Alternative Technologies™

THE DBMS SCALABILITY REPORT

PERCEPTION VERSUS REALITY

David McGoveran
Alternative Technologies
13150 Highway 9, Suite 123
Boulder Creek, California 95006
Telephone: 408/338-4621 Fax: 408/338-3113
Internet: mcgoveran@AlternativeTech.com
Website: www.AlternativeTech.com
Report Number 970718

Disclaimer

For information about this or other reports, or other products and services, including consulting and educational seminars, contact Alternative Technologies directly by telephone, mail, or via our Web site:

Alternative Technologies
13150 Highway 9, Suite 123
Boulder Creek, CA 95006
Telephone: 408/338-4621   FAX: 408/338-3113
Internet: mcgoveran@AlternativeTech.com

Table of Contents

I. Introduction

This report, and hopefully others to follow, is concerned with DBMS scalability (henceforth simply "scalability"), and more generally, high-end database requirements. It provides the results to-date of an on-going study pertaining to scalability including (1) case studies, (2) an analysis of market requirements, and (3) supporting information from a variety of sources. The case studies were conducted as on-site audits of applications which push some or all of the limits of their respective DBMS products. Such sites were defined as "scalability sites." Prior to selection, a telephone investigation was made of each prospective site to determine whether or not it fit the definition of a scalability site. Some sites represented that they had no scalability problems, while others seemed doubtful of their ability to continue to be successful in the long term. Sites used in case studies were chosen approximately equally from among these. As of this report, only Oracle and Sybase sites have been included in the case studies. For this reason, we briefly discuss each of these companies and their approaches to scalability.

Scalability is becoming extremely important for today's enterprise DBMS. Analysts, press, vendors, and consumers alike agree that DBMS scalability requirements are becoming ever more demanding. Even so, there is considerable reason for concern with the current state of analysts', press members', vendors', and consumers' understanding of DBMS scalability. As we hope will be clear, misleading advertisements and false claims have been accepted and repeated by key influencers. For example, it is not uncommon for database size "achievements" to be stated in terms of either allocated space or planned database size. Sometimes the expected user populations and planned database sizes are presented in briefings and new product announcements. More conscientious presenters may add an aside that the current pilot is significantly smaller and may not even be in production.

Analysts, the press, and consumers seem to accept the combination of laboratory benchmarks and user plans as proof of support for VLDB and processor scalability. So far as we have been able to determine, DBMS scalability costs and benefits have never been investigated in detail, let alone published. Our study has already exposed some surprising false perceptions (myths) concerning scalability and VLDB. This report, and those we expect to follow, will expose numerous myths, fallacies, and flim-flam, while providing a source of unbiased information about DBMS scalability. We hope it will be a valuable aid to customers, vendors, press, and analysts in obtaining a better understanding of DBMS products, their capabilities, and the market.

II. Market Requirements

After careful study, it is clear that four marketing requirements (above all others) are considered important by both analysts and prospective DBMS users. The following requirements address the rapidly growing need for high-end business transaction systems:

- *tens of thousands of users online* - Today's businesses need to supply tens of thousands of users with direct access to data. Even higher levels of connectivity and concurrency will be required in the not too distant future. This is due to a combination of:

(a) improved standards of customer service, intended to permit online access to customer-related data

(b) the rise of remote electronic access to databases

- *very large databases* - Database technology is not keeping pace with the ability to collect data, nor with anticipated uses of that data. High end VLDB sizes are growing by a factor of ten approximately every three to five years. The very large databases of today are the medium size databases of tomorrow. Today's high end sites are rapidly discovering that they cannot manage such large amounts of data, and tomorrow's medium size databases will be faced with approximately the same difficulties. The increase in database size is due to:

  (a) an increasingly high volume of data capture transactions, largely due to ever larger user communities as businesses improve their ability to reach geographically separated markets cheaply and easily

  (b) the rise of data warehousing

  (c) the increasing trend toward integration of timely operational, and historical informational data

  Of course, the management of large databases, as well as the development and maintenance of large database applications, requires a single view for all users (end users, applications, and administrators).

