1982年全美计算机会议论文集

中国科学院成都计算机应用研究所情报室译

AFIPS

1982年全美计算机会议论文集

（下卷）

中国科学院成都计算机应用研究所情报室译

数据服务系统设计问题

FRED MARYANSKI

摘要

随着局部地区网络的急速发展，便产生了对网络数据库服务系统的要求。把局部地区网络数据服务系统看成是后端数据库的扩充是合适的。本文主要介绍数据服务系统设计中的几个关键问题：功能分配、高有效性、高度安全性和高性能，特别提到了把后端数据库的经验应用到数据服务系统上来，提出了几种设计方案，并就它们的可靠性，安全性以及在一般意义上的特征作了评价。最后部分着重强调了局部地区网络需要更大的实际经验，以便更精确地对各种不同的数据服务系统结构作出估计。

引　言

七十年代提出了后端数据库系统的概念，作为提高第三代大型机数据处理能力的效能价格合算的一种潜在方法。虽然在后端数据库方面继续有些活动，但在极大程度上，由于通讯的成本高，后端数据库还没有在大量的成功的商业系统中实行。到了八十年代，随着期待的局部地区网络在商业市场上的出现，似乎后端数据库技术找到了一个更适合它应用的基础。严格说来，未来的局部地区网络将不包括传统意义上的后端机。图1画的是一个后端数据库系统图，在这个后端数据库系统中，数据库和数据管理系统被御到专用处理机上，专用处理机直接接到大型主机，与后端数据库等价的局部地区网络是数据服务系统，数据服务系统为网络提供数据管理设施。本文研究局部地区网络数据服务系统的设计。着重强调把后端数据库系统中学到的东西应用到这一新的环境。

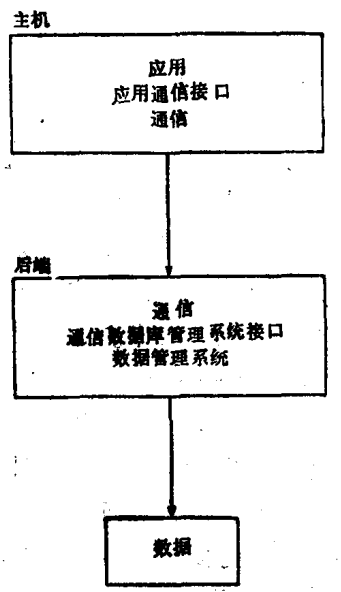

图1　后端数据库系统

第二部分提供了有关后端数据库系统的一些参考资料，并提到一些与网络数据服务系统有关的工作。第三部分规定了局部网络环境，而文中的数据服务系统正是为这种环境设计的。第四部分介绍数据服务系统设计的主要问题。最后就该领域目前所面监的主要问题和对今后的工作提出建议。

基本情况

一九七四年Canaday和他的助手发表了第一篇有关专用数据库处理机的文章，若干从事通用计算机到专用数据库计算机发展过程的研究人员，详细地对后端计算机的功能进行了考查，[4,13,22]因此能够得出对后端数据库系统的一分综述。

从数据资源管理的观点来看，把处理机专用于数据管理操作，引来了强烈的争论[20]。直到最近，宽带通信的高成本还极大的限制着这种配置的功能。随着局部地区网络的出现，把一台或多台处理机指定为数据服务系统似乎是行得通的。现在已经作成了几个微处理机数据服务系统的雏型。

- PHLOX 工程网络数据服务系统，该系统就是把数据管理程序和操作系统集成在一台专制的微处理机上形成的系统。
- MINIMET[14] 和 MICRONET[25] 数据服务系统是由几台微处理机组成的系统，各系统都属数据库的一部分，每个数据库要求广播到全部微处理机，每台微处理机独立地响应主机，由主机软件汇总全部结果。
- UNITY系统由一台微处理机“数据服务系统”组成，这个微处理机“数据服务系统”的数据从小型计算机数据库脱机装入。微处理机服务系统然后以单独的方式操作。小型计算机数据库从微处理机服务系统那里得出周期性的更新。UNITY系统的基本设计决定不向主机提供连续同步。

本文讨论的数据服务系统设计中的问题是针对局部地区网络环境的，这种局部地区网络环境同以前研究的环境稍有差别。下节介绍网络结构。

局部地区网络结构

在研究数据服务程序设计问题以前，必须先确定系统运行的环境。图2给出了数据服务系统所定义的普通局部地区网络。随后讨论，假设使通信使用具有一定距离限制的高带宽广播法。网与环的争论，不打算提出来讨论。

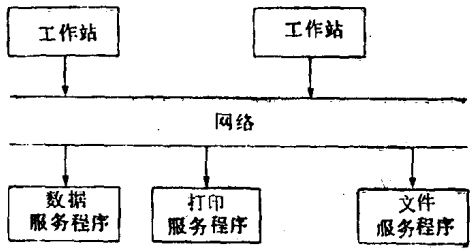

图2　局部地区网络

在网络中，有两种类型的处理机工作站和数据服务系统，工作站直接援助用户终端。一个工作站内含足够的能力支持用户的业务工作。服务系统给所有用户提供对共享设备的利用，比如高速度、高质量打印机、一个文件包和一个公共数据库。按照数据库管理系统在单机环境里与文件系统从功能上相区别的方式，来区别数据服务系统和文件服务系统。已经在某种应用环境中得到开发的，这种性质的网络的描述已由 Kaisler 和Lind[11,12]给出。

