

# 一个基于 SQL 的数据库开发环境

北京华资总公司信息部 朱建民 韩燕波

**摘要:** 本文提出了一个基于 SQL/DS、面向 IBM 中型机系列和 286/386 系列微机所构成的多机系统的数据库应用开发环境 APROT。该系统在联机仿真和 SQL/DS 数据库管理系统的基础上大大地扩展了微机的数据处理功能,并增强了中型机与微机的数据共享能力。研制它的目的在于充分发挥和合理利用中型机和微机资源,形成一个具有多机集中分布处理特点的大型数据库应用系统软件开发环境。

## 一、引言

目前 IBM 提供中型机(以下简称主机)的汉字处理终端设备主要是 5550 个人计算机,我部开发了 3270 汉字智能仿真系统,使得国产 286、386 系列微机与主机连接成为它的仿真终端。在现行环境中,微机要么作为 GW3270 仿真终端要么作为完全独立的系统使用。显然,仅仅依靠微机开发大型数图文综合信息处理系统(数据量在千兆比特以上)是不能满足综合数据库本身及其应用系统的开发需求的。而将微机仅作为主机的终端来开发应用会存在以下几个问题:

1. 面向程序员的开发界面差,开发周期长,开发出来的程序可复用性差,也难维护和升级。
2. 用户界面呆板,友好程度差。
3. 微机只作为大机器的仿真终端,导致无法发挥微机自身处理能力也无法利用微机上一些较为成熟的软件产品、开发工具和环境,因此造成微机资源浪费。

IBM 中型机具有数据管理功能强(备有较大的数据库管理系统)、共享程度高、查询方便、统计速度快、外存容量大等特点。另一方面,微机具有开发环境强、软件丰富、用户界面友好,以及容易实现程序或模块的标准化和通用化等优点。要在已配备的 IBM 系列中型机和 286、386 系列微机这样一个基本环境下建立大型 MIS,迫切需要对现行环境进行改造,使微机成为主机的智能终端,两者互补结合成一个基于 IBM SQL/DS、面向多计算机的大型数据库软件开发环境。本文将详细介绍这样一个基于分布式 SQL 服务的数据库应用开发环境(A

database PROgramming Toolbox upon cooperative SQL services,简称 APROT)。

## 二、设计思想与结构

我们的出发点是利用 IBM 系列中型机和微机两者的优势,开发出前台后台一体化的数据处理环境,较为彻底地解决单纯在微机或单纯在 IBM 中型机上开发与运行应用系统的弱点,形成一个具有多机分布处理特点的大型数据库应用系统的软件开发环境。它的基本思想是:

1. 在主机上建立一个数据库服务程序,它和 SQL/DS 一起构成数据库服务器(DBMS Server)服务器承担大量的数据存储、处理、耗时耗空间极大的计算、处理等工作。这样,数据库建立在主机上,所有 SQL 服务命令(包括数据库的数据定义、数据操纵、数据控制)均由主机完成(服务命令来自微机)。对某些需作特殊处理的数据加工任务(如复杂的统计分析, DSS 模型运算)也可由主机完成,以减轻微机部分的负载。

2. 微机作为客户机(Client Machine)完成应用系统中处理人机交互操作的部分,如数据的采集、校验、显示、打印输出、图形用户界面(GUI)的构造等等。这样,微机完成部分的数据加工和所有 I/O 处理,并通过可把访问主机资源的服务命令嵌入在本地事务(程序)中,以实施对主机 SQL 库操作的实时控制。

主机和微机具体任务分工如下:

主机作为数据库服务器 Server:

1.接受微机关于数据定义、数据操纵、数据控制的 SQL 语句服务命令或去执行激活某种特殊事务的命令。

2.动态执行此 SQL 服务命令或激活某事务。

3.完成对数据库的数据提取、增、删、改、建窗口、授权控制。

4.向微机发送提取的数据和执行的的结果信息。

微机作为客户机器 Client 完成:

1.根据应用程序的要求组织完整的 SQL 服务命令,之后向主机发送此命令(即对 SQL/DS 的操作命令),并接收主机以记录形式发送过来的数据或结果信息。

2.数据加工处理:统计、分析、打印等

3.数据维护:录入、修改,有两种方式:

(1)联机维护--以记录形式

(2)脱机维护--以库为单位,文件形式成批倒入

SQL/DS 库。

4.用户界面:菜单操作画面,数据显出画面、统计图形、求助信息等。

5.图形、图象显示处理(显示在图形工作站上)。

这样,就构成了一个 Server--Client 的应用模型。从资源的利用角度,微机既可利用本身所拥有的资源,又拥有主机部分的数据库资源,同时也发挥主机的优势,达到数据共享;从程序设计角度看,微机程序员面对的是一个大于微机本身的虚机,他可在微机应用程序中方便地完成对主机 SQL/DS 数据库的各种操作。

