计算机科学 1999Vol. 26№. 4

OO7 测试标准及其在 OODBMS 上的实现与分析*)

OO7 Benchmark and Its Implementation and Analysis on an OODBMS

王欣晖 叶 峰 王国仁 于 戈 郑怀远
(东北大学计算机科学与工程系 沈阳 110006)

Abstract OO7 Benchmark, was developed by University of Wisconsin-Madsion in 1993, is intended to be used to testing the performance of database system in CAD/CAM/CASE oriented application. Now, it has been widely used in testing commercial OODBMS. This paper introduces the OO7 Benchmark with its database model, testing-operation, and presents the design, implementation and analysis on an distributed object-oriented database system.

Keywords Benchmark, OODBMS, OO7

1 引言

近些年来,一些 OODBMS 系统已经得到了广泛的应用,这些系统的设计者为最终系统的特性作了许多不同的选择,但是由于 OODBMS 所采用的技术是比较新的,不同 OODBMS 系统的性能到底差别在哪儿,对此人们还没有深刻的认识。事实上,甚至连该用什么指标来对 OODBMS 系统性能进行评价还不十分清楚。当然,对一个数据库系统的评价可以通过建立数学模型进行理论分析和仿真,但由于影响计算机系统性能的因素很多,应用环境复杂,用数学公式进行准确的描述十分困难。相反,进行实际测试提供直观的数据和图表能给出更客观的结果。从 92 年起,在各国学者的努力下,陆续提出了几个 OO 数据库性能测试标准,如 OO1[2],HyperModel[6],OO7[1,5,7]等用于对面向对象数据库系统进行性能测试。我国的数据库研究领域以往比较侧重于对数据库理论、新技术、原型系统等方面的研究,较少涉及对数据库系统的性能测试与分析,事实上,数据库系统性能测试方面的研究已经成为数据库研究领域中很重要的一部分,因此研究和开发数据库测试技术是很有必要的。本文遵循 OO7 测试标准规范完成了对一个面向对象数据库系统的 OO7 测试并对测试结果作了性能分析。

2 OO 数据库系统测试标准

第一个 OODBMS 的测试标准是最早被 Sun 公司实现和使用的 OO1 测试标准,因此也叫做 Sun Benchmark。它被设计成衡量面向工程设计应用的数据库性能的标准,所以 OO1 偏重于对 OODBMS 的导航特性、数据的简单更新,交互式的数据库应用,维护客户端缓存等方面的测试。OO1 测试标准的测试数据库是由一些部件对象及部件对象之间的联系组成的,每个部件对象有 partId, type, xCoordinate, yCoordinate, buidDate 等 5 个数据成员,每个部件有 3 条指向其它部件对象的联系边和不定数量的被其它部件对象指向的联系边,联系边也作为一个对象,有 type, length 等数据成员,由这些部件对象和联系边组成了一张对象图。OO1 测试标准包括寻找、遍历和插入 3 种操作。每种操作分为冷、热两种执行状态。已经有很多的面向对象数据库系统产品作过了 OO1 测试(但由于 OO7 测试标准的提出,现在它已经被废弃了)。相对于 OO1,由 TeKtronix 在 90 年代提出的 HyperModel 测试标准不仅包含了更丰富的模式(更丰富的对象间的关系类型和基本数据类型),而且还提供了大量的测试操作(大量的搜索、遍历和更新操作),提供多达 10 种的测试操作,而且同样要求作冷、热两种状态下的测试。由美国威斯康星大学的学者们于 1993 年提出的 OO7 测试标准在这方面跨出了最为重要的一步,它从 OO1 发展而来,设计了更复杂的测试数据库,更多的遍历、更新的查询操作;它提供了一个综合的 OODBMS 性能描述方法。OO7 测试标准所测试的

*)本课题得到第六届霍英东青年基金和辽宁省自然科学基金资助

性能指标包括：

· 不同类型的指针遍历的速度。包括缓存内数据，硬盘内数据，稀疏遍历和稠密遍历等四个方面。

· 不同类型的更新操作的效率，包括带索引数据的更新、无索引数据的更新、重复更新、稀疏更新、缓存内数据的更新、创建对象和删除对象等。

· 查询处理器在不同类型查询中的性能，包括路径寻找、范围寻找、匹配寻找和即席连接(ad-hocjoin)等。

OO7 测试标准的设计目标不只是提供单个结果数据而是一批结果数据，单个数据的测试标准有其简单清晰，易于用于系统比较的优点。但如果一个测试标准得到的是一批数据的话，那么将会反映出关于一个系统的更为详细的信息。并且，一个单数据测试标准只有在它能精确地反映出系统将作为的实际应用时才是真正有用的，而实际中大量的证据表明，根本不存在如测试标准中所描述的那样"规范"的 OODBMS 应用。因此一个多结果数据的测试标准将会在很大程度上比单数据的测试标准更能真实地反映系统的情况。