局部地区网络的一条基本原则是高通信带宽和不断下降处理机的成本决定了处理机的功能技术条件。对这样的专用处理机可以进行调整，或者专门生产，以便以最佳方式提供所分配的功能。这个工作的焦点是专门用作数据管理功能的处理机或处理机组而设计的。

数据服务系统设计研究

为了建立有效可靠的数据服务系统，必须认真地研究几个重要问题。本文打算着重

讨论这些问题中最紧迫的问题，如下：

数据管理功能的分布

有效性

安全性

性能

功能分布

同后端系统的情况一样，简单地把数据管理软件放置在一台专用处理机里是不能够建立一个数据服务系统的。数据管理软件必须在工作站和数据服务系统之间分配。正如与一个传统的后端系统一样，为了把数据管理系统的各块连在一起，必须在数据服务系统结构上增加辅助通信软件。后端经验的一个教训就是一次记录DML语句，按照CODASY的方式，不是处理机间通信的真单元[16,17,21]。即使在一个具有高通信带宽的局部地区网络中，要使用较高级的数据存取语言。对数据服务系统环境来讲，关系数据操纵语言如象SEQUEL或QUEL语言就是很适合的语言。这一类语言，一个有吸引力的特点就是，一个简单的请求就能够产生大量数据的传送。只要能用高级数据操纵语言，基本上可以不用数据管理程序。Cermano举了一个用于CODASYL数据管理程序[9]的高级操纵语言的实例。为了讨论起见，所设计的数据服务系统是支持关系模型的。

局部地区网络数据服务系统必须支持自发的询问，也要支持已存事务的执行。数据管理功能分布的讨论将用到R系统的结构，已经有好几篇文章描写R系统结构[1,5,7]。正如Chamberlain所说，PLI或COBOL应用程序可以存取一个R系统数据库，用程序里包含的，$LET，$OPEN，$FETCH和CLOSE语句来实现这种存取。这些语句的功能如下：

$LET语句：把SEQUEL询问定义为一个事务处理和命名在询问中应用的程序变量。这个语句经R系统预编译程序处理，为这种事务处理产生一个存取模块。

- $OPEN语句：汇集变量。
- $FETCH语句：为了检索操作，这个功能使数据从被选中的元组中写进程序变量。
- $CLOSE语句：自由变量。

图3描述的是在一个单处理器环境里，R系统文件的预编译。在前一节介绍的局部地区网络中，编译功能是驻留在工作站的。这样，数据管理预编译程序的付本要能送到每个工作站，在考虑事务处理运行时间环境的结构时，要延迟预编译程序产生的存取模块位置问题，R系统事务处理运行时间环境

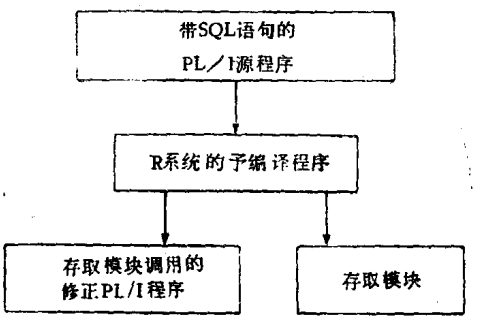

图3　R系统预编译程序步骤

如图4所示，划分运行时间环境最直接的方法是把用户目标程序分配给工作站和把剩余模块放在数据服务系统里。然而，如果严格检查存取模块和目标程序之间的相互作用，自然就可以查出前面提到的划分策略的某些问题。在预编译阶段刚才谈到的R系统的每一

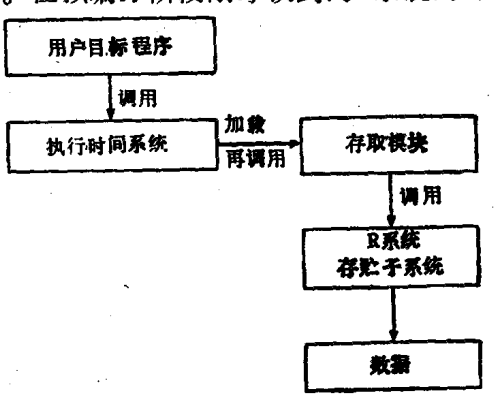

图4　R系统事务处理执行步骤

个算子，被调用程序调入适当的存取模块。如果程序发出一个在几个元组中检索的请求，就执行 $FETCH语句，为每个元组检索数据。下面是抽样事务处理范例：

```
$LET CI BE
  SELECT NAME, SALARY INTO
   $X, $Y
  FROM EMP WHERE JOB = $Z;
DO JOB INDEX = 1 TO NUMJOBS;
  $Z = JOB TABLE ( JOB INDEX );
  $OPEN CI
DO WHILE ( SYR CODE = DONE );
  $FETCH CI;
  . . .
  END
  $CLOSE CI
EDN
```

在一个局部地区网络里，如果整个执行时间系统驻留在数据服务系统上，那么在每次执行 $FETCH语句时，都必须在工作站和数据服务系统之间进行信息交换。实际上，为数据服务系统提供信息的基础，就象在用CODASYL DML一样。使用后端数据库的经验告诉我们要避免这种情况的存取。

