分布式数据库系统的现状和未来

周龙骧　（中国科学院数学所）

七十年代中期以来，由于计算机网络通讯的迅速发展以及地理上分散的公司、团体和组织对于数据库的更为广泛的应用，自然地导致了对分布式数据库系统的需要。十多年来在国际范围内对DDBMS领域的研究投入了大量的经费和人力。如西德的POREL系统先后经历十一年，投资超过450万马克。法国的SIRIUS计划规模则更为庞大。美国、西欧、日本以及我国都先后研究、设计和研制了一些原型系统；国际会议每年都举行多次；发表的文章更是连篇累牍。人们预言八十年代是分布式数据处理及其核心分布式数据库管理的时代。

今天，当我们回顾过去，会得出什么样的结论呢？我们发现这个领域的工作已经取得了决定性的成果：许多基本问题被提出来并已解决；一批原型系统已经研制成功并获得了相当的经验；一些产品正在试制或已经推出，如分布式的INGRES/STAR，ORACLE公司的SQL*STAR等。总之，一句话DDBMS的技术已经成熟，其产品化的时代已经到来。

一、 性质和特征

一切计算机科学和技术的研究和开发课题其产生的源泉不外乎两个方面：应用需求和硬件发展。地理上分散而逻辑上统一的各种组织的应用需求，功能强大的计算机以至32位微机和日益广泛装备的公共数据网和局部网，说明了DDBMS不仅需要而且可能，从而从七十年代后期开始在集中式DBMS成熟技术的基础上推动了DDBMS的迅猛发展。

绝大多数DDBMS系统要求结点透明性，即用户面对的是一个统一的集中式的数据库系统，数据的分布和事务的分布式处理都是对用户隐蔽的。这样的系统注定是复杂的，开销是很大的，因此真正的DDBMS不可能以PC作实现环境就是很自然的了。这和我国早期的晶体管计算机上不可能建立真正的操作系统是一样的道理。

现在我们小结一下DDBMS的性质和特征：

1. **结点透明性** 用户面对逻辑上统一的数据库，数据分布和事务分布式加工对用户透明。用户感觉数据库就在他所在的结点上。

2. **两种体系结构** DDBMS有两类体系结构，均质的和非均质的。前者指各网络结点上的计算机，操作系统等是相同的，后者则指不同的。不同的程度可以有不同的局部DBMS，不同的OS以至不同的计算机。

大多数DDBMS原型系统都是统一地从头在网络上建立DDBMS。各网络结点上的计算机和操作系统如果是相同的就是均质的。如果是不同的操作系统甚至不同的计算机则是非均质的。对于非均质

我们在本文中花了较大篇幅介绍当前面向新的应用领域的数据库技术发展状况，目的是引起国内数据库同行的注意，加紧我们自己的研究开发工作，为提高我国数据库工作的水平，把第16届VLDB会议开成一次高水平的国际学术会议而努力。

情况，系统要在底层实现统一的系统层，以便向DDBMS的上层提供统一的接口。

与以上DDBMS的主流相反，一种完全不同的DDBMS体系结构不是从头建立DDBMS，而是将现成的各结点上的局部DBMS联结起来构成一个统一的DDBMS。这种结构的现实需要是毋庸置疑的，但要真正做到统一、可靠和有效则相当困难。例如数据模型的转换，完整性断言的转换，语言的转换，甚至数据单位的转换，数据表示的转换(公里和英里，全工资和纯工资，ASCII和EBCDIC，字长，精度以至不同的文种等)。这些问题的解决原理可能清楚而简单，诸如大家都先转换到一公共的模型上，但其实现的算法并非直截了当(例如是否等价，完备)，工作量也很大且繁琐。

3. 结点自主性(Site Autonomy) DDBMS存在着两级控制，即全局和局部。有的体系结构中结点不是自主的，只允许全局事务，即在单个结点上的事务也当作一种特殊的全局事务。在另一些系统中则允许结点的某种自主性。结点上可以有纯局部的LDBMS用户，他们只局限于该结点，完全与DDB隔离，不知道DDB的存在。还可以由结点决定哪些数据属于局部结点，哪些可以提交DDB公用。

