伯克利大学的数据库研究

Michael Stonebraker

伯克利的数据库研究可以分为四个方面。其一是M.Stonebraker 领导的POSTGRES项目，正在建立一个具有新颖的对象管理、知识管理和时程(time travel)能力的下一代DBMS。其二是L.Rowe领导的PICASSO项目，正在为使用POSTGRES和其它DBMS构造一个高级应用开发工具箱。其三是XPRS (eXtended Postgres on Raid and Sprite)项目，由伯克利的四个教职员R.Katz, J. Ousterhout, D.Patterson和M. Stonebraker领导。XPRS通过操作系统和数据管理程序开发高性能的I/O系统及其高效实用程序。最后一个项目仍由N.Stonebraker领导，其目标是更有效地处理软件错误，改进DBMS软件的可靠性。

这篇短文总结了POSTGRES, PICASSO和软件可靠性方面的工作，并报告XPRS与数据管理相关的工作。本文仅涉及EECS系计算机科学处的研究。因此，对罗伦斯伯克利实验室(LBL)和工商业管理学校所做的研究未加讨论。

一、POSTGRES项目

POSTGRES项目对准建立一个具有新颖对象管理、知识管理和时程能力的下一代DBMS。该系统由大约170,000行C语言源程序组成，现在可在DECStation 3100, SUN 3, SUN 4和Sequent Symmetry机器上运行。付少量费用，可以得到POSTGRES磁带或者从我们伯克利的计算机拷贝。本节报告该项目有关的高效对象管理、知识管理和时程支持。由于POSTGRES现在已基本可以运

前者是后者的缩影，后者是前者的扩影；而这两者相化合所得的合成结构框，则是其叠影。自然，“〔…=>〕”与“〔<=…〕”是其待征标志。

五、结 论

本文提出的N-S-Z图结构化新技术，兼容并发展了传统的图示技术与语示技术之优点，大大完善了程序设计描述技术。它不仅易于人工设计，亦便于计算机辅助设计；并且可作各种计算机语言程序的同构公共映象与共同中介图语，从而有利于从这一新思路开展对各种计算机语言支撑环境下的程序设计自动化与程序移植自动化之新研究。

参考文献

〔1〕Zhou Qihai, An Improved Graphic Representation for Structured Progtam Design, J. of Comput. Sci. & Technol., 6 (1991), 2:205—208.

〔2〕周启海著，计算机结构程序设计原理，高等教育出版社，1989年11月。

〔3〕周启海编著，PASCAL结构程序设计，重庆大学出版社，1989年1月。

〔4〕周启海、梁成华、杨联伟、龙文智编著，汉字dBASE Ⅲ在经济管理中的应用，电子工业出版社，1988年5月。

〔5〕中国科学院计算技术研究所，中国科学院计算技术研究所科研工作三十年，计算机研究与发展，23(1986), 8:13.

〔6〕I.Nassi and B. Shneiderman, Flowchart Techniques for Structured Programming, ACM SIGPLAL Not., 8:8(1973).

〔7〕C. Bohm and G. Jacopini, Flow diagrams, Turing Machines, and Language with Only Two Formation Rules, Communication of the ACM, 9(1966).

行，该项目正在致力于改进原型的性能，完成预计功能的实现和以更灵活的方式将三级式存储集成为POSTGRES环境。

1.1 对象管理

POSTGRES支持一个通用的抽象数据类型(ADT)系统，用户可以藉以添加新的数据类型和这些数据类型上的操作符与函数。当一个操作符被定义在一个数据类型上时，定义者要提供足够的信息，以使得POSTGRES优化程序为包含该操作符的查询启发式地选取一个最优查询方案。这种信息不包含函数，因此优化程序不能优化包含函数的查询。此外，允许抽象数据类型的定长和变长数组，并且查询语言POSTQUEL已扩充了对数组的标准引用。另外，还支持表(类)的多继承性，并且为特定的表定义的函数以标准的方式被继承层次结构向下继承。