在检索操作中，以全部所选元组从数据服务系统同时移向工作站局部存储器这样一种方式来划分执行时间系统和存取模块，然后从工作站存储器选取各个元组以响应 $FETCH语句，这就是划分执行时间系统和存取模块的解法。与每个R系统语句和它们在网络中的位置有关联的动作，总述如下：

- $LET语句；在预编译阶段处理。结果装入模块被存储在工作站和数据服务系统内。
- $OPEN语句；用装入模块在工作站产生选择的表达式变量汇集，然后数据库请求以调用装入模块的方式传送到数据服务系统，在数据服务系统里完成这一请求，得出结果，以状态结果的形式或以数据元组的形式又传送回工作站，元组缓存在工作站存储器内。

  $FETCH语句，元组从工作站存储器和与程序变量相关联的数据中检索出，在工作站本地完成。
- $CLOSE语句，释放保持在被选元组的缓冲器内，这项工作也是在工作站本地完成的。

上面讨论说明，数据库事务处理的根本部分是工作站的责任。图5表示，在工作站和服务系统之间，为了进行事务处理而划分数据处理模块。

R系统中的询问处理除了预编译步骤外，类似于数据库事务处理，R系统询问处理程序如图6所述[7]。两种特殊的询问操作是：

- PREPARE语句，这个语句的操作数是一个询问文本和一组参数，这两者都是用“？”为标记，这个文本由预编译程序处理以产生一个存取模块。
- EXECUTE语句，这个语句是用PREPARE语句已经处理了的一个询问的名字和正式参数的名字来调用的。形式参数被限制在一个由 PREPARE 语句产生的存取模块上。这些语句由 SQL 询问处理程序使用为自然数据库请求

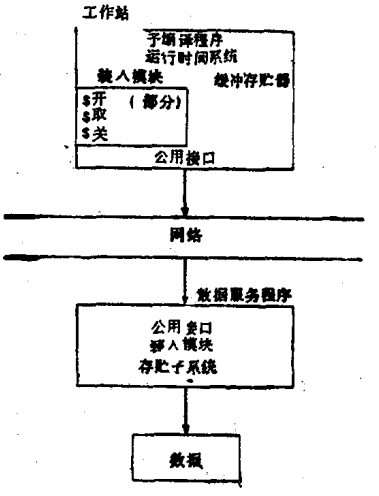

图5 数据管理程序功能分布图

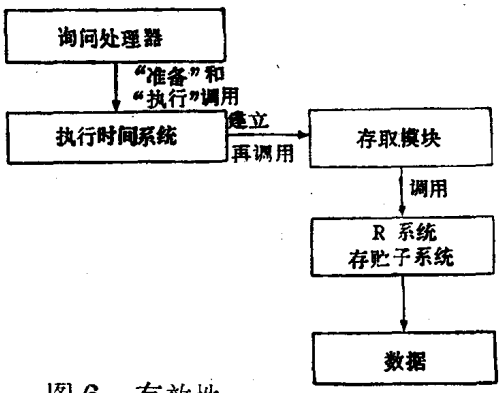

图6　有效性

而建立装入模块。这些模块由询问处理程序用 $OPEN语句、$FETCH语句和 $CLOSE语句调用，其方式同在已存事务处理的情况下所使用的方法一样。一旦询问已经编译，SQL 询问处理程序以应用程序的方式存取数据库。因为，询问处理程序和事务处理程序只是预编译的步骤不同，所以不需要改变上述数据服务系统的结构就可以支援询问。

有效性

在数据服务系统和局部地区的网络讨论中，有这样一种很普遍的论调，比如："如果我要把我的数据放在数据服务系统里，那它应该是万无一失的，"因此有人禁不住要打听，说话者好象现在已经驻留了数据于通用机。所以，接受这样一种论点，即数据服务系统必须具备高可靠性和有效性是很自然的。就高有效性数据服务系统而论，最重要的设计抉择不得不考虑冗余技术。很明显，采用多处理机，能够获得额外的可靠性，任何功能比较强的数据库，都保存了数据备用的副本。那么就数据服务系统来讲，究竟备用副本是作为当前数据驻留在同一个处理机上呢？还是置于另一处理机上，这是个问题。局部地区网络的基本观点是赞成多处理机，当然增加处理机就要增加成本，不过可以提高可靠性，也就等于降低了处理机的价格，在高带宽通信的潜力中得到补偿。

如果决定采用多处理机方案，那么，下个问题就是在多处理机数据服务系统上组织数据和数据管理系统，以便得到较高利用效率。为了便于把数据控制分布在一组数据服务系统上，有两种基本的方案。在一个高度有效的环境里，必须有数据的多个复本，那就是分布数据的控制问题。

1．一个服务系统包含数据的原始文本，而其它节点保存后备文本，最初的全部请求都是指向主数据服务系统。

2．数据控制分散到各数据服务系统。每个服务系统对于数据的某些部分承担主要的责任，对数据库的其它部分只承担备用的责任。

这两个通用方法的根本区别，就在于第二种方法要求分布式并行控制算法，分布式并行控制已被证明是计算机科学中的一个扑朔迷离的问题。尽管在理论上已经提出了若干结论，但这些算法得到的一个效果还是未知数。因此，为了简便，或许也是为了性能的原因，这里给出的是主数据服务系统方案，在一个要求大量数据库服务，而这些服务又涉及若干数据服务系统结点的局部地区网络中，可以采取某种综合的办法。