4. 目录结构 大多数DDBMS都支持全局目录。这种面向数据对象的目录结构对全DDB的数据进行全局控制。另一种是分布式目录结构。各结点将它提供给DDB的数据通知其它结点，各用户将要用的数据记入他自己的目录中，这是一种面向用户的目录结构。在目录的构形方面大多数将目录数据和用户关系一律对待，但有的则建立独立的目录系统。

5. 数据分片(Fragment) 关系可能太大而不够灵活，因此需要将关系分割。分为水平分割和垂直分割。水平分片由限制运算将关系分割为几个子关系(例如学生关系按系别分片)。大多数DDBMS的子关系互不交叠，但VDN允许交叠。垂直分片采用投影运算将关系分割为一组组的属性组，当然应保证无损连接。

6. 副本(Replica) DDB支持副本其目的有二，一是为了提高效率，二是为了数据安全。一个数据对象在不同结点(或同一结点)有若干份拷贝，它对用户是透明的。有了副本对数据更新的一致性带来复杂性，许多论文讨论了它们的算法，但实现的很少。

7. 优化 由于DDBMS的极端复杂性，其系统开销非常可观。为了提供能够被接受的效率，优化是很重要的一环。优化分为局部优化和全局优化。局部优化采用的技术就是通常集中式DBMS的代数优化和非代数优化。全局优化则涉及通信费用，包括发送一条消息的费用，消息长度(通信量)等。对于远程网，通信费用占主导地位，对于局部网，则CPU，I/O及通信等开销均应考虑。在设计优化算法的目标函数时可以有两种选择。一种是以总时间作为优化准则，一种是以响应时间作为优化准则。对于后者则可以充分计及在网上各结点并行执行的收益。

关于优化的时机问题也存在着两种不同的策略。一种是静态的优化，一种是动态的优化。前者也叫编译时优化，后者则称为执行时优化。在优化时中间关系的尺寸是重要的依据，因此应有好的算法以便给出尽可能接近实际的结果。对于静态优化中间结果的尺寸估计失误将在后续的估算中传播和积累以致大大背离优化的初衷。静态优化方法的优点是简单，系统开销小，可以充分发掘在各结点并行执行的好处。缺点是不易有好的方法去估计中间结果的尺寸，因而难于得到最优的执行计划。动态优化方法可在执行中(一般是当两个关系连接时或考虑更多的关系运算时)估算中间结果的尺寸，以决定送往哪个结点。这样将减少错误估计的累积和传播。缺点是要作更多的统计，系统开销大，且不能充分利用并行执行的好处。为了取长补短，有的算法用混合策略，只有当中间结果的尺寸与静态的估计相差太远时才采用动态方法进行估算。

考虑了优化方法作出查询计划(Query Plan)，作出的过程存在不同的方法。一种是集中式的，即查询计划由递交事务的结点来作出。一种是分布式方法，由所有参与加工该事务的结点来共同决定查询计划。还有一种是半集中式方法，其主要决定由事务的递交结点作出，次要的决定则可由其他参与加工的结点作出。

关于优化的范围或作用域，通常的优化器只考虑对一个查询语句进行优化。进一步的考虑可以一次对多个查询语句进行优化，一个用户的后面的查询可以利用他前面的查询的结果。对于多个用户可以考察他们的平均响应时间，考虑负载对于响应时间的影响。

8. 数据模型 绝大多数DDBMS都是关系型的。关系特别适宜于分布。E.F.Codd在其图灵奖演讲中谈到由于分布式数据库系统的发展显示出层

次型和网状型模型的不足。

9. 并发控制 已经提出了封锁，时间戳和乐观方法三种并发控制技术。大多数的DDBMS都是采用封锁方法。尚未见到实现乐观方法的系统。

10. 死锁 存在死锁预防，死锁避免，死锁检测和恢复及超时（time-out）方法。死锁检测可在集中结点进行，也可在一组结点分布式检测。

11 解释和编译 联编时间(Binding time)的早晚涉及系统的效率。编译时的早联编可以提高执行效率，特别对于多次重复执行的查询其效率的改善更大。但如果从编译时到执行时数据库的状态发生了变化(如某些存在的存取路径被撤消了)则要重新编译。解释方法每次都重新联编，故可以灵活适应各种变化。