通过允许表(类)的域包含过程(可执行的)对象，POSTGRES还支持复杂对象。表的域值是过程执行的结果。我们已花了大量的时间研究过程域的优化。我们考察了从需要时完全执行过程到查询改写的算法。借助查询改写，POSTGRES的过程定义被代进用户查询，为其后的执行产生编辑后的查询。此外，我们还考察了在用户请求计算之前，过程的高速缓存。我们已经得到了模拟研究和POSTGRES过程实现实验两方面的结果。

1.2 知识管理

DBMS扩充包括使用产生式规则的知识管理支持，这是POSTGRES项目的主要目标之一。这样的规则系统允许支持复杂的完整性约束，并允许一定的专家系统能够整个地在DBMS核内实现。此外，也易于实现复杂的保护和视图支持。

我们考察了两种不同的规则实现。第一种是查询重写系统。在这种方法中，来自用户的查询或更新被规则系统修改成为一个或多个可替换的查询和/或更新，确保其满足当前强施的所有规则。因此，在语法分析之后、查询优化之前，规则的支持直接出现在DBMS的高层。

与深埋在DBMS核心的实现相比，这种实现的主要好处是比低层次的规则支持减少了内务操作。此外，当少量的规则用于大量元组时，我们预料该系统的性能优于较低层次的实现。

我们正开发的第二种实现是元组级的实现。当一个事件(检索、插入、删除或更新)发生于一个元组时，一个特殊的模块，即规则管理程序检查所有可用的规则并激活它们。我们使用规则锁和规则存根记录来有效地检索影响给定元组的所有规则。当大量规则在场但每个规则只影响少量元组时，这个规则系统应当是非常有效的。

我们的知识管理研究集中在若干问题上。首先，我们已经构造了一个通用算法，支持改写所有的POSTGRES命令。第二，我们还着眼使规则系统适应于支持比当前可用的商品系统更一般的视图机制。第三，我们已经在使规则系统支持对象版本方面做了一些工作。第四，我们正在探讨可以在两种实现中实现的规则的各种语义。第五，我们已经使得两种实现运行，并正在设计一个性能基准，以在各种条件下测试它们的性能。第六，我们正在考察可以建立在该规则系统之上的较高层次的规则表示法。最后，我们正在进行各种类型的锁、各种粒度和不同锁判定算法的实验。由于对所有这些参数的最佳选择依赖于特定的应用，我们也打算设计和实现一个规则锁优化程序，为特定的规则选择最有效的锁模式。

1.3 时程

POSTGRES是一个非重写的数据管理系统，它保持数据库的状态历史，并支持对历史数据的时程查询。因此，这一研究集中在多种索引技术上，以使得对历史数据的时程查询较为容易。此外，许多概念被一般地扩充到空间存取方法。该工作包括三个主要方法：(1)段索引(segment index)，它对历史数据以及由任意的多维区间数据组成的空间

数据是有用的；(2)偏斜索引(lop-sided index)，它是一种多路树结构索引，对于支持非一致的查询分布来说，不必是平衡的；(3)混合介质索引(mixed-media index)，对临时划分的存储结构中维持的大量历史数据关系进行索引，它是有用的。下面简略介绍以上各个课题。

段索引是一种树结构索引，它在叶结点和非叶结点中存放行段。本质上，段索引概念是在最高层结点存放每个段，使得段的扩展跨越(覆盖)被其子孙结点包含(表示)的所有区间。使用这种方法，长区间存放在较高层结点，而短区间存放在较低层结点。我们的研究集中在将这一思想应用于诸如B-树和R-树等多路树结构。

偏斜索引是多路树结构索引，对于支持非一致的查询分布，它不必是平衡的。已经开发了各种分裂结点的方法来控制索引的演化，以便维持其所期望的偏斜度。

多介质索引是跨越磁盘和光盘介质的索引，以便利用二者的优点——磁盘存取时间的高性能和光盘的低成本与大容量。已考察了从磁盘到写一次、读多次(WORM)光盘转移索引结点的多种方法，以便确定这种介质索引是否可以与整个在磁盘或光盘上维持索引的方法竞争。