这里所提出来的主数据服务系统方案是一个向单机环境发展体系结构的方案。图7里表示的是由Maryanski[18]定义的容错数据库体系结构。这一个面向可靠性的以微分文件的概念和仔细替换为基础的方法。简言之，微分文件方法涉及把对数据库的全部修改保存在一个分离文件中，这个微分文件，周期性地与主数据库合并，主数据库里还保存一个副本，利用筛选程序确定受读操作请求的记录是否在微分文件中出现。最后结果经修改后直接送主数据库，以后就没有修改数据读的请求了。产生一次结构修改以前，就微分文本而论，在时间上请求处理的比例

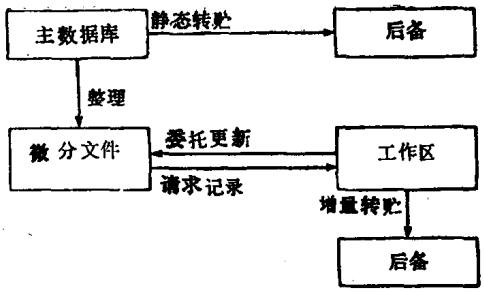

图7　容错数据库体系结构

不利于主数据库。Maryanshi[18]给出了一个在线结构修改的算法。

仔细地替换技术避免“在位更新”。当产生一次更新时，当前值就被复制进入工作区，只在更新的地方产生修改。新值写进来后，随即调整指示字。只是在把更新记录到微分文本里时，才使用仔细替换策略。微分文本的利用，为第一次更新记录提供了有效地仔细替换。

最初假定两个处理机专门用于提供数据库服务，采用在两个处理机之间分配图7中的元件这样一种战略的目的，在保持容错的情况下获得最大性能。

一种分配方案（见图8），把微分文件放在主处理机上，主数据驻留在后备或从属处理机上。在此方案中，所有的更新请求都由主处理机处理。如果读操作涉及微分文件的数据，这个操作就由主处理机来完成。为了不改变数据，只好把读请求送到后备处理机。因为主节点要处理所有的更新，所以，它也承担并行控制的责任。工作分散于两个处理机上是由于更新频率而引起的。最初，主处理机只处理更新，而所有的读请求都被送往从属处理机。实际上，在这种结构中，不在微分文件里完成一次数据的更新，而要求主处理机从后备处理机中获得数据。

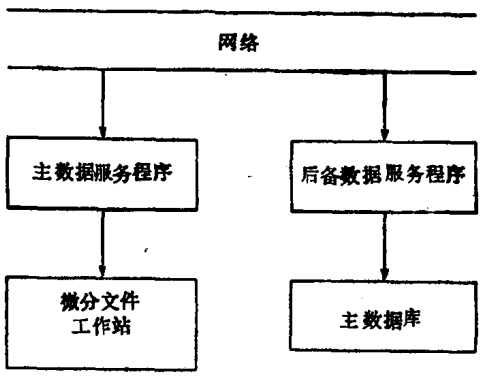

图8　冗余数据服务程序，配置1

在这种结构里的另外一个问题是，处理机的利用完全靠数据驱动。处理机之间的这种分工可能是不平衡的，这是由读请求从微分文件里存取数据的频率所致。起初，后备处理机处理全部读请求。当微分文件的规模增加时，由主处理机承担这种活动的比例越来越大。

图9所介绍的结构在更新不在微分文件里的数据时，用不考虑处理机间内部通信的办法，可以缓和前面提到的那种结构出现的第一个问题。因为主数据库的一个复制本保存在主处理机里，也保存在后备处理机，所以从主数据库到微分文件的转换来得更快。在两个节点上都保存主数据库的复制本，为平衡数据库处理机的利用率提供了一个机会。当微分文件规模比较小的时候，这样改变，策略上会产生作用。根据80—20法则，[15]，绝大多数数据库活动都集中在相对小的一部分数据。当数据库达到一个稳定规模的时候，微分文件就会包含数据库最频繁存取的部分。那末，主处理机将处理大多数数据库请求。然而，由于微分文件是整个数据库

