

未来十年数据库系统研究方向

——Asilomar 报告

杨冬青 裴健 唐世渭 编译

(北京大学计算机科学技术系 北京 100871)

Asilomar 报告是数据库领域 16 位国际知名学者对未来十年数据库系统的研究方向进行讨论后形成的报告。报告认为,数据库学界应该拓展研究范围,鼓励更多的高风险的长期研究工作,要着重解决海量联机数据的获取、存储、分析和表达等问题,要扩充数据库管理的概念,使之包容 Web 及其它联机数据存储的所有内容,要根据技术的转变来重新考虑数据库的基本前提假设。

1. 引言

1998 年 8 月 19 日至 21 日,16 位来自学术界、产业界和政府的数据库系统研究专家 Phil Bernstein, Michael Brodie, Stefano Ceri, David DeWitt, Mike Franklin, Hector Garcia-Molina, Jim Gray, Jerry Held, Joe Hellerstein, H. V. Jagadish, Michael Lesk, Dave Maier, Jeff Naughton, Hamid Pirahesh, Mike Stonebraker, 和 Jeff Ullman 荟萃于加利福尼亚的 Asilomar, 研究未来十年数据库系统的研究方向。这次会议的形式与过去十年中的几次类似的会议相近(例如 1988 年 Laguna Beach 会议, 1990 年和 1995 年 Lagunita 会议, ACM1996 年会议中的“数据库系统战略方向—打破陈规”报告)。会议的主题是讨论当前数据库系统的研究方向,并提出适当的建议。Asilomar 报告总结了这次会议的成果。

在过去的十年中,数据库系统学界在查询优化、对象-关系数据库系统、主动数据库、数据复制和数据库并行性等领域取得了重大概念性突破。这些思想成功地转化为产业成果,研究学界应该为其业已取得的成就而自豪。

但是,值得关注的是,学界大部分的工作集中在不断精炼这些思想,这种方式被称为“delta-X”研究模式。这类“delta-X”研究常常只关心短期目标,致力于改进某个已广为人知的思想 X。通常情况下,这些思想的本质已经出现在一些产品中,因此,这种“delta-X”研究可以获得风险资本的支持,在产业开发实验室中进行。

应该鼓励数据库研究学界避开这种“delta-X”

研究,拓展研究的重点,探索那些主要用于十年以后的应用之理论和技术,而把面向短期目标的工作留给他人。各种资助机构和程序委员会应该鼓励这种具有远见的研究,因为高度创新的研究工作虽然具有比较高的风险,但它们要比那些着眼于短期目标的改进工作具有更高的层次。

在过去的十年中,数据库系统的基本问题发生了显著的变化,因此,数据库系统的研究工作要探索大量新问题。数据库学界应认清研究方向,把重点从面向短期目标的改进工作转移到研究新领域上来。

报告的后继部分组织如下,第 2 部分讨论了改变数据库系统研究方向的驱动因素。这些讨论提出了一些问题,这就是在第 3 部分所建议的数据库系统研究方向。第四部分是总结。

2. 驱动力量

改变数据库系统研究重点的驱动力量主要有以下三个方面:

1) Web 和 Internet 使得能便捷地在电脑空间中发布各种信息,几乎每个人都可以访问这些信息。

2) 越来越复杂的应用环境更加迫切地需要把程序和系统集成起来。

3) 硬件的发展推翻了目前数据库管理系统技术所沿用的那些假设和设计策略。

相信读者也一定注意到了这些趋势,面临以上的新变化,数据库学界必须重新认识数据库研究的方向。

2.1 Web 为数据库研究提供了用武之地

Web 和相关的工具使得创建信息的开销大幅

度下降,然而,真正的革命性在于 Web 使得信息发布几乎是免费的。几乎每个人都可以用低廉的费用,发布大量的信息,这是前所未有的。Web 是传递应用和信息的主要平台,不断增加的网络带宽正进一步加速这个过程。