- *very high transaction rates* - High transaction rates are often associated with OLTP systems. OLTP transaction rate requirements are indeed growing. However, it is important to keep in mind the database definition of a transaction: the transition of a database between consistent states. This definition includes read-only transactions. Read only transaction rates must be considered in any environment in which the database is changing. Very few databases can be considered read-only, even though the bulk of the user community is only reading the data. As databases become increasingly larger and the time distribution of user access expands, administrative operations (such as reorganization, version migration, resynchronization, loads, backup, restore, recovery, re-partitioning, re-indexing, statistics collection, etc.,) must be performed online and, ultimately, continuously. Most such operations involve writes to the database. As such, even informational databases must support high transaction rates with full transaction isolation (serializability). All the business trends that are driving larger user populations and larger databases are also driving higher transaction rates, whether for OLTP or informational systems.[1]

- *electronic business transactions* - The growth of Internet (and Intranet and Extranet) use is still in the early stages, and yet Internet access providers are unable to keep up with

[1] Financial transaction processing capacity planned by one major credit card company for the year 2000 (the trillion dollar mark) was recently passed a full two years ahead of schedule. Based on projected growth rates there is currently no DBMS that can handle the actual transaction rates now anticipated for the year 2000.

demand. Today, its uses are limited primarily to information access. However, as the problems of secure electronic business transactions and high volume user loads on the net are solved, electronic commerce will become much more important, ultimately overwhelming current volumes and growth rates. Synchronous transaction models and connected access are not viable solutions for this market. Users must be able to send a transactional message to the database server without requiring a connection. They must be able to obtain the results of their request at a later time. Increasingly, this model will be the dominant one for use in all distributed applications, not just Internet applications. It is particularly important for mobile (a.k.a., remote or "occasionally connected") computing. It is the "new" client-server model.[2]

Collectively, the first three of these market requirements demand open-ended scalability. Clearly, none of today's DBMS products can be expected to meet the anticipated need. When the load implied by the fourth requirement is taken into account, the requirements are impossible for today's DBMS products to meet. There is no consensus on how to meet the demand. Most RDBMS vendors do not even seem to be aware of the seriousness of the problem. Vendors who are aware propose a variety of "solutions": Only time will tell which, if any, of these "solutions" will have the desired result. This leads us inescapably to a denial of the first commonly held belief, *Perception 1*.

Perception 1: DBMSs can be produced and consumed as though they were commodities.

> ***Reality Synopsis 1:*** As ever larger database sizes, numbers of users, and workloads (such as transaction rates) must be supported by today's DBMS products, the DBMS features that lead to success or failure have become increasingly more difficult to identify. Every vendor has chosen differentiating implementations of similar functionality (and sometimes unique functionality). Customers use a variety of "workarounds" to circumvent the many unsolved DBMS scalability problems.

Case studies of existing applications have shown that most scalability problems have not been solved. Rather, customers use a variety of unsupported techniques (that is, "workarounds") to build and maintain databases that can meet high demands with respect to transaction rates, transaction complexity, database sizes, database complexity, numbers of concurrent users, and growth rates. Some of these techniques are described in the section on scalability and others are described in the case studies section.

Over the past few years, each of the primary open systems RDBMS vendors have attempted to convince the world that its products are scalable. Our case studies (and over fifteen years of direct experience with the various DBMS products while working with clients) have convinced us that each product has its deficiencies when it comes to scalability. Before we can examine these issues, we need to have a clearer understanding of scalability.

[2] Actually, this model of client/server interaction has always been the most appropriate one. I have used, taught, and recommended it for more than fifteen years with great success. This model is <u>not</u> "network computing" which offers no scientific basis on which to build a distributed design methodology.