二、PICASSO项目

PICASSO项目包括三方面相关的工作。第一个方面集中在支持程序设计语言的持久性，在数据库中存放程序，和对这样的程序数据库提供一个易于使用的界面。第二，该小组构造了一个永久的CLOS对象的实现(虽然是一个共享对象分层结构)。在永久的CLOS顶部，构造了PICASSO程序设计环境，能方便建立高级数据库应用。最后，各种计算机集成制造(CIM)应用已使用该工具箱建立起来。我们在下面讨论这三个研究课题。

PICASSO对于UNIX系统已可以使用，包括SUN 3，Sparostation，DECStation 3100和Sequent Symmetry机器，并使用Franz Allegro Common Lisp。

2.1 程序设计环境的数据库界面

我们正在为PICASSO界面程序设计系统实现一个共享的对象分层结构。它是对*Common Lisp Object System*(CLOS)的一种扩充，允许把类说明为永久的和共享的(共享类)。一个共享类的定义和它的所有实例对象都放在POSTGRES数据库或一个商品化的关系DBMS中。我们已完成了一个单用户应用程序高速缓存的实现，允许程序使用相同的抽象引用私有或共享对象。早期的实验表明，高速缓存中对共享对象的访问时间低于对私有对象访问时间的10%。

一个典型的多用户环境将在通过网络连接到数据库服务器的工作站上运行应用程序。这种结构导致管理分布的高速缓存问题。我们正在进行不同的并发控制算法和分布高速缓存更新协议的实验。正进行模拟研究，确定这些不同算法和协议的相对性能。

2.2 PICASSO程序设计环境

PICASSO程序设计环境是一种图形式用户界面开发环境(GUIDE)。系统由一个应用框架、一个界面工具箱和一个可视的WYSIWYG应用构造器/编辑器组成。PICASSO系统是用上述的以永久对象扩充的CLOS写的，并使用X窗口系统。

应用由一组框架、对话和画面组成。这些对象称为PICASSO对象(PO)，类似于传统程序设计语言中的过程，因为它们可以带参数调用，并且可以定义局部量。通过参数化的PO库，PICASSO框架支持写可重用界面成分。

PICASSO工具箱提供各种界面抽象，可以用于创建感兴趣的图形界面。除按钮，无线电按钮，检验框，下拉式与弹出式菜单，和滚动文本工具(widgets)外，PICASSO还提供彩色图象和绘图，可以包含多种数据的滚动表，可以实时显示全动作(由磁

盘或磁带）的视频工具，和将所有数据类型连接在一起的超介质文档。该工具箱是可扩充的，用户可以容易地添加新的工具，对已存在的工具添加新的外观和在应用中增加设计工具的新策略。

它还包含了一个浏览程序，允许用户通过显示应用的“调用图(call graph)”来观察PICASSO应用的结构。启发式算法被用来以可见的方式显示该图。

2.3 IC-CIM数据库应用

使用连接到一个商品化关系DBMS的PICASSO，我们已建立了我们的IC-CIM应用程序组。它由一个设施管理程序工具，一个过程流语言，和一个超介质系统组成。

设施管理程序显示IC制造实验室的二维原理视图，并允许用户访问存放在DBMS中的其它设施和制造信息，包括设备、实用程序和场地信息。

过程流系统允许用户指明制造和测试一个半导体集成电路的操作的完整表示。过程流语言和该语言的执行程序都已构造。此外，更加复杂的启发式方法可以用过程流语言编程，以自动执行传统的手工动作。最后，执行程序支持实际运行的动态改变。该系统已用于描述伯克利微实验室的CMOS过程。进一步的实验和版本控制系统已列入规划。

为允许偶然的用户，如设施管理者和过程工程师容易地访问CIM数据，我们正在设计一个称作CIMTOOL的图形设施管理工具。CIMTOOL的独有特色是提供了一个易于使用的用户界面，它使用一种新颖的查询说明风范，允许空间的、图形的和数值的数据查询的自由混合。最近，我们也对CIMTOOL进行了扩充，包括支持视频和音频数据，制做了一个非常灵活的多介质数据库浏览程序。多介质数据使用视频指令可以用于个人训练，并将诸如图象的非传统数据与产生数据这样的传统信息结合在一起。