APROT 的 SQL 服务命令中的数据操纵语句(DML)、数据定义语句(DDL)和数据控制语句(DCL)的定义方式与 SQL 语句一致,是由微机产生,借助于主机的 SQL/DS 中的 DBMS 进行处理的,APROT 可分为二大部分。

1.支持大型 MIS 开发的工具,包括:通用菜单模块、通用查询模块、通用报表打印机模块、通用数据维护(录入)模块、通用统计分析模块等。

2.前后台一体化数据库服务程序,这是 APROT 的 MIS 开发工具建立的基础,这个服务程序的主机部分由于需要对 SQL/DS 进行动态库操作,我们采用 IBM370 汇编语言嵌入 SQL 语句的方式处理。微机部分与仿真板关系密切,需要对位、字节和地址的操作,我们用 C 语言和部分汇编语言编程。

APROT 的整个结构如图 1:

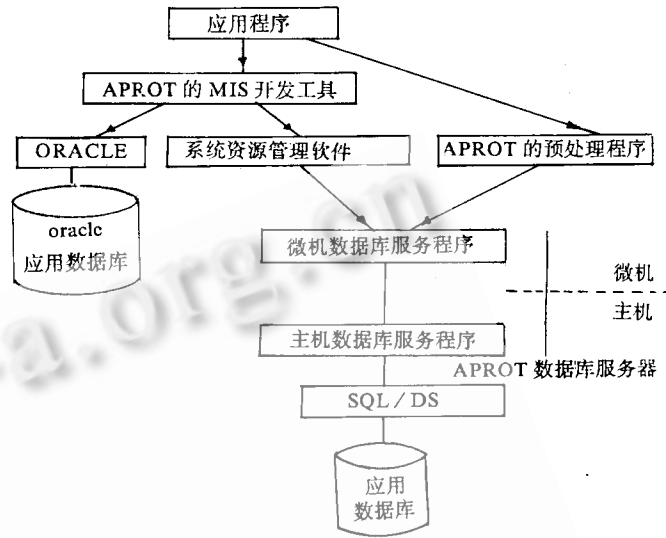


图 1

### 三、智能终端上的应用程序员界面

微机通过 APROT 的数据库服务器构成主机的智能终端,智能终端上的开发环境 APROT 主要面向程序员。根据开发人员对 APROT 的使用情况,程序员界面分为两类:

1.应用程序员直接使用 APROT 的数据库服务程序。APROT 允许应用程序员直接使用数据库服务程序提供的功能,其编程方式是微机 C 语言嵌入 SQL 服务命令,即以 EXEC IBM 为前缀的 SQL 语句(我们称此类程序为 APROT \* C),类似 ORACLE 的 pro \* C 中的 EXEC SQL 语句。从主机获取的数据由程序员直接控制,即可将数据存放在内存变量或数组中,也可存放在微机 ORACLE 数据库的表中。APROT 提供 APROT \* C 的预处理程序,它把含有 SQL 服务命令的 C 语言应用程序转换成 C 程序,使之能访问主机数据库中的资源。预处理程序,把输入文件(APROT \* C)中的 EXEC IBM 语句转换成输出文件中对数据库服务程序适当的调用,输出文件随后就可以以 C 语言程序的正常方式进行编译、链接和执行。

2.应用程序员使用 APROT 的 MIS 开发工具。

大型软件的开发是一项集体劳动。在该领域搞接力式开发是明智的方法。这样,应用程序设计人员不必从零做起。事实上,我们所用的编译程序、操作系统、DBMS 等就是避免应用软件人员从二进制码做起的辅助工具的典范。APROT 各种通用工具在资源管理软件的支持下分别实现若干独立的功能,都有明确的入口条件与出口形式。应用开发人员指定好入口参数后就可以把他所需要的通用工具纳入到其应用程序中,这可减少大量的编程时间,可以迅速构造出一个应用系统原型。

对于以上两类 APROT 的程序员界面,由于 APROT \* C 与微机 ORACLE 的 pro \* c 非常类似,因此程序员可以稍加修改 APROT 程序,即可方便地变成 OPACLE 的 Pro \* C 程序,再通过 ORACLE 的 PRO \* C 预处理程序 PCC 就将在 APROT 上建立的应用系统移植成微机 ORACLE 上的应用系统。这样,一个应用系统可以由联机方式运行方便地变成脱机方式运行,解决了有的最终用户需要同一个应用两个系统同时并存的应用需求,也减轻了开发者的编程工作量。