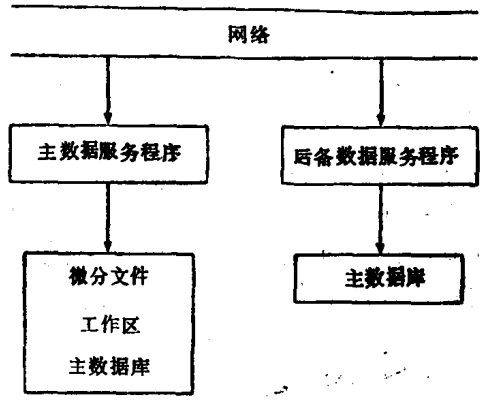

图9　冗余数据服务，程序2

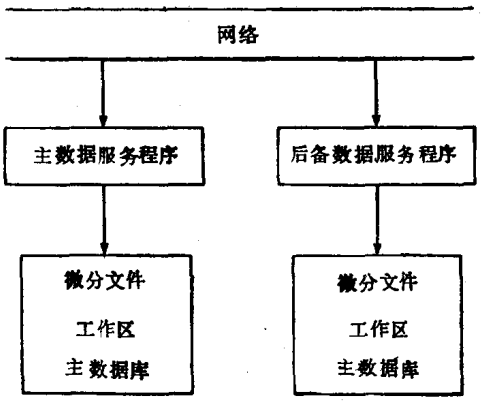

图10 冗余数据服务程序，配置3

的一个小的子集，所以搜索微分文件要比搜索主数据库快些。

把构成高可靠性数据服务系统结构元件进行分配的最后一种可能性涉及把微分文件和主数据库复制在两个处理机上的结构。这种结构如图10所示。克服了图8中的数据服务系统体系结构的两种毛病，另外比图9提供的结构提供了更为一致地利用处理机的能力。遗憾的是，这最后的结构方式又引起了保存一致的微分文件复制本的问题。要保持微分文件的一致性，可能的方法有：

- 在处理机上完成全部更新，然后把所作的改变广播到后备处理机。这种方法限制了负载的均衡，只有检索才可用任意一台处理机。然而只有并行控制保留为仅是主处理机才有的功能。
- 在数据服务系统处理机中使用负载均衡的结构方式分配更新，实际上，主处理机和从属处理机的主从关系就不再存在了。虽然这种方法能够在处理机中公平分布工作负荷，但引起了分布式并行控制的问题。结果所得到的结构与分布式数据管理系统完全相似。

在这些可供选择的分配方法中，第一种方法比较简单，容易实现。控制软件需要的存储器和时间都比较少。因为数据服务系统专用于一种单一的功能，似乎效率不是关键的问题，然而对这种资源的要求高，因此数据服务系统必须尽可能有效地执行。在这一小节里给出的多处理器数据服务系统结构，以及Maryanski[18]，Severance和Lohman[23]还有Verhofstad[27]他们提供的算法，一起保证了数据的高度有效性，正如Maryanski[18]以及Severance和Lohman[23]所说，以微分文件概念为基础的系统的性能极大地依赖于更新的频率和所出现的场所。这里所谈的系统结构，尽一切努力提供检索和更新命令的迅速执行。

安全性

安全问题在局部地区网络的数据服务系统里和在后端数据库系统中是相类似的。基本问题是："网络数据服务系统是不是要比单个多用途处理机上的数据管理系统提供更多的安全性？"怎样来评价这个问题，涉及两个重要的因素

- 隔离
- 特许

隔离

要求后端数据库系统提供更强的安全性之根本原因在于后端数据库系统是专门用来完成数据管理功能的。错误百出的应用绝不会越过数据管理系统而秘密地存取数据库文件，在许多建立标准文件系统上的数据管理系统中，直接通过操作系统而不通知数据管理系统就可以直接存取数据文件。后端数据库和数据服务系统这两种结构，采用从所有应用程序中隔离数据的办法，克服了这个问题。这样，伪用户要想以一种未经许可的方法存取数据库必须经过伪装。这样的观察，得出下面的结论，假如特许的问题已解决的话，由数据服务系统提供的数据隔离只是对提高安全性有用。

特许

数据库系统总是依靠特许法来提供最基本的安全。在数据管理系统和应用不是同时

驻留在一台处理机的局部地区网络中，它的依附性更大。在R系统中，特许信息作为关系保存在Griffiths和Wade图形中。指定的用户，采用发出对那些关系起作用的特许命令的方法，便可允许和取消对数据库某些部分的存取，INGRES使用了一个类似的特许办法[24]，稍微改动一下，这种类型的特许方法就能在数据服务系统环境里应用。包含有特许信息的关系驻留在数据服务系统上，允许和取消特许权的命令从工作站发散到数据服务系统。

上面的描述假设，基本的网络控制结构提供了一个安全的用户鉴定方法。局部地区网络是增加特许结构的有效性还是减少特许结构的有效性，这是一个可自由讨论的问题。显然，数据服务系统紧紧地依赖于这个网络的特许方法。如果网络的安全法确实能把侵入者从真正的用户中区别出来，特许的标准关系法将适用于数据服务系统。

性能

数据库的性能当然要取决于象数据库结构和使用形式这样一些难于描述的变量。数据库服务系统能评价的两个基本问题是：

- 怎样把局部地区网络中数据服务系统的性能和运行数据管理系统与应用程序的单一大型机的性能进行比较？
- 在一个特定的环境中，哪种数据服务系统结构给出最好的性能？

回答第一个问题特别困难，因为在通用大型机上，数据管理系统的性能不仅受大型机非数据库工作负载的影响，而且受数据库请求的影响。综合磁盘的利用对数据库管理系统操作有很强的影响。在一个带有专用服务系统的局部地区网络上，数据管理系统不必要同别的子系统竞争磁盘资源。然而，局部地区网络的处理机间通信将会引起传输延迟和需要额外通信软件等不良后果。决定单一大型机系统与专用处理机的局部地区网络的性能调整，需要进行仔细的分析。对于一个特定的应用环境，仿真才是判断这些调整的最好方法。