3 OO7 测试标准

OO7 模型模拟了一个工程设计数据库，基本上代表了 ECAD、MCAD 和 CASE 中的典型应用，但不针对某一特殊的应用领域。OO7 数据库见图 1，其中的关键元素是组合部件，所有的组合部件组成设计库，设计库中的组合部件个数由参数 NumCompPerModule 控制，见表 1。每个组合部件具有如 id，buildDate 等属性及两种重要联系，第一种是组合部件与 OO7 元素文档之间的双向联系，文档是用来存储描述组合部件的变长文本，文本的长度由参数 DocumentSize 控制。第二种是与原子部件集合之间的联系，原子部件通过联接组成一张随机有向图。每一个组合部件中的原子部件个数是由参数 NumAtomicPerComp 控制，该参数由数据库的大小决定，OO7 数据库一共有小、中、大三种不同规模的测试库。在原子部件图中，从每个原子部件出发指向其它原子部件的联接数，即所谓的"扇出"由参数 NumConnPerAtomic 控制，该参数由 OO7 数据库的版本决定，有三种不同的版本(扇出=3，6，9)。而从其它原子部件出发指向每个原子部件的联接数由产生有向图的随机算法决定，OO7 不作规定。组合部件和与它相关的原子部件，联接，文档等组成了 OO7 测试数据库的基本框架，为了支持 OO7 中广泛的操作，所有的组合部件组成了一个装配层次，该层次一共有七层，由参数 NumAssmLevels 控制。在装配层次的第一层，数个组合部件形成一组且属于某一个简单装配，简单装配中的某些属性描述了其本身与组合部件之间的双向联系。简单装配还能形成复杂装配，形成复杂装配的简单装配数由参数 NumAssmPerAssm 控制。一定数量的复杂装配形成一棵代表设计对象层次的树，其中的高层节点是由底层节点聚合而成。整棵树在逻辑上形成一个模块，依测试数据库的大小不同，可以有 1 或 10 个模块，由参数 NumModule 控制。模块是用来模拟数据库应用的最大子单元，应用于多用户 OO7 测试。每个模块有一个与之相联系的手册对象，手册本身是一个大文本，其大小由参数 ManualSize 控制，用于测试 OODBMS 处理大对象的能力。表 1 总结了 OO7 测试数据库中的各项参数。

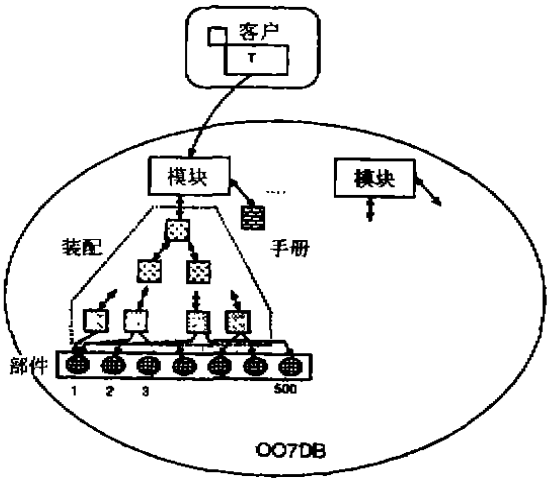

图 1 OO7 测试数据库结构

OO7 测试中的数据库操作分成三类：遍历操作、查询操作、插入/删除操作。遍历操作包含只读型遍历操作和更新型遍历操作。只读型遍历操作和查询需要做冷和热两种状态下的测试。在冷测试的条件下，被测系统的缓存应为空(包括服务器和客户端)。在热测试的条件下，被测系统应先做某一测试操作的冷测试，然后重复相同的测试操作三次，取除第一次测试操作以外的后三次测试操作结果的平均值作为热测试的结果。所有的插入/删除操作都只做冷测试。OO7 测试操作种类众多，有如 T1，T2A…T9 等十几种遍历操作，Q1，Q2…Q8 等数种查询操

作。限于篇幅，本文择其中较重要的，涵盖的测试面较宽的并且在大多数 OO7 测试中均采用的测试操作作一介绍。

表 1 OO7 数据库参数

Parameter	小库	中库	大库
NumModules	1	1	10
NumCompPerMoule	500	500	500
NumAssmPerAssm	3	3	3
NumAssemLevels	7	7	7
NumCompPerAssm	3	3	3
NumAtomicPerComp	20	200	200
NumConnPerAtomic	3,6,9	3,6,9	3,6,9
DocumentSize(bytes)	2000	20000	20000
ManualSize(bytes)	100k	1M	1M