12. 网络拓扑 对于网络通信一般均假定在link 一级有相同的速率。各种算法和各个原型系统中考虑了各种网络拓扑。

13. 关系运算 并非所有系统都支持完全的关系运算。

14. 恢复和可靠性 恢复的原理很简单，就是两个字：冗余。但是相对说来对恢复的研究不太深入。由于系统开销庞大严重影响系统效率，故常常

图1 几个DDBMS原型系统

系统名	数据模型	分片	副本	是否编译	查询计划	优化因素	并发控制	死锁
SDD-1	关系型	水平垂直	支持	解释	集中式	通信	时间戳	
Distributed INGRES	关系型	水平	否	解释	集中式	通信,CPU	封锁	
R*	关系型	无	否	编译	半集中式	通信,CPU,I/O	封锁	分布式检测
POREL	关系型	水平	支持	编译	集中式	通信	封锁	预防
C-POREL	关系型	水平	否	编译	集中式	通信,CPU	封锁,时间戳	预防
SIRIUS-DELTA	关系型	水平垂直	支持	编译	集中式	通信	封锁	预防
POLYPHEME	关系型	无	否	解释	集中式	通信	无	
ENCOMPASS	关系型	水平	否	解释	集中式		封锁	超时
DDM	函数型	水平	支持	编译	集中式	通信,CPU	封锁	集中式检测
VDN	关系型	水平(可交叠)	支持		集中式		由用户封锁	

图2 一些分布式查询算法一览

论文及系统	优化方式	目标函数	优化因素	网络拓扑	是否用semi-joim	是否考虑分片	是否用副本	算法的复杂性
P.M.G.Apers,A.R.Hevner,S.B.Yao	静态	响应时间或总时间	消息个数,消息长度	一般	是	否	否	多项式(接近最优)
SDD-1	静态	总时间	消息长度	一般	是	否	否	多项式(局部最优)
W.W.Chu,P.Hurley	静态	总时间	消息长度,CPU	一般	非	否	是	指数(全局最优)
Distributed INGRES	动态	响应时间或总时间	消息长度,CPU	一般或广播	非	是	否	多项式(局部最优)
UNITY	静态	响应时间	消息长度,I/O	星形	是	否	否	
R*	静态	总时间	消息个数,消息长度,CPU,I/O	一般	非	否	否	动态规划
POLYPHEME	动态	总时间	消息长度	广播	非	否	否	动态规划
MERMAID	混合	总时间	消息个数,消息长度	一般	是	是	是	简单且有效

是做得很全面的恢复子系统在运行时不得不忍痛割爱。这里的问题是恢复是用于加强系统的健壮性和容错能力的，它是用来对付异常情况的。为了对付少量或极少的异常情况而付出庞大的系统开销以致严重降低系统在正常情况下的处理能力和效率是否合适？是否为用户所容忍和谅解？理想的办法是系统开销不致严重影响正常处理的效率而又能可靠地对付各种异常情况或故障。许多文章提出了不少复杂的协议和技术，它们在非故障情况下开销都很大，由于它们大多数并未付诸实现因而这种低效的缺陷不曾暴露得很明显。

恢复通常分为两级即局部恢复和全局恢复。

前面的两张表总结了各原型系统的特征和各算法的性质。

二、研制计划和原型系统

七十年代后期第四代数据库系统——关系数据库系统——的理论和实践都取得了深刻的极大的进展，数据库的研究和开发正处于巅峰状态。远程数据通信网已开始装备和应用(如美国的ARPANET)，法国，西德、加拿大、日本等国继美国之后正在开发全国范围的数据通信网，局部网也有了发展。在应用上的要求则更是方兴未艾。在这种背景下各国相继提出了壮观的甚至是野心勃勃的分布式数据库管理系统研制计划。当时是目标甚高，信心十足，巨额的研究经费便是明证。十来年过去了，这众多的色彩纷呈的DDBMS研制计划得到了实施，频繁的国际会议，大量的论文，提出了许多新的思想和新的算法。但是令人不很满意的是在大叠的论文中难于判别哪些仅仅是想法，哪些是真正实现了的系统。与大量的关于系统设计的讨论，关于新思想、新算法的提出等相比，关于系统集成(integration)，关于系统调试和运行，关于系统性能和关于各种算法的比较分析的文章则太少了。这就说明了分布式数据库管理系统的巨大复杂性和困难性，一些原型系统也许从未真正运行过，在许多复杂的子系统集成起来之后其性能可能令人大失所望。因此人们有理由对各原型系统的集成和实际运行的经验表示关注，只有在系统以人们能够接受的效率真正运行之后才能够说它是成功的。