III. What Is Scalability?

Scalability is a measure of the efficiency with which a hardware and software system can make use of the resources available to it. As used by marketing and sales organizations, and often by analyst groups, scalability has an imprecise, floating meaning. Worse, “scalable” is often used as though it referred to an absolute property, as if a DBMS were either scalable or not. In fact, scalability is a relative property. Technically, we should only say that a system is X% scalable relative to a particular resource, such as processors or memory. A reasonable definition of DBMS scalability should also take into account the limitations imposed by hardware and OS scalability by identifying the testing environment.

There are two main types of scalability: speedup and scaleup. *Speedup* is a measure of the ability of the system to do a fixed amount of work faster when given more resources. *Scaleup* is a measure of the ability of the system to do a larger amount of work in a fixed time when given more resources.

The method of computing X is precisely defined. In particular, a system is said to be 100% (or *linearly*) scalable with respect to a particular resource if and only if the amount of work achieved by the system can be doubled whenever twice the amount of resource is made available to the DBMS. A more general definition can be given. Suppose that W1 is the work possible on the system when the amount of available resource is R. Suppose further that W2 is the work possible when that same resource is doubled to 2R *while keeping other factors constant*. Then, we may define the percent **speedup P** (with respect to the resource and over the range R to 2R) as:

$$P = (W2 - W1) * 100 / W1$$

Similarly, suppose R1 is the amount of resource required to perform a fixed amount of work with load L on the system. Suppose further that R2 is the resource required when the load is doubled to 2L *while keeping other factors constant.* Then, the percent **scaleup S** (with respect to the resource and over the range L to 2L) is defined as:

$$S = (R2 - R1) * 100 / R1$$

Because there are a number of resources that might be doubled, and a number of ways in which the amount of work might be measured, there are a number of kinds of scalability. When referring to DBMS scalability, load is usually measured in numbers of concurrent users (though size of transaction might also be used). The corresponding amount of work or throughput is usually measured in transactions per unit time (for a particular transaction profile), although response time, rows processed per second, and so on may also be used.

Measuring and comparing workload requirements is not obvious. A key problem arises in particular with the concept of a transaction. The definition of a transaction (as a measure of work) varies considerably among DBMS applications. It has an even broader definition between non-relational mainframe applications and relational open systems applications. For example, transaction rate requirements or loads in a traditional non-relational mainframe environment might very well include data communications transactions, cross-region transactions, pseudo-transactions, and sub-transactions. Most of these have no impact at all on the DBMS. If such transaction rates are reported in the context of case studies or surveys pertaining to DBMSs, the definition of transaction needs to be clearly stated before any conclusions can be drawn. Also, a business transaction (conceptual unit of work) often includes multiple logical database transactions (transitions between consistent database states) which are submitted to the database as multiple physical transactions (recoverable units or DBMS commit points). Exactly how business transactions map to physical transactions depends greatly on the DBMS product's capabilities and on the application design. Ideally then, workload should be stated in terms of business transactions since these are the common requirements regardless of the implementation. Consider *Perception 2*.

Perception 2: Workloads defined in terms of the number of physical transactions a DBMS processes are meaningful for scalability.

> ***Reality Synopsis 2:*** Each DBMS requires a different implementation of a business transaction if it is to give the best performance. The number of physical transactions that result often vary greatly from product to product and environment to environment.

One of the most common types of scalability is *processor scalability*, in which the number of processors is the resource doubled. Processor scalability can be applied to DBMS software on SMP, cluster, or shared nothing systems. Processor scalability should be contrasted only among similar architectures. It does not make sense to refer to a DBMS as scaling from one to hundreds (1 - 100s) of processors when the largest SMP platform it runs on is thirty-two (32) and an MPP platform or cluster is required in the range above thirty-two. In practical terms, users perceive SMP processor scalability as important only up to some limited number of processors.[3]

An alternate, and weakly defined, form of scalability is *platform scalability*. A software product that provides either the speedup or scaleup expected from using increasingly powerful platforms (desktop, uniprocessor, SMP, cluster, MPP, mainframe), might be said to provide platform scalability. Platform scalability might be considered moderately well-defined if all the platforms use the same operating system and the same family of processors. Just because a product line uses the same name does not mean it is the same product in all architectures. Likewise, a product line can use different names and be the same product in all architectures.