这对数据库系统研究来说无疑是一个好消息,因为 Web 就是一个巨型数据库。但是,数据库研究学界迄今为止在 Web 方面的工作为数不多。数据库系统在当前的 Web 结构中仅仅充当一个配角,尚未进入这个有机整体。首先,数据库系统通常用作高端 Web 服务器,在这些应用场合下,Web 管理者只不过把上百万个页面不加变化地转换到一个由数据库技术管理的 Web 站点上,以替代文件系统技术。其次,电子商务中所使用数据库系统,往往只用于以传统的方式来保存用户的档案、交易、单据和簿册。第三,大型的信息发布者只是把数据库系统用作存储信息的仓库。但是,最大的那些 Web 站点,特别是那些搜集信息的公司的站点,还没有采用数据库技术。而且,小型的 Web 站点通常使用文件系统来存储信息,例如静态的 HTML 页。

可以预见,今后的 Web 要管理动态的信息,而不仅限于静态 HTML 页。例如,分类零售商不会简单地把商品目录转换成一系列的静态 HTML 页面。他们会设计一个电子目录,使得客户能够不必浏览就能找到所需要的项目,例如,客户不用浏览就可以知道,商家是否出售纯棉大号衬衫。零售商还要提供个性化的模特,以显示客户穿上要买的衣服后的形象。个性化需要非常复杂的数据模型和应用。要支持这类下一代的 Web 应用,就需要非常复杂的数据库服务。

而且,HTML 正在扩充为 XML。XML 是一种描述结构化数据的更好的语言。不幸的是,XML 可能会给数据库系统带来混乱。XML 中正在发展起来的查询语言类似于 25 年前流行的过程化查询处理语言。XML 要求有客户端数据缓冲,以支持更新操作,这使 XML 设计者陷于分布事务的困境中。不幸的是,目前 XML 的绝大部分工作并没有得到数据库系统学界的参与。

Web 信息创建者需要有效的工具来迅速而低开销地构建巨型数据存储,以支持复杂应用。这反过来对数据库技术产生了巨大的需求,要求能自动地创建、管理、搜索信息,保证信息安全。Web 使用者需要有相应的工具来发现和分析 Web 上的信息。

这些趋势为数据库研究人员发挥自己的长处解

决新问题提供了用武之地。

2.2 将程序逻辑统一到数据库系统中

早期的数据库系统只关心用户数据的存储,而把程序逻辑交给其它子系统。出于对性能和方便性的考虑,关系数据库系统中后来加入了存储过程和触发器机制。目前的数据库产品允许应用系统存储并激活数据库系统的过程。随着对象-关系技术的出现,以及 Java 越来越成为一种标准的语言,使得数据库系统能够和程序逻辑相互结合,其中,程序逻辑是以标准的程序设计语言和类型系统来表述的。由此可见,数据库系统正从单纯对数据进行存储和操纵逐步转变为对逻辑和数据进行存储和操纵。

然而,还有很多工作要做。仓储(repository)是典型的程序逻辑数据库。在大多数已有的系统中,并没有很好地支持仓储的需求,例如版本控制与浏览等。显然,在当前的数据库系统中,代码不是最基本的对象,它们与数据的地位并不相等。

数据库系统从单纯对数据进行存储和操纵逐步转变为对逻辑和数据进行存储和操纵,这是非常重要的。大规模的企业有成百上千的大型复杂应用。这些应用之间的互操作性对企业的灵活性至关重要,企业在推出新的基于 Web 的应用服务,适应变化的需求,缩短对市场的响应时间,降低成本和实施业务合并时都依赖于这些应用之间的互操作性。要解决这些应用集成问题,就需要数据库技术的新成果。

当前,大规模应用的系统集成主要采用软件工程方法,注意开发过程、工具和语言等技术。数据库领域在这个方面应该大有作为。这要求数据库系统对应用提供更多的支持,要解决这个问题,一方面靠对象-关系技术,另一方面要开发更适合于管理应用接口描述的技术,并开发高层模型驱动工具,这些工具充分利用接口描述,以利于应用系统的集成、演化、移植和替代。这里,应用系统既包括单个系统,也包括由多个系统组成的整个大系统。