下面我们对国际上著名的DDBMS作一介绍，这些实例会给我们以启迪。

1. SDD-1(System for Distributed Databases) —SDD-1由美国国防部委托CCA (Computer Corporation of America)设计和研制。SDD-1是最早研制的并且是DDBMS领域影响最大的系统之一。SDD-1是关系系统，所支持的事务是单语句事务。它的查询使用Datalanguage，由系统翻译成QUEL以便进行性能优化。并发控制采用时间戳技术和冲突图分析方法。SDD-1的原型系统在DEC机上实现，以Datacomputer作局部DBMS，结点间的通信采用ARPANET的Mail系统。其目录管理的特点是将目录数据同用户数据一律对待，也是一对一的关系，从而对它的存取可通过统一的接口。对目录本身的管理则通过目录定位器(Catalog locator)进行。SDD-1支持对关系的水平和垂直分片，并支持数据的副本。

2. Distributed INGRES 分布式INGRES是INGRES系统的后继工程，仍在美国加利福尼亚大学伯克利分校由Stonebraker等人进行。分布式INGRES是关系系统，和SDD-1一样，它仅支持单语句的事务。它的用户接口语言是QUEL。分布式INGRES在DEC的VAX机上实现，使用UNIX操作系统。它支持对关系的水平分片但不支持数据的副本。并发控制采用封锁方法。目录管理策略则和SDD-1不同，它区分两类数据，即局部数据(仅从单个结点上存取)和全局数据(可从全网存取)，并建立局部目录和全局目录。分布式INGRES的产品名为INGRES/STAR，但对其是否可称为真正的DDBMS人们仍表怀疑。

3. R* R*是在IBM圣约瑟研究实验室设计和开发的分布式数据库系统，它是继该实验室的System R之后的分布式后继项目。众

所周知System R标志着关系系统中一个新的里程碑，其市场产品即SQL/DS和DB2。R*也是关系系统，以著名的SQL为其用户接口语言。凡是在System R上能运行的程序在R*上也能运行，R*就表示任意多个系统R的意思。R*的运行环境是MVS，各结点经由CICS进行数据通信。不同结点上的CICS则通过IBM的SNA来通信。结点自主性是R*的突出特点，即每个结点控制它自己的数据，网络中不存在中央控制。R*中的对象有专门的命名机制，允许不同结点上的不同用户可使用不同的外部名去访问同一个对象，或使用同一个外部名去访问不同的对象。R*的目录采用分布式目录管理策略。这种分布式目录管理结构是面向用户的，它不同于SDD-1，分布式INGRES，POREL等的面向数据对象的目录结构。关于并发控制，R*采用封锁技术，并采用分布式死锁检测方法。R*不支持关系分割和副本，它提供了很好的授权检查机制。据称R*已投入试验性运行。

4. POREL POREL系统是西德斯图加特大学以E.J.Neuhold教授为首的研究组设计和研制的分布式关系型数据库系统。它采用SQL-like的关系数据库语言RDBL作为用户接口语言，以PASCAL作为宿主语言。POREL的设计可以针对计算机网络上的异种机型，但其原型系统则仅在PDP上和VAX上运行。它支持关系的水平分片和数据的副本。并发控制采用封锁方法，用预防来防止死锁。POREL的目录结构较接近于分布式INGRES，它区分长目录和短目录。短目录驻留在网上的每个结点上，包含全局的变化较为缓慢的稳定信息，如关系的模式。长目录则仅驻留在相应的数据对象及其副本存贮的结点上，包含变化较为频繁的信息，如关系的势(Cardinality)。目录也按关系来组织，可以共用系统的并发控制，恢复及存取机制。POREL的用户接口除RDBL事务，PASCAL+RDBL事务之外还提供应用支持接口，这是更高级的面向应用的接口。POREL提供统一的通信系统，系统内各子系统之间的通信及结点之间的通信都由它管理。网上的通信遵从可靠状态控制(RSC)协议。POREL系统的原型系统先在PDP11/40，PDP11/44上后来在VAX750上实现，采用模拟的通信系统，它从未在真正的网上运行，许多设计目标并未实现。