All of the foregoing should serve to explain that scalability is not a simple characteristic of a DBMS. Clearly, a DBMS might even be 60% scalable with respect to CPU resources over the range of one to forty-eight CPUs while another is 80% scalable over the range of one to twenty CPUs. The invalidity of *Perception 3* should be equally clear:

[3] In a report dated November, 1995, the Gartner Group estimated this number at twenty (20).

Perception 3: A DBMS is either scalable or it is not.

> ***Reality Synopsis 3:*** There are many types of scalabilty: processor scalability, platform scalability, administrative scalability, etc. A DBMS can be said to be "X% scalable" only with respect to a particular resource or workload and over a certain range. As a qualitative property of DBMSs, scalabilty is not meaningful.

Linear (100%) scaleup or speedup on one platform might easily translate into extremely poor scaleup or speedup on another platform. There is a difference between products that must preserve existing hardware and operating system scalability (such products have an advantage) and those that must circumvent a lack of it. Practically any scalability or speedup is possible for any DBMS if a specific application is carefully selected. Scalability is, in fact, specific to the intended application's characteristics. For example, a read-only application might scale extremely well while a read-write application does not: the differences in DBMS resources used are extreme. It follows that demonstrating scalability is sensitive to the platform and application, contrary to the evidence often presented (and blindly accepted). See *Perception 4*.

Perception 4: A demonstration of DBMS scaleup and speedup on any platform and application of the DBMS vendor's choosing is sufficient, irrespective of the intended platform and application.

> ***Reality Synopsis 4:*** Practically any scalability or speedup is possible for any DBMS if a specific application is carefully selected. Scalability is, in fact, specific to the intended application's characteristics (for example, read-only versus read-write). Transaction and database design also have a powerful effect on scalability. Subsecond response times for a few users often go to tens of minutes or even hours when the load and resources were scaled up with some designs, but scale smoothly when redesigned.

In fact, transaction and database design can have a powerful effect on scalability. During site investigations, it was not uncommon to find transactions and database designs that offered subsecond response times for a few users. It was also not uncommon to find response times measured in tens of minutes or even hours when the load and resources were scaled up. Even more important, these applications scaled quite well when the transaction and database were redesigned. For example, response times for row-at-a-time processing algorithms rarely scale compared to set processing algorithms. During site investigations, rewriting queries to use set processing often resulted in significantly improved scalability. Such rewrites often improved elapsed time by a factor of ten or more.

When analyst groups say that a DBMS product is scalable, the meaning is completely unclear. Most often they *seem* to refer to processor scalability and specifically speedup (namely, increase in transaction rate with increase in number of processors). However, analysts have not been consistent in providing numbers to back up their conclusions that a product is or is not "scalable" (in any sense of the term).

Even if one product can be said to be "more scalable" than another, the value of that scalability must be taken into account. An example may suffice to explain this issue. Suppose that the scalability of a particular product A with respect to some resource is X (say 80% for definiteness) and greater than that of some other product B (say 70%). Suppose further that product A supports more complex configurations than product B. (For example, product A supports SMP platforms with larger numbers of processors or clusters with more nodes than does product B.) Superficially, it would appear that product A is a better purchase than product B.

This is not a good conclusion. To see why, suppose the performance and the cost of the highest performance systems that can be configured with each product are compared. Suppose further that it turns out that product B actually outperforms product A with a lower purchase cost, lower maintenance cost, and greater ease of operation. Clearly, the better choice is then product B, *despite B having the poorer scalability*. This simple argument demonstrates the invalidity of a second commonly held and argued belief in the industry, *Perception 5*.

Perception 5: The more scalable the system, the more efficient and cost effective the product.

> ***Reality Synopsis 5:*** It is possible to have two systems with identically the same percent scaleup or speedup, but with widely differing absolute throughput. The product with the poorer scalabilty may well provide better performance over the range of available resource. Most published scaleup or speedup percentages are derived from TPC (and related) benchmark numbers. Unfortunately, these tests do not properly measure scalability since more than one resource is allowed to vary.