图8到图10中的数据服务系统结构相对性能比较，可用排队模型分析法。这些模型必须参数化以描述特定的环境，使用的关键因素为请求在数据库内的分布更新频率，通信延迟，包括线路和软件延迟和数据服务系统间的延迟。这里也要用到仿真技术来反映在各种环境中各种类型的结构的性能。在数据服务系统性能分析中还有一个有趣的变量，那就是一个数据服务系统结构中处理机的最佳数。这里还要说明并行操作和附加通信之间的调整必须保持平衡。在处理机数量可变的分析中，价格就成为一个比较重要的因素。

后端数据库系统的经验告诉我们，可以期望用仿真或分析研究得到的性能反映，与实际样机的特性还存在差别。理论和实际性能的差别可归咎于工作负载的不准确建模和对完全通信软件的时间估计不足。实验者实难用几个应用并行存取数据库的办法来产生大的工作负载。除非充分利用数据库，否则就不可能实现后端和主机并行操作的好处。较高速度的局部地区网络连接，当然要比后端系统使用标准点到通信产生的特性好。然而，使用高带宽的通信介质，并不自动意味着在工作站中或数据服务系统中有更为有效的通信软件。如果这些局部地区网络要想满足他们期待的功能，那么，许多后端数据库系统原形中出现的过分冗余的通信软件，必定要在局部地区网络的数据服务系统中减少。

结　论

这里谈到的数据服务系统设计中出现的问题，只是对问题进行初步研究的结果。正如前面提到的一样，后端数据库系统的经验不仅强烈地影响到问题的定义，而且影响到许多提出的结论。因为局部地区网络的经验

有限，很难准确地建立用户工作负荷特征的模型，因而也就影响到建立数据服务系统合理的特性模型，目前数据服务系统设计者面临的主要问题是

- 恰当地定义高级通信规范，这种规范允许应用和数据服务系统之间高速度进行实际数据转换。
- 综合精确特性模型，以对不同体系结构进行评价。
- 对在特定环境里提出的高性能，安全性，可靠性要求的理解。

为了获得对一个实验系统性能的真正理解，作原型机仍不失为最好的方法，这对局部地区网络数据库服务系统当然是有效的。元件技术没问题，关键任务是要把通信、数据管理和应用系统组装在一个精心集成的网络数据服务系统上。

参考文献

1 ,Astrahan,M.M.,et al.,"System R:Relational Approach to Database Management,"ACM TODS,1,2 June 1976, PP.97-137.

2 .Bernstein,P.A.,et al."The Concurrency Contol Mechanism of SDD-1: A System for Distributed Database (The Fully Redundant Case),"IEEE Transaction on Software Engineering,SE-4 (1978),PP.154-169.

3 .Bernstein,P.A.,and N.Goodman,"Fundamental Algorithms for Concurrency Control in Distributd Database System," CCA-80-5,Computer Corp.of America, Cambridge,Mass.(1980).

4 .Berra,P.B.,"Data Base Machines,"ACM SIGIR Newsletter(Winter 1977), PP.4-23.

5 .Blansgen,M.W.,et al.,"System R:An Architectural Update,"IBM Research Report,San Jose,Calif.(1979).

6 .Canaday,R.E., et al., "A Back-End Computer for Data Base Management," CACM,17,PP.575-582.

7 .Chamberlain,D.D.,et al.,"Support for Repetitive Transactions and Ad Hoc Queries in System R," ACM TODS, 6 (1981),PP.70-94.

8 .Delvechio,B., and P Penny, "The P-HLOX Project:Three Database Management Systems for Microcomputers,"ACM SIGSMALL-SIGPC Symposium (1980),PP,173-178.

9 .Germano,FJr.,"DSEED:A. Distributed CODASYL Prototype System,"Ph. D.Dissertation,Whartion School,University of Pennsylvania (1980).

10.Griffiths,P.P.,and B.W.Wade, "An Authorization Mechanism for a Relational Database System," ACM TODS, 1 (1976),PP.242-255.

11.Kaisler,S.,"The Agency Personal Information System,"ACM SIGSMAII-SIGPC Symposium(1980),PP.114-125.

12.Lind,L.,"An Actual Implementation of a Distributd Database on a Minicomputer,"in State of the Art Report on Distributed Databases, INFOTECH (1979),PP.187-202.

13.Lowenthal,E,I.,"A Survey—The Application of Data Base Management Computers in Distributed Systems,"VLDB October 1977,PP.85-92.

14.Maekawa,M.,and S.Ishii,"An Extensible Distrbuted Data Base System,"AFIPS Proceedings of the National Computer Conference,47(1978)PP.831-822.

15.March,S.T.,and D.G.Severance,"The

Determination of Efficient Record Segments and Blocking Factors for Shared Data," ACM TODS,2 (1977)., PP. 279-296.

16. Maryanski, F.J.,et al.,"A Prototype Distributed DBMS, "Hawaii International Conference on Systems Science, Voj.2 (1979),PP.205-214.

17. Maryanski,F.J.,"Backend Database Systems," Comouting Surveys 12 (1) 1978, PP.3—7

18. Maryauski,F.J., and P.Charoenpong, "Au Architecture for Fault Tolerance in Database Systems, "ACM Annual Conference,October 1980,PP.389-398.

19. Metcalfe,R.M.,and D.R.Boggs,"Ethernet: Distributd Packet Switching for Local Computer Networks,"CACM,19 (1976),PP.395-494.

20. Nolan, R.L., "Restructuring the Data Processing Organization for Data Resource Management,"IFIP Information Processing 77 (1977),PP.261-265.