5. SIRIUS SIRIUS是由法国政府经由INRIA发起和资助的全国性分布式数据库管理课题的研究计划。项目包含了法国范围广泛的组织和人员。这一庞大的计划涉及了许多大学和研究所，产生了一系列重要的理论成果和若干原型系统。其中最重要的是SIRIUS-DELTA系统，此外还有POLYPHEME，MICROBE，FRERES，SYSIDOR，ETOILE，TYP-R，MESSIDOR，MRDSM，MUQUAPOL等原型系统。这些系统考虑的网络是欧洲和法国的公用网，如EURONET和TRANSPAC

SIRIUS-DELTA系统的设计可适应网上的非均质结点，它是一个关系系统，支持对关系的水平和垂直分片以及同时既水平又垂直分片。并发控制采用封锁方法和死锁预防策略。支持数据的副本。建立了较完善的恢复机制。据说SIRIUS-DELTA系统已经运行。它被称为前工业原型系统(pre-industrial prototype)，但关于它的集成及运行的文章未曾见到。

6. DDM(Distributed Data Manager) DDM也是CCA设计和实现的分布式数据库管理系统。它支持Adaplex作为用户接口语言。Adaplex是将数据库子语言DAPLEX嵌入宿主语言Ada之中的一个集成化程序设计数据库应用语言。这是第一个支持Ada的并由美国军方支持的分布式数据库系统。支持Adaplex的集中式数据库系统也在CCA设计和开发，叫做LDM (Local Data Manager)，于1984年3月完成。

将若干个在计算机网上的LDM联结起来构成一个完全透明的分布式数据库系统。D-

DM支持数据副本。在某些副本破坏时系统照常运行。DDM还支持动态地加入新的结点。DDM的DAPLEX语言支持函数数据模型，它较之层次型，网状型和关系型有更强的模型化(Modelling)能力。DDM采用象R*那样的混合连接(join)和半连接(Semijoin)策略以进行优化。并发控制采用封锁方法，并周期性地检测和解决死锁。此外还采用多版本机制使只读事务永不回退，对老版本的刷新采用了高效的方法。DDM提供可靠而高效的恢复机制。在通信方面DDM引入了强化网(Enhanced network)遵从可靠状态控制(RSC)协议。

DDM的实现采用Ada语言以保证可移植性，原型系统在VAX11/780的VMS上建立。

7. POLYPHEME 它由法国Grenoble大学和CII-Honeywell-Bull Scientific Center在Cyclades network联合设计和实现，是SIRIUS计划的产物。POLYPHEME支持关系模型，支持非均质的结点。POLYPHEME的原型在1979年运行，当时是均质的且无并发控制。POLYPHEME的经验对法国以后的DDM项目有很大影响。

8. VDN 它由柏林技术大学的R. Munz后来在NIXDORF计算机公司的Intel Microprocessors上建立的。采用ISO-PASCAL实现。VDN支持关系的水平分割，各片可以交叠(这是与其它系统不同的)。它也支持数据的副本。它提供很好的授权机制。并发控制采用封锁方法和预先要求策略。出故障时可以利用具有相关副本的姊妹结点来进行恢复。

9. ENCOMPASS ENCOMPASS同一般意义的DDBMS不同，它是个别产品的汇集，允许客户在分布式环境下开发和安装联机事务处理。它是关系系统，但不保证事务的可串行性，它使用排它性封锁。

10. MICROBE MICROBE也属于法国的SIRIUS计划，由Grenoble大学在LSI-11机的广播网上实现的小型机DDBMS。它用P-ASCAL语言实现，支持功能界于QUEL和SQL之间的MIQUEL。它的分布式查询优化是动态执行的。

11. Multibase 由CCA设计和实现的Retrieval-only系统。采用函数数据模型，通过DAPLEX语言对非均质分布式数据库提供统一的存取。它没有并发控制。Multibase的实验版提供集成式存取一个层次式数据库和一个CODASYL数据库。

12. Prime Computer, Inc. 该公司开发了一个网络模型的分布式数据库系统。并发控制采用封锁方法和死锁预防策略。它对数据项保留多个版本，只读事务读取数据项的旧值，从而缩短了只读事务的响应时间。当然存贮开销会大大增加且对旧版本的刷新应有高效的算法。CCA的DDM采用了Prime的多版本方法。

