_i$ 传递到 NODE，还是从 NODE 传递到 NODE$_i$，从下面讨论的过程可以得出结论：数据已被一致性处理是根据数据对象的属性 a_{ij}^3=="已传递"和 a_{ij}^5="处理毕"来体现的。

3.3.1 C1类型一致性处理操作

设数据传递方向，由 NODE$_i$ 指向 NODE。

1. 对数据库的增加操作。若 $Add(a_{ij})\in A_i+\{a_{ij}\}$，在 NODE$_i$ 端，置 a_{ij}^4="增加"，a_{ij}^1="待传递"；在 NODE 端处理完成时有 $Add(a_{ij})\in A+\{a_{ij}\}$，并返回信息通知 NODE$_i$ 端，两端均置 a_{ij}^3="已传递"，a_{ij}^5="处理毕"。

2. 对数据库的删除操作。若删除数据对象 $a_{ij}\in A_i$，则应根据 a_{ij} 在 NODE$_i$ 端是否已参加 F$_i$ 处理（如帐务处理等）决定在 NODE$_i$ 端的删除方案；在 NODE 端的删除应根据 a_{ij} 在 NODE 端是否已参加 F 处理（如帐务处理等）而定。

1）在 NODE$_i$ 端的删除方案是：

①若 a_{ij} 在 NODE$_i$ 端未参加 F$_i$ 处理（如帐务处理等），则：

$$Del(a_{ij})\in A_i-\{a_{ij}\}$$

同时置 a_{ij}^2="删除"，a_{ij}^3="待传递"，a_{ij}^5="处理毕"。

②若 a_{ij} 在 NODE$_i$ 端已参加 F$_i$ 处理，则：

$$Add(Inverse(a_{ij}))\in A_i+\{Inverse(a_{ij})\},\ Add(a_{ik})\in A_i+\{a_{ik}\}$$

同时置 a_{ij}^2="删除"，a_{ij}^5="处理毕"，a_{ik}^1="增加"，a_{ik}^3="待传递"。

需要说明的是，在这里执行的条件是该数据对象的关键字不变，即数据对象 a_{ij} 和 a_{ik} 的关键字是一样的；属性的变化是描述了其操作的过程，如 a_{ij}^2="删除"，a_{ik}^1="增加"，这里 a_{ij}^2 与 a_{ik}^1 是同一个属性，其不同是描述了它的操作过程。下面对于这种情况的删除操作和修改操作的道理相同，不再赘述。

2）在 NODE 端的删除方案是：此时应根据 a_{ij} 在 NODE 端是否已参加 F 处理（如帐务处理等）而定。

①如果未参加 F 处理，则：

$$Del(a_{ij})\in A-\{a_{ij}\}$$

并返回信息通知 NODE$_i$ 端，置 a_{ij}^3="已传递"。

②如果已参加 F 处理，则：

$$Add(Inverse(a_{ij}))\in A+\{Inverse(a_{ij})\}$$

$$Add(a_{ik})\in A+\{a_{ik}\}$$

并返回信息通知 NODE$_i$ 端，置 a_{ik}^1="删除"，a_{ik}^3="已传递"，a_{ik}^5="处理毕"。

3. 对数据库的修改操作。若修改数据对象 $a_{ij}\in A_i$，则应根据 a_{ij} 在 NODE$_i$ 端是否已参加 F$_i$ 处理（如帐务处理等）决定在 NODE$_i$ 端的修改方案；在 NODE 端的修改应根据 a_{ij} 在 NODE 端是否已参加 F 处理（如帐务处理等）而定。