21. Passafiume,J.J.,and J.Rivan, "Providing Network Data Services Using a Backend Data Base Machine," IEEE COMPCON,February 1980,PP.251-262

22. Rosenthal,R.S.,"The Data Management Machine,A Classification," Workshop on Computer Architecture for Non-Numeric Processing,May 1977, PP.35-39

23. Severance, D.G.,and G.M. Lohman, Differential Files:There Application to the Maintenance of Large Dasabases," ACM TODS,1 (1976), PP. 256-267.

24. Stonebraker,M.,et al.,"The Design and Implementation of INGRES,"ACM T-ODS,1 (1976)bPP.189-222.

25. Su,S.Y.W.,et al.e"A Microcomputer Network System for Managing Distribcted Relational Datatabases,"VLDB,September 1978,PP.288-298.

26. Ting,P.D.,and D.C.Tsichritzis,"A Micor-DBMS for a Distributed Data Base,"VLDB,September 1978,PP. 200-206.

27. Vercovery and Crash Resistance in a Filling System,"ACM SIGMOD Conference,August 1977,PP.158-167.

28. Verhofstad,J.S.M.,"Recovery Techniques for Database Systems," Computing Surveysv10 (1978),PP.167-195.

29. Wilkes,M.V.,and D.J.Wheeler,"Thc Cambridgc Digital Communication Ring" Loaal Area Communications Network Symposium,May 1979. 1617

秦学文译

徐宏宇校

社会的和组织的推论

计算机安全性验收标准

WILLIAM NEUGENT

摘 要

验收标准决定要求的质量等级和评价质量等级时必须进行检查的鉴定范围。本文提出计算机安全验收标准的三个范畴：功能性，性能，研制方法。每个范畴又分为小范畴。文中提出了将要求和标准列出公式的辅助手段，包括组织策略的使用和危险分析的方法。定量被认为是非永久性存贮的工具，因为数字常常被当作是单独的数据点，而不是区域。文中提出一组制定验收标准时应当遵循的原则。这些原则如下：（1）良好的开始，（2）确信每个人都能理解，（3）区别“将要”和“应当”，（4）解释为什么。讨论了验收确定过程，关键问题是中间产品必须得到批准。验收标准的意义在于使产品更好和更容易判断。

引 论

没有人比那些首次面对“按照他们的要求”建立起系统的计算机用户更加感到惊异的了。有些人承认他们的感觉就象那些不期而遇的人一样：辨认不能接受的东西比确定可接受的东西要容易得多。

这个问题在计算机安全领域内特别普遍。产生问题的一个原因是很少知道验收标准所起的作用。人们把要求确定认为是唯一确定他们需要什么能力的过程。他们忘记了要求确定还必须考虑如何确定产品可否验收。这种决定可否验收的标准就叫做验收标准。

本文提出一种验收标准分类法，以及使验收标准比较容易确定的一组原则。目的是帮助人们确定的计算机安全要求既可以改进最终产品，又可以简化确定产品可否验收的办法。文中只涉及到软件和硬件的研究，不过对其它方面也可以适用。

验收标准

验收标准是专业性的安全要求。它们是专业性的，因为它们代表着不同于其它安全要求的远景。而正常要求的典型公式是回答问题：“我们需要什么？”验收标准回答的问题是“我们如何决定产品可否验收？”这些显然是重叠的规定，因为如果产品满足要求，通常都规定产品是可以验收的。问题是如果只问第一个问题，要求并不是常常都充分确定的，验收标准的作用是保证要求包括下列两个问题的充分定义，即（1）“要求什么质量等级？”和（2）“在评定质量等级时要检查什么”？

因此验收标准是所需安全功能的可计量和可验证的特点，而安全功能表征它们所要求的质量。它们在确定产品是否遵守安全要求时用作判定的标准。他们还可以指导那些

必须决定产品质量如何的研制人员。

在研制工作的所有各级都要对质量作出判定。这是因为每个设计级都作为以下各级的一组要求（图1）。每一级都告诉下面的级必须干什么，并说明如何实现以上的级。

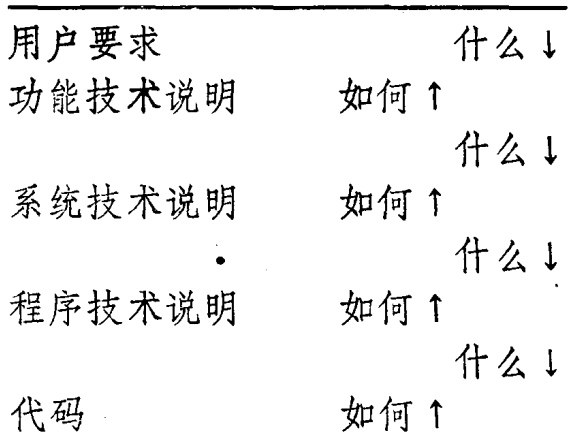

图1　设计间的关系

告诉要求什么的过程就是要求确定过程，并包含着验收标准的某种形式的需要。例如在功能技术说明级需要标准来帮助那些确定系统技术说明的人决定要嵌入多少冗余；而在系统技术说明级则需要标准来帮计程序检查人决定误差检验和处理的范围。