#### 四、主要技术问题

APROT 是一个新型的具有多机分布处理特点的大型数据开发环境,它的开发技术关键主要有以下几个方面。

1. APROT 的数据库服务程序的主机实现方面。服务程序的主机部分需要接受和完成客户机发送过来的任何 SQL 服务请求,因此,需要有 SQL 的动态处理功能。由于 SQL / DS 具有预编译的语言配置的只有 COBOL.FORTRAN 和汇编(IBM370 汇编语言),而具有完全动态定义执行语句功能(指 SQL 语句在查询条件部分和查询内容部分以致整个语句均可在运行时定义)的只有汇编,那么必须使用 IBM 370 汇编语言并嵌入全动态 SQL 语句和有关 CICS (Customer information control sysytm)语句,以实现主机部分对 SQL / DS 数据库的全动态操作,其中包括汇编语句对 SQL 动态存取区 SQLDA (SQL Dsta Area)的处理。

2. APROT 的数据库服务程序的微机实现方面。首先是通过解剖 GW3270 仿真程序,编制仿真处理模块,

使微机可以通过程序方式(不是通过键盘仿真)完成以下三个功能:(1)直接向主机发送数据(命令);(2)接收从主机传送到 IRMA 仿真板上数据或结果信息;(3)随时读取当前主机的系统状态。

其次 SQL / DS 对全动态语句的支持有一定的限制,它不支持以下几个语句的全动态定义:

INCLUDE	SQLDA	ROUBACK	WORK
INCLUDE	SQLCA	COMMIT	WORK
WHENEVER		CONNECT	
OPEN		PREPARE	
CLOSE		EXECUTE	
FETCH		EXECUTE	IMMEDIATE
DECLARE	CURSOR	DESCRIBE	

嵌入在用 C 语言编制的应用程序中的 SQL 命令语句中可能在该语句的条件部分或结果部分含有 C 语言的变量,作为 SQL 语言的宿主变量(SQL 语句里用(:)作为前缀的变量,在 C 程序中用 DECLARE 段中说明), APROT \* C 的预处理程序首先必须对嵌入的 SQL 命令进行语法检查,之后将它转换成主机 SQL / DS 可以识别并可动态执行的 SQL 语句(不含微机宿主变量的真正 SQL 语句字符串),其中还需用一定的内部格式来表述上面所说的动态功能 SQL / DS 不支持的 SQL 语句。数据库服务程序还必须自动地将主机获取的数据送入 C 语言的变量(宿主变量)中,其中包括数据的类型转换。这些技术难点体现在 APROT \* C 的预处理程序和微机数据库服务程序的实现上。

3. 由于主机和微机服务程序之间的通讯对 APROT 的用户(程序员)是透明,因此需要建立一套主机和微机通讯的内部协议,以定义微机向主机发送命令以及主机向微机发送数据和 SQLCA (SQL Communicatoin Area)内容的格式。由于微机从主机的记录形式获取的数据其长度、类型是随 SELECT 语句内容不断变化的,因此协议确定的格式应满足这种变化,并能使 SQL / DS 的数据类型与 C 语言的数据类型相对应。同时在智能终端上必须构造微机上的 SQL 动态存取区 SQLDA,来仿真主机的 SQL 全动态定义执行功能,以支持通用 MIS 开发工具的实现和应用程序的特殊需求。

(下转第 43 页)

(上接第4页)

4. APROT 的 MIS 开发工具部分。这些工具与应用系统之间的关系较为密切,如何开发出适应性强、通用程度好的菜单模块、查询处理模块、联机数据录入模块、报表打印模块,以满足应用系统的要求,这是本系统开发的重要方面。开发的 APROT 工具的关键问题之一就是搞好系统资源管理。在此,系统资源包括 IBM4300 系列机的 SQL/DS 上建立的共享数据库资源以及本地的 ORACLE 数据库资源。每个客户上所见的系统资源视图一般是不同的,不同的视图产生不同的应用,而 APROT 工具不须更改。

## 参考文献

- [1]. J.T. Johnson, "Managing The PC to Mainframe Link", 《Personal Computing》 Aug. 1990.
- [2]. S. Fisher, "Making the Micro- to- Mainframe Connection" 《BYTE》, Sep. 1990.
- [3]. 《SQL. Date System Application Programming for VSE》, IBM Refence SH24-5018-3, 1984.
- [4]. "IBM 4300 计算机与微型计算机通用工具技术报告" 中国华资总公司计算中心 1990.10.