2.3 硬件的进展:上至兆级服务器,下至电器规模的可扩展性

在未来十年内,摩尔定律仍将成立,CPU 会越来越快,磁盘容量会越来越大,在广域通讯速率方面将出现突破。在未来十年内,百太(10^{14})字节的数据库将普及,在这些数据库中,将使用太字节的内存来作为缓冲区。这样一来,除了一些最大的数据库表外,其它所有的数据库表将驻留在内存中,这些技术进步打破了目前数据库系统体系结构的基本假定。在新的计算机体系结构下,数据结构、算法和工具都

要重新进行评价。

也许更为重要的是,计算机资源和人的精力之间的相对成本发生了变化,人的精力是宝贵的资源。这种新的经济学要求计算机系统全面自动化,能自动安装、自动管理,自动恢复和自动程序设计。计算机可以增进人的智能,它可以分析和汇总数据,组织数据,智能地回答显然的问题,当有意义的事件发生时,它可以通知人们。

整个企业范围的应用软件包的普及给数据库系统增加了巨大的压力,这些应用软件包如 SAP™、Baan™ 和 Peoplesoft™ 等。一个数据库系统应用具有 50,000 个并发用户是很常见的需求,这样的应用所使用的计算引擎和数据库系统必须在可扩展性和可用性方面有跨数量级程度的改进。

如果按这样的技术趋势发展下去,大型组织机构将具有拍(10^{15})字节规模的存储,这些存储由数以千计的处理器来管理,其处理器的规模将成百倍地超过目前的水平。现在数据库学界正为其在事务处理和数据分析的并行处理上取得的成就而自豪,但是,目前的技术不能满足两个数量级以上的可扩展要求。

在十年内,数十亿人将使用 Web,除此以外,数万亿新装置将连接到 Web。在下一个十年内,在智能卡、电话和其它信息电器内将使用功能越来越强大的计算机。在各种可移动的装置(如 Palm Pilots™)和蜂窝式电话中将会有充足的计算引擎,而且,机顶盒和其它家用电器将充分计算机化。智能大厦中的电灯开关、售货机和许多电器中都将安装计算机,每种商品上都会贴上一个标识芯片,所有这些信息电器都有其内部数据,要与其它数据存储进行交互。每个新的装置上都可能要用到数据库系统技术,因为它们中的绝大部分都要存储和管理一些信息。

从这些新的装置上我们可以预见到,数据客户机和服务器在规模和可扩展性方面将有很大的发展,数以万亿计的新装置将需要数以十亿计的服务器。新的装置数量之多、可移动性之强、间歇性连接之频繁,都使得目前的客户-服务器和三层软件体系结构不能支撑这些装置。大多数新的装置没有用户界面,没有数据库管理员,它们必须是自管理的、非常安全的和非常可靠的。遍及各个角落的新装置是下一节所讨论的研究方向的主要动力。

3. 关于研究方向的建议

本节讨论值得特别注意的研究课题。前面所讨

论的因素是这些研究课题的推动力。为了简明起见,我们把这些题目分为五个主要方面,依次讨论。

3.1 即插即用数据库管理系统

在这儿“即插即用”这个短语表述两重意思。首先,由于新装置中的数据库没有数据库管理员,因此,一个新装置中的数据库必须能够具有自调整的能力。在这些数据库中,可能没有人工设置的参数,因此,数据库系统必须能适应条件的变化。这称作无干预运行(no knobs operation),数据库研究学界应探索如何使得数据库系统不需要外界干预就能自适应地进行调整,也就是说,要从目前的数据库产品中取消众多可由用户指定的性能调整参数。这项工作的下一步是要研究物理数据库设计,如自动索引选取技术等,它们在目前产品和最近的研究中已受到一些重视。更一般地讲,系统还必须有助于逻辑数据库设计(例如表和约束),有助于应用设计,能自动地产生有用的报表和工具。无干预的系统在条件变化时必须能自适应,以保证随着时间的推移,仍然具有良好的性能。