本文讨论用户要求级的验收标准。由于这个级涉及到求解的主要问题的定义，所以这一级的标准最为重要。这些标准驻留在用户文件中。在许多较低级文件中出现更多特殊形式的标准，最前面的是功能技术说明和验收测试过程。

以上对验收标准作出一般的介绍，以下将更为详细地对它们进行研究。

验收标准的结构

确定验收标准的任务是在说明控制质量时用一只眼睛看着评价。为了作好这个工作，必须首先决定那一个控制特征是控制质量的主要决定因素。现在提出三种决定因素：

1．功能性。需要的是什么控制功能？

2．性能。控制功能必须完成什么性能？

3．研究方法。系统和控制必须如何研究？

以下逐个进行讨论。

功性能

这个范畴包括那些经常被想作是安全要求的东西：控制功能和数据和灵敏性要求。

控制功能

这些不仅包括象鉴别功能和核准功能那样的控制本身，而且包括和监督它们所需要的功能。管理包括象改变鉴定或核准表那样的功能。这就必须考虑象谁能改变表格和是否改变能动态地进行一类问题。监督包括象误差和文件存取之类安全事件的记录。监督功能的定义必须包括系统需要什么能力来计量它本身的性能。例子是资源使用和响应时间的计量。操作系统必须能够报告它的质量的计量。监督也包括可以检查功能在内。

在确定控制功能时，有几种探索性的辅助手段可以用来帮助保证完整性。这些是类似于解说记录的是谁，什么，为什么，什么时候和什么地方。

1．控制目的：防止，发现，或校正安全暴露。

2．由控制阻止违法行动：揭发，改进，拒绝服务，破坏。

3．控制功能：核准（存取控制），鉴定（识别），监督。有时扩大到包括流控制，推论控制和编密码。

控制功能的定义能够包括以下因素，如象何时，如何经常地，和功能要使用多久（生命的长短）；它执行的细节级；操作条件和约束；和功能中的关系。对安全来说，需要提到的特别重要的附加因素是用户中分享的数据数量和用户功能能力的范围[1]。还必须确定用户验收和容错的需要。用户验收需要包括引导在什么用户中将得到可以验收

的，因而他们不回避或推翻的控制。容错需要阐明用户是如何熟练或经受良好的训练，和系统如何容忍他们犯的错误。

数据和灵敏性要求

数据要求确定系统输出，输入，数据元和数据结构，还有估计的最大容量和平均容量及期望的增长。灵敏性要求包括数据，软件，硬件和人员安置的灵敏性范畴。这些范畴必须明确各种不同灵敏性范畴的保护要求。必须注意到数据的处理或归并是否改变它们的灵敏性。

性　能

控制质量要比正常的功能操作多得多。本文列出在总标题下面的一些质量验收标准。它们可以用于单独的控制或系统。

利用率

要规定出多大一部分时间内，系统必须用以执行重要的或全部的服务。利用率具体体现了可靠性，冗余性和维护能力的许多方面。它经常比精度更重要。有些如象电源系统的系统需要超过99%的利用率。另外一些系统，如象化工过程控制系统和电话网络换接系统的利用率也差不多一样高。安全控制通常要求比其它部分的系统有更高的利用率

生存能力

确定系统必须经受重大故障和自然灾害的能力有多大，在这里经受包括在发生故障时采取紧急操作的支援，以后的后援操作和返回到正常操作的复原作用。重大故障是那些比与利用率有关的较小的瞬间故障更为严重的故障。在故障不能修复时，就象在空间系统和心脏起博机一样，生产能力和利用率是一回事。

精度

确定控制必须精确到什么程度。精度包括误差的数目、频率和重要性。控制什么样的精度计量特别适用于等同检验技术（即采用特征信号、声音）和通信路误差处理技术。在这里可以采用软件质量尺度的研究[2]。

穿透阻力

确定对打破或绕开控制所需要的阻力，这里的阻力是系统和控制必须阻塞或推迟冲击的程度。密码分析就是打破控制的技术的例子（编密码）。建立和使用虚假系统记入效用来发现通行字是绕开控制的例子。确定谁可能是穿透者很重要，可能是：用户，操作人，应用程序员，系统程序员，经理或外部人员。要记住最大的损失来自执行他们的批准任务的人。

响应时间

确定可以接受的响应时间。控制响应时间慢可能诱使用户绕开控制。响应时间极关重要的控制例子是通行字（尤其是分布式系统中）和等同检验技术。响应时间可能对控制管理也极为重要，如象在安全表的动态修改中那样。在确定响应要求时，注意到由于降级而使级改变的影响是有用处的。

解题能力

确定必须支予支援的容量。在这里容量包括象用户和服务请求那种东西的最大和平均负荷。这可能涉及到性能比的使用，如象全部用户与响应时间之比。

费用

确定为了操作和维护控制所能接受的费用。建立控制的费用也是重要的，并包括低于研制方法的标准以下。采用一些正确的安全措施可以尽力达到你可能失掉多少和你能够拿出多少来花在减少损失上之间的平衡。

研制方法

任何技工都知道好工具的价值。好的工具使有些任务更容易完成。而另外一些任务有可能完成。研制计算机系统所用的工具也是同样的情况。研究控制所用的方法是控制质量的主要决定因素。研制工作的重要安全方面如下。

目标

必须确定与操作性能，费用和其它因素有关系的安全的重要性。这样将帮助研究人