Traverse T2：带有更新的遍历，一共有 A，B，C 三种更新模式。

· 先序遍历整个装配层次，当每个简单装配被访问时，要访问其私有的组合部件，并深度优先遍历该组合部件的原子部件图，再按不同的更新模式处理

—A：修改每个组合部件中的一个原子部件，返回所做的修改操作的次数。

—B：修改每个组合部件中的每个原子部件，返回所做的修改操作的次数。

—C：修改每个组合部件中的每个原子部件四遍，返回所做的修改操作的次数。

该操作通过遍历设计树的所有节点来测试系统在处理对象指针的寻址和转换方面的性能，同时测试对象更新的效率。OO7 测试标准中的每个测试操作分别衡量数据库系统不同方面的性能，如系统的更新效率，处理大对象的能力，处理大文本的能力等等，这些操作的结果以明确、直观的图表形式显示，便于对不同的系统进行性能比较和评估。正是凭借这些特点，OO7 测试标准成为了评价面向对象数据库管理系统性能的事实上的标准，几乎所有的 OODBMS，无论是商业产品如 Ontos，Objectivity，Versant，O_2 等，还是实验室的原型系统如美国威斯康星大学的 E/Exodus，都完成过 OO7 测试[7]，商业系统中好的 OO7 测试结果已经进入其宣传广告。

4 测试目标及被测系统

本文中的被测系统 Fish 是一个运行于 Solaris 平台、基于 NOPC 网络计算环境的分布式面向对象数据库管理系统。考察该系统在面向工程应用方面的表现是对其进行 OO7 测试的目标。Fish 系统是我们和日本九州大学合作开发的一个分布式面向对象数据库管理系统，包括分布式页式对象服务器 WAKASHI[11]，符合 ODMG-93[8] 规范，支持持久化对象的 C++ 风格编程语言 INADA/ODMG[3]。图 2 为 Fish 的系统结构。

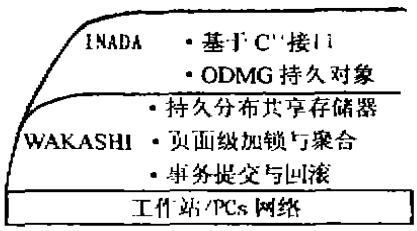

图 2 Fish 的结构

5 OO7 测试设计

OO7 测试标准只给出了标准及各个测试项的定义，没有规定也没有给出具体的被测数据库系统的结构形式，因此，我们遵循 OO7 规范进行了 OO7 测试的设计。如图 3 所示，被测试数据库位于两个不同的场地上，场地间由 100MB/S Fast Ethernet 相连，进行 OO7 测试的客户程序 OO7 Clinet 和被测 OODBMS 系统 Fish 运行于这两个场地之上。

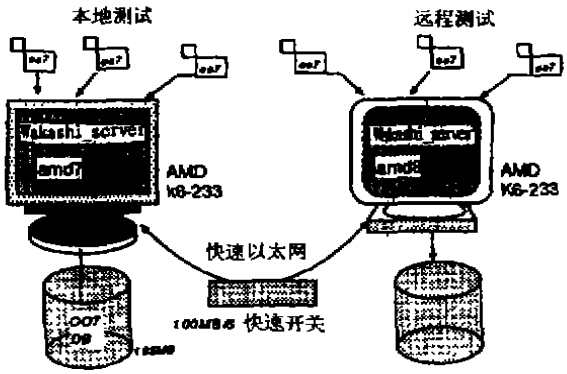

图 3 测试平台的结构

5.1 数据库模式图

本文中所采用的 OO 模型为 ODMG(Object Database Management Group)对象数据库管理组织于 1993 年提出的面向对象数据库模型 ODMG-93。用 OMT(Object Modeling Technique)[9] 方法来设计符合 ODMG/93 标准的 OO7 数据库模式。总体上说，定义了十个不同的类，其中有两个基类 Design Obj 和 Assembly，这十个类描述如下。