In particular, it is possible to have two systems with identically the same percent scaleup or speedup, but with widely differing absolute throughput[4]. Part of the problem is that most scaleup or speedup percentages are derived from TPC benchmark numbers. Unfortunately, these tests do not properly measure scalability since resources are not held constant. Indeed, for this reason it is important that quoted TPC benchmarks should include not only transaction *rates*, but transaction *costs*.

Another type of scalability that is as ill-defined as platform scalability, but is much more important, might be called *administrative scalability*. For OLTP databases, administrative performance requirements have been determined traditionally by the size of the database and a window of available time to perform administrative operations. Gartner Group's Radcliffe (reference 13) asks the question that often arises, especially for OLTP VLDB administration: "Is it possible to load, back up, restore, reconfigure and reorganize the data within the time window the application allows?" Certainly the speed with which administrative operations can be performed is crucial to scaling up a database (either in size, concurrent users, or transaction rates). Administrative operations should be as susceptible to scaleup and speedup

[4] (For example, consider the TPC-C 4 way DECchip 21164/400MHz results under Digital UNIX. Oracle achieved 6056.04 TPM while Sybase achieved 7598.63, even though Oracle is generally considered more "scalable.")

considerations as any other database work. Nonetheless, the question of operating within an available window of time is becoming irrelevant: As databases push size limits, the available time window is also shrinking. The ultimate result will be an inability to perform administrative operations offline, regardless of any speed considerations.

Radcliffe's question assumes that the stated operations must occur within the window because they cannot be performed at the same time as online operations. To the contrary, there is no inherent reason that administrative operations cannot be performed online so that no window is needed. The administrative workload imposed by a product, and whether that workload can be accommodated during ordinary operations, is much more crucial to evaluating overall DBMS scalability. This is obvious if we consider the only slightly artificial case of a DBMS product A which has a highly efficient index reorganization utility while product B either has an inefficient utility or none at all (requiring index re-creation instead). One would assume product A to be the better product with respect to this operation. However, if product A requires offline index reorganization (and therefore a "window" that increases with index size) while product B performs index reorganization dynamically (that is, continuously) and online, this assumption is wrong. Even if the online operation impacts performance negatively, additional resources can compensate far more effectively than for the offline operation.

The frequency with which a product might require reorganization must be taken into account when comparing the scalability of products. In general, the importance of dynamic or continuous administrative operations of all types should not be minimized. Such features are much more valuable with respect to scalability than superficially similar features which simply limit the impact of an operation to some unit of storage. For example, taking a portion of a database offline to perform backup does not scale as well as being able to perform backup online continuously with low impact to user operations. This is because the impact of a taking a portion of a database offline becomes less unpredictable as a database scales up. In most cases, physical redesign of the database will eventually be required in order to preserve availability.

The costs of administrative operations are often limiting factors on scalability. Although it is possible to build a terabyte database with a number of DBMS products, consider the difficulties of backing up such a database. At a reasonable forty (40) GB per hour, a terabyte requires 25 hours. Similarly, when refreshing a data warehouse takes greater than twenty-four (24) hours, resynchronization is almost impossible and there is insufficient time for defragmentation and query tuning. See *Perception 6*.

Perception 6: The speed of administrative operations suffices to determine administrative scalability.

> ***Reality Synopsis 6:*** All the speed in the world will not lead to administrative scalability if administrative operations must be performed offline, the database is sufficiently large, and the available "window" is sufficiently short. Conversely, if administrative operations are performed continuously and without conflicting with user operations (that is, dynamically), then they need only keep up. The speed of a corresponding offline administrative utility in then of little importance.