1）在 NODE$_i$ 端的修改方案是：

①若 a_{ij} 在 NODE$_i$ 端未参加 F$_i$ 处理（如帐务处理

等)，则：

$Edit(a_{ij})=a_{ip}\in A_i$

同时置 a_{ij}^2="修改"，a_{ij}^3="待传递"，a_{ij}^5="处理毕"。

②若 a_{ij}在 NODE$_i$ 端已参加 F$_i$ 处理，则：

$Add(Inverse(a_{ij}))\in A_i+\{Inverse(a_{ip})\}$

$Add(a_{ik})\in A_i+\{a_{ik}\}$

置 a_{ij}^2="修改"，a_{ij}^5="处理毕"，a_{ik}^2="增加"，a_{ik}^3="待传递"。

2)在 NODE 端的修改方案是：

此时应根据 a_{ij}在 NODE 端是否已参加 F 处理(如帐务处理等)而定。

①如果未参加 F 处理，则：

$Edit(a_{ij})=a_{ip}\in A$

并返回信息通知 NODE$_i$ 端，置 a_{ij}^3="已传递"。

②如果已参加 F 处理，则：

$Add(Inverse(a_{ij}))\in A+\{Inverse(a_{ip})\}$

$Add(a_{ik})\in A+\{a_{ik}\}$

并返回信息通知 NODE$_i$ 端，置 a_{ik}^3="修改"，a_{ik}^5="已传递"，a_{ik}^5="处理毕"。

从以上讨论可知，对一些数据的更新，特别是对异地数据的更新，往往不能直接进行，而需要考虑到与更新的数据有关联的处理事件。

同理可考虑，由 NODE 指向 NODE$_i$ 的传递情况。

3.3.2 C_2类型一致性处理说明 对于第二类数据对象 C_2，实践中可能会出现在 NODE 和 NODE$_i$ 上同时对同一数据对象进行增加、删除和修改操作，由此而产生奇解数据，使得应用系统无法确定该数据是否有效。

实践证明，解决 C_2一致性处理的较好方案为：

1. 首先，在任何一个应用系统的设计中必须规范化编码体系，使确定数据对象的编码在系统中具有唯一性和可行性。无论在 NODE 端还是在 NODE$_i$ 端，一旦新的数据对象产生，不允许对其关键字作变更。这个道理是显而易见的，不再赘述。

2. 解决方法：设数据对象 $a_{ij}\in A_i$(或 A)(i=1,2,…,n;j=1,2,…,m)，由前所述，a_{ij}可表示为：

$a_{ij}=a_{ij}^1+a_{ij}^2+a_{ij}^3+a_{ij}^4+a_{ij}^5$

其中属性 a_{ij}^3描述了该数据对象进行信息传递时的状态，即待传递和已传递；a_{ij}^5描述了该数据对象进行信息处理时的状态，即待处理和处理毕。在此我们就是利用这两个属性，把对第二类数据对象 C_2的分布处理过程分为两个阶段：预处理阶段和处理毕阶段。亦即 a_{ij}的属性

a_{ij}^3="待传递"和 a_{ij}^5="待处理"

的处理阶段称为预处理阶段。它说明在该阶段内，新的数据对象 a_{ij}的产生是预产生，只有当经过 NDOE 端(或 NODE$_i$ 端)确认且接到确认的通知后，数据对象 a_{ij}才称为有效，此时置 a_{ij}的属性

a_{ij}^3="已传递"和 a_{ij}^5="处理毕"

只有这两个属性都置为这种状态时，才称为进入到处理毕阶段，否则称之为正在处理。

这样就实现了 C_2到 C_1变换，可应用对 C_1的处理方法实现对 C_2的处理。

3.4 方案讨论

总起来说，对分布数据一致性问题的讨论，可以归类为对 A$_i$ 和 A，B$_i$ 和 B(i=1,2,…,n)这四方面数据的讨论，即对这四方面数据的组成及保持数据一致性方法的讨论。纵观对一致性问题的研究，就数据的组成而言，一般人们或者只讨论传统的数据，或者只讨论多媒体数据，而我们提出的方案是把两者统一讨论的；分析这四方面数据，显然 A$_i$ 是其他三方面数据的基础，所以一般人们只是注重讨论 A$_i$ 和 A 数据之间一致性问题，而忽略对 A$_i$ 和 B$_i$、A 和 B 数据之间一致性问题的讨论，在我们提出的方案中，不仅讨论了 A$_i$ 和 A 数据之间一致性问题，而且还讨论了 A$_i$ 和 B$_i$、A 和 B 数据之间一致性问题。这不仅是全面的讨论问题，就用户而言，他们更关心的是信息集 B$_i$ 和 B，因此信息集 B$_i$ 和 B 也是数据一致性问题的重点讨论的内容。在我们提出的方案中，定义了数据对象的属性，只要某个端点数据集合属性 a_{ij}^3="已传递"和 a_{ij}^5="处理毕"，或者两者均为空(表示没有更新的数据)，就可以进行 F$_i$ 或 F 处理，即可得到信息集 B$_i$ 和 B。这种处理是动态的且随机的，只要用户需要就可以得到，而且得到的数据是完整的且最新的，即得到的数据与整个系统的数据是一致的。这就解决了在分布式系统中，一个端点已对数据集进行了 F$_i$ 或 F 处理，而另一个端点又传来需要更新数据的问题。因为对于检测 a_{ij}^3="已传递"和 a_{ij}^5="处理毕"，或者两者均为空的算法是非常容易实现的。

结束语 本方案基于数据对象的构造，把声音、图像等多媒体数据和普通数据放在一起，统一定义了对象的属性，并且统一讨论了分布数据一致性处理的问题。不但讨论了传统的分布数据一致性处理的问题，而且还讨论了已对数据进行方案处理后又出现了新的更新要求的分布数据一致性处理问题，这说明本方案是解决分布数据一致性处理难题的一种新的解决方案。

参考文献

1 巩志国，周龙骧，董淑珍. 分布式多媒体数据库系统. 软件学报，2000，11(1)：40～48

2 Aljeroh D A，Nwosu K C. Multimedia Database Management-requirements and issues. IEEE Multimedia，1997，4(3)：24～33

3 Pazandak P，Srivastava J Evaluating Object DBMSs for Multimedia. IEEE Multimedia，1997，4(3)：34～49

4 周龙骧，柴兴无. 多媒介数据库系统的分析及设计. 软件学报，1995，6(2)，69～77