虽然在这儿不想指定一个特定的解决方案,但是,让数据库系统记录所有处理过的通信量是一种有希望的方法。然后,用一个嵌入在数据库系统中具有对系统进行调整的详细知识的智能机制来检查这些通信量,并自动调整系统。这种做法的一个好处是传统的商业数据库系统会变得容易管理得多。事实上,大多数的机构没有足够的数据库管理人员,因此,无干预运行将非常有用。

即插即用数据库系统的另一重意思与信息发现有关。前面已经谈到,Web 是一个巨大的数据库,而且,大多数商业企业难于把它们的各种系统集成起来,这些系统称为“信息岛”。因此,应能把一个数据库系统连接到公司的网络或 Internet 上,让数据库系统自动地发现在网络上能访问到的其它数据库,并与之交互。这种发现与识别数据的支持和操作系统对硬件提供的支持相类似,操作系统能发现和识别所有可以访问到的设备。

这个信息发现过程要求数据库系统提供更多的元数据,以描述它们所管理的对象。此外,数据库系统要有丰富的数据类型转换功能。当然,还可能有什么信息发现的方法。

3.2 数百万个数据库系统的联合

数以十亿计的 Web 用户将访问数百万个数据库,企业将建立大规模的联邦数据库系统,其基础是目前正投入大量资源建造的许多分散独立的系统,

而且,Web 就是一个大规模的联邦系统。我们必须使得这些数据库中的信息易于集成。在建立可扩展的联邦系统方面,有几个主要的挑战。

首先,需要能在超过 1,000 个站点以上的联邦数据库系统上有效地工作的查询优化器。在这样的系统中,各个站点必须是本地自治的。因此,一个联邦查询优化器不能仅限于制定一个查询优化规划,因为各个站点有权拒绝执行分派给它们的任务。局部的约束可能使得全局优化规划不可行。此外,在各个站点上的负载可能变化。传统的基于静态开销的优化器在计算优化规划时,假定查询是运行在网络上的唯一任务。这样的规划不是“负载敏感”的,即使它是负载敏感的,在编译到运行的过程中,以及运行期间负载是可能变化的。在一个动态网络中,优化器必须适应负载的变化。在一个联邦数据库系统中,在各个节点中可能有多个副本,这些副本的质量(及时性)是各不相同的。优化器必须能够处理这样的服务质量问题。由于这些原因,应该重新检讨传统的基于静态开销的方法,开发在新环境下的查询优化器。

联邦数据库系统的第二方面是查询的语义和执行问题。一个用户可以向有 1,000 个节点的联邦数据库发出如下查询:

“查找整个企业范围雇员的平均工资。”

传统的数据库系统在这样的情形下,通过长时间的计算后给出精确的结果,而更好的模型是让数据库系统把这个问题看作是一个信息积聚过程。数据库系统应先迅速地计算一个粗糙的结果,然后不断地精化,直到用户认为结果足够好为止。当然,这就要求对查询优化器和执行引擎做大的改变,同时,还需要将数据传递和用户界面与统计估计技术进行合成。

非精确信息不仅仅作为查询的输出,它还出现在数据源中。在这个方面,信息积累方法将有更大的作用。例如,考虑用户提交的以下查询:

“离我所住的地方 5 英里范围内有没有非常好的意大利餐馆?”

可能有 10 个或更多的餐馆数据库存储了意大利餐馆的信息,可能还有多个地理数据库。因此,这个查询引出了一个有意义的数据库集成问题。对这个查询并没有精确的答案,因为可以有不同的评判标准。查询引擎必须把这作为一个信息积累问题,甚至给出比上一个例子更不明确的结果。在这些情况下,应使用渐进的逐步求精策略。