· DesignObj(设计)：设计类是建立 OO7 测试

数据库的一个基类。

· Assembly(装配):装配类是建立复杂装配类和简单装配类的基类。

· Module(模块):模块是含有设计树的设计的集合。

· Manual(手册):手册对象用来存放对模块对象的描述。

· ComplexAssembly(复杂装配):复杂装配对象是设计树的非叶子节点。

· BaseAssembly(简单装配):简单装配是设计树的叶子节点。

· CompositePart(组合部件):组合部件对象是由原子部件构成的部件对象。

· AtomicPart(原子部件):原子部件对象是建立一个设计的基本元素。

· Connection(连接):连接对象用来连接各个原子部件。

· Document(文档):文档包含了对组合部件的描述。

6 一个测试实例

本文以下介绍用 OO7 测试标准对被测系统进行测试的一个实例。

6.1 NOPC 测试平台

NOPC(Network Of PC)是指一组由网络联结,由 PC 组成,运行支持内存映射,多线索调度,高级网络功能(RPC,NFS…)等特点的新一代操作系统的网络计算机群。随着计算机硬件技术的飞速发展,今天的 NOPC 的整体性能已经可以和高档的工作站相媲美。事实上由高速网络(快速以太网,ATM),高性能 PC(Pentium Ⅱ CPU;128MB 或更多的内存;4GB,更大,更快的硬盘等)所组成的 NOPC 可以获得与由中档 SUN 工作站相当的性能表现。在计算机业界 DownSizing 的大趋势下,NOPC 不失为一种构架应用系统的明智选择。本文中的被测试 OODBMS 即是运行在配有 Sun 公司的 Solaris 2.5/X86 的 NOPC 平台之上的。具体配置见表 2。

表 2 测试平台配置参数表

参数	节点	处理器个数	内存
值	AMD K6-233	1 个/节点	64MB

参数	交换区	硬盘	磁盘随机寻道时间
值	1.5GB/节点	4.3GB	小于 7ms

参数	网络	页大小	
值	100MB/S	4KB	

6.2 测试结果与分析

在 OO7 测试中,要求对 3 种不同类型(扇出=3,6,9),每种类型又有 3 种不同大小(小库,中库,大库)的数据库进行 8 种基本测试操作(T1,T2A,T2B,T2C,T4,T6,T8,T9)。对每一种操作,要求做在不同场地(本地,远程),不同状态(冷,热)下的测试,共有 288 种测试操作需要完成。其中的冷测试要求在操作系统的缓存和被测试数据库系统的缓存都为空条件下进行,热测试的结果是通过重复 3 次同样的操作,取最后一次的结果。由于 OO7 的测试项较多,限于篇幅本文选择了其中的中库的 T2C 操作的结果给出分析。其它所有的数据详见 Fish 的 OO7 测试报告[4]。中库的 T2C 测试操作的结果见表 3。中库的 T2C 测试操作的结果曲线见图 4。

表 3 中库的 T2C 测试结果

	Running Time(Seconds)		
Fanout	3	6	9
Local(cold)	70.821	318.961	640.091
Local(hot)	11.602	198.971	495.916
Remote(cold)	132.082	452.017	781.984
Remote(hot)	12.810	199.623	518.953
	CPUTime(Seconds)		
Fanout	3	6	9
Local(cold)	29.750	55.153	86.187
Local(hot)	11.893	34.002	60.974
Remote(cold)	30.622	61.444	93.971
Remote(hot)	11.252	33.973	61.624

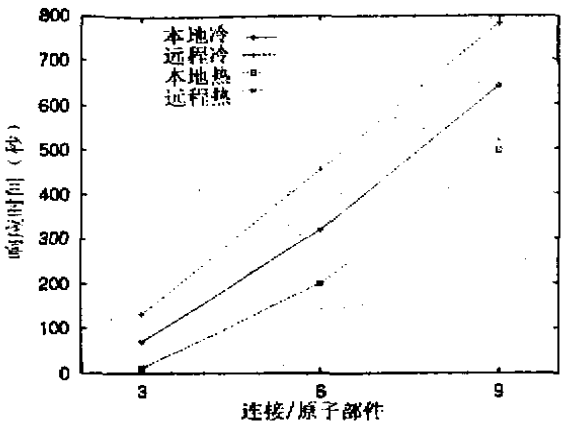

图 4 中库 T2C 测试结果曲线

6.3 结果分析

T2C 测试遍历整个装配树,深度优先遍历每个简单装配的原子部件图,并对每个原子部件作更新

操作四遍。这是典型的沿着对象之间联系进行导航操作的过程，其中主要进行的就是指针转换的操作。指针转换技术是为实现面向对象数据库系统中的对象持久化而采用的一种技术。许多对象与对象之间的联系是靠指针来实现的，在挥发性的内存中的指针与在持久性的磁盘上的指针的格式是不一样的，为实现挥发性对象的持久化，将内存中的对象存入磁盘，两种格式的指针之间的转换是必须的。在转换过程中转换算法的效率，I/O 开销是影响系统性能的主要因素。I/O 开销又由客户与服务器之间的传输粒度、缓存策略及是否存在聚簇等因素决定。可见，T2B 测试覆盖了影响系统性能的大部分方面，不同系统的比较测试中，这是很重要的一个测试项。