In principle, partial database operations can meet the need. Unfortunately, partial database backup, restore, and recovery generally do not automatically maintain consistency, creating a serious manual operational load. For example, imagine the difficulties involved in recovery if the database crashes while performing a partial backup on an offline portion of the database. Even under ordinary conditions, restoring and recovering a database from a partial backup is tedious. Some operations, such as indexing, cannot be run as a partial table operation in today's products (unless they happen to contain data belonging to a single partition of a horizontally partitioned table). Clearly, such operations would be extremely valuable for newly loaded fragments of a table. See *Perception 7*.

Perception 7: Partial database operations provide administrative scalability.

> ***Reality Synopsis 7:*** Partial database operations improve availability and may offer some speedup due to parallelism. Unfortunately, the complexity of their use does not scale. Partial database backup, restore, and recovery generally do not automatically maintain consistency, creating a serious manual operational load. Even with moderately sized databases, managing the restore and recovery operations on a database from a set of partial backups is tedious and uncertain.

Although parallelism is important in improving the efficiency of a DBMS, it's affect on scalability is not nearly as clear. The range over which processor scalability is efficient is particularly sensitive to parallelism. However, once the degree of parallelism equals the number of CPUs, little additional benefit can be expected. The degree of speedup that parallelism can offer is limited by the number of processors and processor scalability. It also means that speedup is strictly limited by the inherent coarseness and serial character of the workload. Products that demonstrate "linear" speedup through additional processors do so over a limited range, with inherently parallelizable workloads, and by removing pre-existing inefficiencies in DBMS products processing that show up with lower numbers of CPUs (that is, higher overhead). See *Perception 8.*

Perception 8: A DBMS with good processor scalability can provide 100% speedup.

> ***Reality Synopsis 8:*** Processor speedup is inherently non-linear. The maximum speedup T that can be expected in a system with N processors running in parallel M% of the time is given by the ***non-linear*** Amdahl's Law:

$$T = 1 / ((1 - M) + (M / N))$$

Implementation details such as scheduling, dependencies among parallel tasks, and cache coherency reduce the achieved speedup. The ability of a system to achieve the maximum speedup with each additional processor for a highly parallelizable task would be a good measure of the scalability of a vendor's implementation of parallelism (*parallel processing scalability*). In effect, parallelism is simply a potentially more efficient means of processing, similar to incorporating solid state storage in place of disk storage. System resources have to be balanced to take advantage of such capabilities. *Perception 9* expresses this incorrect belief.

Perception 9: Parallelism is necessary for scalability.

> ***Reality Synopsis 9:*** The scaleup that parallelism can offer is strictly limited by the number of processors and processor scalability. The speedup that parallelism can offer is strictly limited by the inherent coarseness and serial character of the workload. Parallelism can help remove the cost of administrative functions as a barrier to availability by reducing the time required for loading, backup, restore, recovery, reorganization, and so on. Note, however, that at a reasonable forty (40) GB per hour, backup, load, or resynchronization of a terabyte requires 25 hours offline.

Parallelism can help remove the cost of administrative functions as a barrier to availability by reducing the time required for loading, backup, restore, recovery, reorganization, and so on. Of course, while parallelism can reduce the need for an offline window, it cannot remove it.

IV. Evaluating Database Size

There are numerous issues of efficiency that affect the value of scalability measures. One of the most misunderstood aspects of scalability is database size. Serious skepticism should be exercised when a vendor claims that it supports databases of a particular size. A further differentiation should be made between QA or lab tested support and production proven support. These have entirely different levels of credibility due to the variability in production requirements and practical issues that the vendor may overlook in a QA environment. Claims of support for database sizes due to design changes are especially suspect. For example, increasing storage addressability may be an important change to the vendor's design limitation, but such limits are rarely stress tested -- even in the vendor's QA labs.[5]

Reports as to which vendors support VLDB best often quote some measure of database size. In the course of this study, we were not able to confirm a single such report. When the sources were examined in detail, we found that reported database sizes were not meaningful enough to permit comparisons among products. In particular, the term database size has been found to mean any of the following:

- storage planned
- storage purchased
- storage attached to the system
- storage allocated to the DBMS

[5] Imagine the cost of testing a database of hundreds or even thousands of terabytes!