联邦数据库的第三个方面是辅助集成过程本身

的工具,系统管理员要能够很容易地把他的系统加入到一个更大规模的联邦系统中去。如果一个基于 Web 的服装零售商决定要为一家联机旅行社提供旅行服装,那么服装订购和收款系统必须与旅行社系统联合起来。要自动完成这个集成动作就要求应用定义和数据库接口定义能融合为一个整体的数据库,同时要有相应的工具辅助技术人员协调新系统。OMG 的统一建模语言(Unified Modeling Language)标准是朝着表述这些定义的方向工作的,但是,要以可计算方式为工具搜集更多的语义,以支持进一步的自动化。

3.3 重新考虑传统数据库系统的体系结构

第 2 节中所讨论的技术趋势使得用户能实现规模越来越大的数据库系统应用。这导致了多种体系结构,包括共享内存、共享磁盘(cluster)、非一致内存(NUMA)和无共享群机等。目前的数据库系统在无共享系统方面比较擅长,因为这样的系统具有较好的可扩展特性。计算机群机也可以充分利用各个部件,降低费用。在一个大型群机中,数据库系统优化器必须处理多用户负载均衡、磁盘空间的可用性,以及可行规划和副本上的约束等问题。在这些情况下,需要重新考虑联邦数据库系统优化问题的大部分原因都还会存在。

此外,典型的计算引擎可能会有太字节规模的内存。经常被使用的表和大多数索引将驻留在内存中。这就要求对存储体系结构进行重新考察。例如,对内存数据来说,B-树并非优化的索引结构。而且,目前商用数据库系统中的缓冲、恢复和并发策略也许是不合适的。

而且,与磁盘容量迅速增加相比,寻道时间的改进相对比较缓慢。因此,一个平均寻道时间内可以传送到内存中的数据总量增加很快。换言之,一次寻道的开销与传送一个字节的数据的开销之比增长得很快。这就要求更注重磁盘臂优化的存储体系结构。那些 RAID 5 之类的“浪费磁盘臂操作”的体系结构将不适应未来的需要。

大多数机构都需要连续运转的系统。设计一个无故障的软件系统需要远程复制和动态重构。现在还不清楚怎样做远程复制更好,究竟是应该在磁盘级别上使用 RAID 思想,还是应该在数据库系统级别上把数据库系统日志发往远程节点,并进行前滚。

一些新的应用需要超大型的数据库,规模要以拍字节或艾(10^{16})字节来衡量,这些应用包括卫星图像和数字电视归档。当磁盘存储很便宜,能够在标

准的两级存储层次结构下处理所需要的数据量时,这些应用就是可行的了。而另一种可能是出现一种新的基于全息技术的三级存储设备,所以,三级存储结构是完全可能的。提供包括副本和备份服务在内的多层体系结构的艾字节规模的存储,是一个严峻的挑战。

最后,三层(厚中间层)应用体系结构越来越普及。在这种体系结构中,服务器层只运行一个程序,即数据库系统,中间层也只运行一个程序,即应用服务器。它们都需要支持数以千计的连接,优化数据库系统(和操作系统)去适应这样的环境是一个挑战。

总之,数据库系统的基本体系结构经过近 20 年的发展,日趋成熟。但是,面临着 2010 年将出现的新环境,必须重新考虑大多数基本的体系结构假设。

3.4 智能-数据:把过程和数据在数据库系统中统一起来

关于把应用逻辑作为未来的数据库中基本成员的问题,有几种思想值得深入探讨。首先,描述应用的一种可能的模型是把应用作为业务规则的一个工作流。目前,许多商家都提供这种工作流系统,以作为应用层的框架。似乎可以把工作流图编译为一个数据库触发器和报警器的集合,它们运行在一个主动数据库系统中。在数据库系统内部运行数据密集的工作流比在系统外部运行它们要快得多。然而,对工作流的支持需要系统具有支持数以千计的触发器的可扩展能力,目前的产品只能支持少量的触发器。要具有能支持 10^3 以上数量的大量触发器的可扩展能力,还需要做深入的研究和实验。