· 由表 3 可见，不管是本地的测试操作还是远程的测试操作，热执行的速度都要快于冷执行的速度，最高的要快 11 倍（扇出＝3，远程），最低的快 0.24 倍（扇出＝9，本地）。执行速度是指一种测试操作从事务开始到事务结束所消耗的时间，CPU 时间指的是测试操作进程处于运行时的核心态和用户态的时间，即测试进程在 CPU 中执行系统核心代码和用户代码所花费的时间，可以认为这段时间是进程在 CPU 中执行的时间，简称为 CPU 时间。执行时间减去 CPU 时间的差为进程花在 I/O 和其它一些方面的时间。由于 I/O 设备的速度相对于 CPU、内存和 cache 来说是相当慢的，因此在这段时间中 I/O 时间占了绝大部分。在热测试时，Fish 的内存 cache 技术已将部分数据装入缓存中了，在操作这部分数据时没有 I/O 开销，使热执行要快于冷执行。但实际上并没有表现出热执行远远快于冷执行，造成这种状况的原因是 50MB 的中库已经超过了系统内存 cache 的容量，系统的实存只有 64MB，除去被 Solaris 系统守护进程所占用的 24MB 外，仅剩 40MB 左右的实存，此时不管在何种情况下，都不可能将整个中库 cache 到内存中，也就是说 I/O 操作是不可避免的，此时的内存 cache 技术只能在很有限的一个范围内提高系统的性能，而不可能像小库那样有显著的变化。事实上，小库的冷执行和热执行之间的速度差距是巨大的，某些情况下热执行要比冷执行快 50 倍之多[4]。这是因为 10MB 的小库能被完全装入系统的缓存，使得在作热测试时完全没有 I/O 开销，这就大大地节省了执行时间。

· 由表 3 还可以发现，不管是热执行还是冷执行，本地测试都要快于远程测试，这种情况在扇出等于 9 时尤为明显。原因在于，远程测试除了有与本地测试相同的本地磁盘访问开销之外，还有额外的网络传输开销和远程磁盘访问开销。

结束语 根据本文所述的 OO7 测试标准的分布式设计，我们已经实现了一套符合 ODMG-93 规范的 OO7 测试程序，并完成了对一个 OO 数据库原型系统的测试，测出了该系统在 NOPC 环境下的性能表现。通过对测试后结果的分析和评价，肯定了 Fish 系统在处理面向工程应用方面的表现，比较资料中给出的其他系统的性能，发现该系统在 OO7 测试中的表现要优于一些市场上的 OODBMS 产品。同时也发现了影响系统性能的几个关键因素，主要有：持久性对象引用的实现方法，外延的索引结构，死锁的检测方法，事物的重启动等待时间等。这些因素对于该系统的改进有重要的作用。我国正在大力推进计算机软件产业，数据库软件在其中占极其重要的地位。因此，数据库测试标准和测试程序的研究与开发，对于提高我国的数据库技术有重大意义。

参 考 文 献

1 Carey J N, et al. The OO7 Benchmark. In: Proc. of SIGMOD. 1993. 12～21

2 Cattell R, Skeen J. Object operations benchmark. ACM Trans. on Database Systems, 1992, 17(1)

3 Yamamoto K. INADA/ODMG User Guide, Release 1.0, Graduate School of Information Science and Electrical Engineering. Kyushu University, Sept. 1996

4 Yu Ge, Wang Xin Hui. Report on Benchmarkning Shusse-Uo by OO7. Department of Computer Science and Engineering. Northeastern University, Spet. 1997

5 Gray J. The Benchmark Handbook 2nd Edition. Morgan Kaufmanmm, 1993

6 Anderson T, et al. The HyperModel Benchmark. In: Proc. EDBT Conf. March 1990

7 Carey J M, et al The OO7 Benchmark. [CS Tech Report]. Univ. of Wisconsin-Madison, April 1993

8 Cattell R, et al. The Object Database Standard ODMG 1.3. Morgan Kaufmanmm, 1995

9 Coad P, Yourdon E. Object-Oriented Design Yourdon Press, 1991

10 Franaszek A P, et al. Distributed Concurrency control based on Limited Wait-Depth. IEEE Trans. on Parallel Distributed Systems, 1993(11): 1246～1264

11 Yu Ge, et al. Transaction management for a distributed object storage system WAKASHI-design, implementation and performance. In: Proc. of 12th ICDE. New Orleans, February 1996. 380～389