三、XPRS项目

XPRS项目致力于建立廉价盘的冗余阵列(RAID)，在RAID之上运行的文件系统，分布式RAID，和在RAID环境下可被数据管理程序利用的并行机制。我们依次报告这四个方面。

3.1 RAID

RAID(Redundant Array of Inexpensive Disks)已被提议作为一种以低的容量成本获得高带宽、高I/O率和高可用性的组织磁盘系统新方法。该项目已经完成了第一个由现用部件构成的原型。硬件由一个SUN4/280组成，该机具有128MB RAM，4个美洲虎双SCSI串位主机总线适配器，和32个Imprimis 5.25英寸同步SCSI磁盘驱动器。第一个原型可以运行，但遇到了I/O控制器瓶颈和高CPU开销问题。

现在，我们正在努力研究RAIDⅡ，它将包括一个定制设计的I/O控制器，以缓解瓶颈问题。RAIDⅡ的目标是支持100MB/秒的峰值传输率，和40 MB/秒的标准传输率。一个基于150个3.5英寸IBM快速磁盘驱动器的原型将于1991年夏天投入运行。

我们刚开始考虑如何以磁盘阵列为支持将三级式存储集成在一个存储层次结构中。我们选择的技术不是光盘，而是螺线形扫描磁带(8毫米或4毫米)。它们具有和小型格式化盘相同的期望特征和缺点：很高的组装系数(例如，每立方英尺以内1兆兆字节)，低成本的读带机，但传输率低(250~500千字节/秒)。我们正在考察跨越多磁带（即磁带阵列）的水平校正以及从货架到输入机装带的机器人处理的实现方法。另一个有趣的领域是将应用专用的压缩嵌入I/O系统，以便在系统级进行压缩。其目标是改进严格限制的带宽，而不是增加已经很大的带容量。技术上的挑战是如何从应用向I/O系统提供提示信息，使得最有效的压缩技术被使用，以及如何在变长和被压缩的数据上建立索引和目录

结构。

3.2 RADD

RAID具有所期望的性质：它们可以免于磁盘崩溃，并且对每个G盘组只需一个附加盘。因此，高可用性的空间花费与传统的模式相比是适中的。传统的模式以100%的空间花费镜象映射每一个物理盘。

该研究的目的是将RAID的概念扩充到分布式计算机系统。我们称结果结构为RADD (Redundant Array of Distributed Disks)。RADD能够支持横跨计算机网的数据冗余拷贝，其空间花费与RAID为局部数据进行数据冗余拷贝时相同。这种拷贝提高了单个站计算机系统在临时和永久性故障以及磁盘故障出现时的可用性。因此，RADD将被看作传统的多拷贝技术的一种替代方法。此外，RADD也是对诸如热备份(hot standby)等高可用性模式的候选替代方法。

我们的研究集中在识别出RADD可行性的关键问题。所有的RADD算法都直接在分布环境下运行，然而分布会产生它自己的值得注意的问题。首先，我们必须考虑当基础网络不完全可靠时，如何进行更新。算法还必须处理每个站上的不等量存储。其次，我们必须能够在同一网络上混合不同大小的组，还要不必全局重新构造就能从一个RADD添加或减去磁盘存储。最后，我们已经使用我们的基于DECStation 3100环境的LAN，构造了一个RADD原型，并且正在进行细调。我们的目标是大幅度地提高性能，以及进行事务处理中产生的奇偶更新的不同算法的实验。

3.3 RAID环境中的文件系统和事务

这一研究集中在文件系统分配策略，这些策略导致的文件系统的性能，和在这种文件系统内和文件系统上提供的事务处理支持。我们对利用基础磁盘阵列的优点特别感兴趣。文件系统的设计空间被分为分配磁盘存储以优化写的文件系统和分配磁盘存储以优化顺序读的文件系统。第一类出现于日志结构(log-structured)的文件系统，它将修改过的页面顺序地写到磁盘块，而不管块的先前位置。由于来自多用户的块可能散布在盘上，因此逻辑上的顺序读可能不是物理上的顺序。第二类是基于范围和基于大块的文件系统，它确保逻辑上的顺序读是物理上的顺序。