可扩展的触发器系统还有一些别的好处,因为有一些应用需要在数据项上的大量的传统触发器。例如,股票市场系统的用户希望当某个特定条件满足时(例如,某支股票达到某个价位关口),就获得及时通知。一个可扩展的触发器系统能够支持这样的在数据上定义数以百万计的触发器的应用。当然,这样的触发器系统必须能在一个无共享或联邦系统上实现。对大规模的分布触发器的有效支持是一个尚未解决的问题。

把应用逻辑作为未来的数据库中基本成员的第二个问题在于给定的程序设计语言中,过程的传统意义上的逻辑。构件是表达这样的逻辑的一种通常方法,这样的构件应该在数据库系统内部获得支持。然而,至今为止还没有这样的通用的构件。CORBA、OLE、企业 Java Bean(EJB)和 Jini 都可能获得普及。在一个对象-关系数据库系统内部支持多种(不兼

容)的构件模型是一个巨大的困难。但是,数据库学界必须使得这些模型支持数据库系统对类型和过程的良好集成。

第三个问题是可视化程序设计方法学。许多数据库设计人员使用图表框架工具来进行数据和应用的设计,这些功能强大的工具既可以对应用建模,也可以自动生成应用。如果构件要放在数据库系统内部,那么就on须革新这些方法学,使得它们能够处理对象-关系数据库系统的问题。研究适用于以上系统设计问题的可视化工具是一件很有意义的工作。

这个领域的最后一个问题是关于持久程序设计语言。对于大量应用来说,SQL 是主要的数据库访问机制。但是,少量的应用(或应用构件)要用到持久程序设计语言。如何在同一个系统中既提供有效的 SQL 又提供有效的持久程序是一个没有解决的问题,对于那些更新频繁的环境更是如此。

3.5 结构化与半结构化数据的集成

XML 的出现可能会导致大量新型数据的出现,这些数据是层次型的,而不是关系型或面向对象的。因为许多不同形式的 Web 页都适合于同一个模式,所以这类数据被称为“半结构化”的。尽管最近以来数据库学界在这类数据的查询语言和环境方面进行了大量研究,但这个领域才刚刚开始。

数据库研究人员提出了针对 XML 的声明性查询语言。但是,从产业界的活动来看,这些 XML 及其正在发展起来的数据操纵语言类似于带过程型数据访问语言的传统的层次数据库系统。数据库研究人员必须进行深入的工作,把 Web 和数据库技术结合起来,包括解决使 Web 环境的语义表达更方便等问题。这方面有代表性的问题包括处理不同类的、自描述的、可能深度嵌套的对象的集合,开发合适的声明型语言和弱一致性事务模型,自动分解、版本管理和在可更新性和缓冲之间的交互方法等。

结论 在下一个十年,研究学界应努力完成大的飞跃。第 3 节所论述的大部分问题都是巨大的挑战,但要集中在一个重要的目标上。这个目标应容易被这个领域以外的人士理解,使他们受到鼓舞,并且这个领域的成果要能被其它的领域使用,来推进各自的工作。

在未来十年中,大部分的人类信息将出现在 Web 上,在数量上达到艾字节的规模,以各种形式出现。如果没有新的工具,查找和理解我们的问题答案都将变得比现在更困难。一个理想的系统应能简

(下转第 4 页)

应于上一节的编译技术,主要内容有:

1) 针对于一些图像处理和图像匹配算法、研究高效的计算结构;2) 研究通用的 SYSTOLIC 阵列的综合技术,特别是线性阵列的综合技术,因为线性阵列结构简单,易于在 FPGA 中实现,并且仅在阵列两端有 I/O 端口,非常易于与其它部件互连。这方面已获得了一些很好的结果。

· 研究 FPGA 的非计算用途,如用 FPGA 实现微处理器的 I/O 接口,灵活性很大,并且可以硬化一些 I/O 指令,加快 I/O 操作。最近开展的一项工作是:利用 DPGA 来实现多线程体系结构的线程切换机制。

· 研究部分可重构 FPGA 的实现技术,并探讨部分可重构 FPGA 对多线程的支持技术,包括 FPGA 资源的组织、管理、调度。从想法上看,这是很有吸引力的,可是,可行性及代价还是一个未知数。