13. DDTS (Distributed Database Testbed System) 由Honeywell的Corporate Computer Sciences Center在Honeywell Level 6小型机上设计和实现。它支持五模式信息体系结构，这是ANSI/SPARC的三模式体系结构的扩充。概念模式是实体关系模式的变种，以CORDAS作数据语言。局部的DBMS是CODASYL的面向网络的IDS/II系统。此研究计划的目标是用DDTS作为评价有关DDBMS的各种算法的试验台。

14. SCOOP (System for Cooperation) SCOOP是巴黎第六大学和都灵大学的联合研制计划，目的是建立一个分布式系统将现成的非均质数据库和应用程序完全集成。它使用POLYPHEME原型系统，对分布式数据提供非均质用户接口。

15. JDDBS (JIPDES Distributed Data Base System) JDDBS是Japan Information Processing Development Center设计和开发的非均质DDBS。它是关系系统，采用具广播通信功能的局部网。分布式查询优化是动态执行和分布执行的。

16. C-POREL C-POREL是中科院数学所设计并由该所和上海科技大学，华东师范大学合作实现。它以POREL为蓝本，并吸收各先驱系统如SDD-1，R*，SIRIUS-DELTA，DDM，分布式INGRES，VDN等的经验进行设计。主要目标是实用性，先进性和有限的可移植性。C-POREL是关系系统，以RDBL作为用户接口语言，宿主语言是PASCAL，以PASCAL语言实现。实现环境是用以太网联网的UV68高档32位微机，操作系统是与UNIX兼容的UNOS。C-POREL的编译程序由自行设计开发的编译程序生成工具CGT自动生成。它支持关系的水平分片。在代数优化，非代数优化，分布式查询优化方面实现了精致的算法。并发控制采用封锁和时间戳相结合的方法。恢复功能很强，局部和全局恢复采用系统日志支持。目录管理上进行了独特的设计，独立的目录管理子系统采用分离和副本机制以提高性能。通信系统划分同一结点上各系统层之间的通信和网上结点之间的通信以尽可能提高效率，在以太网的底层自行开发广播通信和全局时钟系统，并采用可靠状态控制(RSC)协议。这是国际上第二家实现RSC协议的系统。C-POREL的各子系统已完成分调和部分联调。现正进行全系统的联调。C-POREL第一版不下万行PASCAL代码。

17. H* H*是合肥工大微机所设计和研制的分布式数据库系统，它的用户接口和INGRES兼容，支持数据副本。实现环境是三台DUAL System 83/20以RS-232互连。H*系统已通过了鉴定。

18. WDDBS WDDBS是武汉大学设计和研制的分布式数据库系统。它是关系系统，实现环境是三台IBM PC/XT用3COM以太网互连。

19. DDBASE DDBASE由广州军区第四通信总站开发。它用以太网连接DBASE Ⅱ。

20. D-dBASEⅢ D-dBASEⅢ是上海第二工业大学设计和研制的分布式数据库系统。实现环境是OMNINET下的IBM PC/XT上的dBABSⅢ系统。

21. DMU/FO DMU/FO是东北工学院设计和实现的网络型分布式数据库系统。

22. D-NITDB 由南京工学院设计和实现的分布式DBMS，NITDB为该校自行设计实现的微机集中式DBMS。

23. DdBASE-Ⅱ 由南京大学设计和实现的分布式汉字数据库管理系统。

三、展望

如前所述，当前DDBMS技术已趋成熟，其中的大部分基本问题已经解决，各种市场产品正在开发并逐步推出。总之，正处于收获和已有成果深化和提高的季节。需要进一步考虑的问题有：

1. 实现经验的收集和分析比较。DDBMS的文章已发表很多，众多的研制计划也纷纷告一段落。和大量的研究文章相比，关于原型系统的集成，系统的调试运行，算法的比较分析，系统的性能分析，各种思想和方法的实现经验的总结等方面的文章则较为欠缺。

2. 由于缺乏实际运行的DDBMS系统，因此许多问题的研究不易深入。例如负载均衡性的研究，分布式数据库设计问题，用户存取模式(Pattern)的收集和统计分析问题等。

3. 用户接口问题。提供对用户十分友善的界面。多窗口的屏幕管理及将其独立于系统和应用的设计和实现。报告，表格，图形管理。自然语言接口，第四代语言接口，支持个性化用户的接口等。

4. 大数量结点(例如50000个结点)的DDBMS。在大量结点之下许多算法必须重新考虑。

5. 非均质DDBMS。

6. 和工作站的关系。

7. 和一般分布式系统的关系。