我们已构造了一个多种读优化和写优化文件系统分配策略的模拟程序，并对设计用来提供事务处理应用，巨型机应用和分时环境的广泛负荷进行了实验。我们的模拟表明，读优化文件系统并没遇到因内部和外部碎片而导致的磁盘低利用率。并且，基于范围的文件系统对范围广泛的负荷提供了最好的I/O性能。我们期望在一个实际的文件系统中实现这些策略中的一个或多个，并将测量的结果与我们的模拟进行比较。

我们还研究了在读优化和写优化环境中事务处理系统的性能。我们的研究集中在两个方面：在文件系统内部使用物理加锁和登录实现的事务管理，和在文件系统外部使用逻辑登录由DBMS软件支持的事务处理。我们的模拟指出，在一个写优化的文件系统内支持的事务处理看来可以与DBMS实现竞争，特别是在小型事务负荷下。我们期望对两种文件系统增加事务处理支持，并比较它们的性能和实现的难易。我们还期望将实际的性能与我们早先的模拟进行比较。

3.4 XPRS的并行机制

对于XPRS，我们的目标是在一个包含RAID的共享主存的多处理机计算机系统中，获得近似线性的查询加速。此外，我们还把注意力集中在多用户环境下的装载平衡和资源的智能管理问题上。有两个问题我们正在试图解决：优化和调度。顺序查询处理规划的搜索空间要足够小，使得传统的优化程序可以进行穷举搜索。然而，在多用户环境下，在所有可能的并行查询处理规划的搜索空间中进行穷举搜索的代价太高了。我们克服这一复杂性的策略是把优化分成两遍：第一遍，我们固定缓冲区的大小，然后，在第

二遍根据当前可用缓冲区的大小动态地调整规划。

调度的问题是并行查询之间的资源竞争。我们考虑的主要资源是CPU周期，磁盘带宽，和主存缓冲空间。并行机制太少将使得资源利用不充分，而并行机制太多会使机器饱和、颠簸。我们正在设计和评估调度算法，这些算法将选择并行的最优程度和最优形式。

在过去的一年里，我们已经实现了XP-RS的并行查询优化程序和执行程序。我们的最初性能结果表明，该系统确实实现了近似线性的加速。我们的实验还表明，我们的两遍优化策略在大部分情况下并不损害并行方案的最优性。关于调度问题的进一步研究仍正在进行。

四、软件错误

软件错误(而不是硬件故障)正在成为数据库不可用的主要原因。因此，我们已集中注意力于出现这种错误的情况下，提高数据的可用性的机制。

我们的方法已扩展到操作系统，以允许对诸如锁表、缓冲池、缓冲池元数据等重要的共享数据结构进行写保护。因为当需要访问时，DBMS软件必须明显地解除对数据的保护，所以要阻止由未初始化的指针和数组越界导致的错误危害这些结构。

利用一种写时拷贝(copy-on-write)的更新方式，保护机制能限制错误的传播。当一个事务首次试图写入一个记录时，DBMS拷贝该记录到一个可写的存储区。在该事务结束时，一个系统调用将拷贝这些改变到写保护的存储器中。如果一个讹误的事务在该事务处理结束之前检测出它的错误，则共享数据并不曾失去保护，并且不曾被修改。推迟更新也限制了允许访问未保护的存储器的模块数量，并降低了系统调用的开销。

我们当前正在估价保护的性能和可靠性。最初的性能测试表明，当记录较小（小于1k字节）时，保护和推迟更新导致性能下降百分之二到百分之四。对于可靠性估计我们使用取自商品程序的故障数据来开发一个软件错误模型。根据该模型，将错误播入POSTGRES，我们可以观察由于保护而引起的错误检测和传播的变化。(参考文献略)

〔范明 译自《SIGMOD RECORD》Vol. 19, No. 4, Dec. 1990, pp113-118, 仲源校〕