参 考 文 献

- 1 Xilinx, The Programmable Gate Array Data Book, 2100 Logic Drive, San Jose, CA 95124, 1994
- 2 Estrin G, et al. Parallel processing in a restructurable computer system. IEEE Trans. Electronic Computer, 1963(Dec.): 747~755
- 3 Turley J. Soft computing reconfigures designer options. Computer Design, 1997(Apr.): 76~81
- 4 Mayer J H. Reconfigurable computing redefines design flexibility. Computer Design, 1997(Feb.): 49~52
- 5 Villasenor J, Mangione-Smith W H. Configurable computing. Scientific American, 1997(June): 66~85
- 6 Mangione-Smith W H, et al. Seeking Solutions in Configurable Computing. IEEE Computer, 1997(12): 38~43
- 7 Gokhale M, et al. SPLASH. A reconfigurable linear logic array. In: Proc. Interl. Conf. on Parallel Processing, 1990. 526~632
- 8 Gokhale M, et al. Building and using a highly parallel programmable logic array. IEEE Computer, 1991, 24(1): 81~89
- 9 Athanas P M, Silverman H F. Processor reconfiguration through instruction-set metamorphosis. IEEE Computer, 1993, 26(3): 11~18
- 10 Dehon A. DPGA-coupled microprocessors: Commodity Ics for the early 21st century. In: Proc. IEEE symposium on FPGAs for custom computing machines, 1994. 31~39
- 11 Wittig R D, Chow P. OneChip. A FPGA processor with reconfigurable logic. In: Proc. IEEE symposium on FPGAs for custom computing machines, 1996. 126~135
- 12 Hauser J R, Wawrzynek J. Garp: A MIPS processor with a reconfigurable coprocessor. In: Proc. IEEE symposium on FPGAs for custom computing machines, 1997. 24~33
- 13 Hartenstein R, Kress R. A datapath synthesis system for the reconfigurable datapath architecture. In: Proc. Asia and South Pacific Design Automation Conf. 1995. 479~484
- 14 Ebeling C, et al. Rapid-Reconfigurable pipeline datapath. In: Proc. Field Programmable Logic. Springer-Verlag, Heidelberg, 1996. 126~135
- 15 Mirsky E, Dehon A. MATRIX: A reconfigurable computing architecture with configurable instruction distribution and deployable resources. In: Proc. IEEE symposium on FPGAs for Custom Computing Machines. IEEE CS Press, Los Alamitos, Calif., 1996. 157~166

(上接第 9 页)

明地回答问题,并通过向用户通报有意义的事件来预测问题。

因此,对数据库研究界今后十年的目标建议如下:把 Web 变成一个在下一个十年中更有用的信息工具,使其易于为每个人使用,以存储、组织、访问和分析大部分的联机人类信息。

为了鼓励在前沿领域的创新性工作,特别是加速研究重点的转变,而且,考虑到以电子方式出版的技术报告已经改变了科技文献的搜集和发布方式,因此建议主要的会议和期刊的程序委员会改变录取文章的方法。

首先,会议和期刊应不再强调其纸介质形式的文集,而应把投稿放在 Web 站点上,加上编辑的评语。建议会议采取所有的文章都张贴,或大部分文章

张贴的方式。大会报告应该分配给那些突破“delta-X”研究模式的新思想,以及在创新领域和开拓新领域方面对最新研究进展的总结。

此外,相信接收更多的文章更有利于信息发布,这将容纳更多的创新性文章,而又不必排除那些会议所钟爱的“delta-X”类结果。

最后,建议要有一个公共审阅过程。一旦一个作者把一篇文章公开,志愿审阅人就应该能在一个适当的论坛上发表他们的审阅意见。这些经过组织的对相关文章的审阅意见对有关材料进行了比较,将十分有用。

(Asilomar Report 刊于: ACM SIGMOD Record, Volume 27, Number 4, December 1998)