- storage allocated to the database
- storage used by the database for anything
- data, index, temporary, recovery, catalog, and process space
- data and index space
- user data and derived or summary data space
- user data space (raw data stored in DBMS format)
- raw data space (data stored in native format

The evaluator should not forget that each of these possible meanings of database size might actually pertain to multiple databases on a single platform, multiple databases on multiple platforms, a single database on multiple platforms (clusters), etc. In the course of gathering information from case studies, user experience, and surveys, we found that most large databases were supported on multiple platforms and the amount of user data was not even close to the report database size. Given data available today, it appears impossible to pronounce a given vendor as supporting terabyte databases and another as limited to hundreds of gigabytes. Indeed, there is reason to doubt that any purported single platform databases exist that support more than a terabyte of user data. See *Perception 10.*

Perception 10: Many open systems databases are in production with a terabyte (or more) of data.

> ***Reality Synopsis 10:*** Most of the space associated with terabyte (and above) databases is due to overhead including storage inflation, indexing, temporary space, log or recovery related space, and mirroring. Many sites reported as having terabyte plus databases (often as reference sites) were simply planned, or purchased storage. Often multiple databases are reported as though they were a single database.

Disk space is the primary determinant of system purchase cost, maintenance costs, and system availability. As such, the amount of disk space required to store a large amount data should be of particular concern. The purchase price for disk storage is currently estimated at about $1 million per terabyte. Maintenance costs include the cost of tape, optical, or other devices and the associated media. The time required for administrative operations (such as backup, restore, index creation, and reorganization) increase non-linearly with the size of the database and have a high impact on availability. Storage inefficiencies (due to overhead and storage format) can cause disk space requirements to expand far beyond user data requirements.

When a DBMS product is introduced as having support for large databases, its storage addressability is often quoted as a determining factor. If DBMS product A has greater storage addressability than product B, it is usually assumed to be capable of supporting larger amounts of user data as well. However, storage inefficiencies can easily outweigh any storage addressability advantages a product may have.

For example, suppose that product A not only supports twice the storage addressability of product B, but also requires more than twice the storage overhead of product B. Then clearly product B is capable of supporting larger amounts of user data than product A, and is the best value (remember the high cost of disk storage). Although some sites involved in our case studies reported differences in storage efficiency among products, we were surprised to discover how great the difference was when it was computed for a simple database. We found that the space required by Oracle 7.3 to store a fixed amount of user data was require over 1 terabyte compared to 511 gigabytes for Sybase System 11.[6] Indeed, we were unable to identify any Oracle database that would not have been implementable using Sybase, even though Sybase is deficient in storage addressability compared to Oracle. See *Perception 11*.

Perception 11: Space addressability is an indication of DBMS scalability and value.

> ***Reality Synopsis 11:*** The full storage addressability of today's DBMSs is rarely tested by vendor QA due to the expense of building (over $1 million per terabyte) and testing large databases. Practical considerations generally limit production database sizes long before the storage address limitations in the product are reached.

Understanding the sources of database storage overhead is essential to evaluating the database size aspect of scalability. Included in database storage overhead are:

- temporary space (for sorting, intermediate result sets, versioning, etc.)
- recovery guarantee space (before and after images, logs, redo journals, etc.)
- processing efficiency space (striping, extra indexes, redundancy, or summary data)
- space required for consistency (multiple versions, rollback, etc.)
- page (block) overhead
- row (record) overhead
- column (field) overhead
- lock overhead (if on disk)
- overhead associated statistics, trace, audit, etc.
- free space

[6] The computation is outlined in Appendix C and details are available on request.

- allocation management overhead
- page overflow or chaining

All of these considerations, taken together, point out one more mistaken belief, *Perception 12* below.

Perception 12: It doesn't matter how the space is used, as long as the DBMS can manage it.

> ***Reality Synopsis 12:*** For a simple (but real) database consisting of a single indexed table, the computed storage requirements for 2 billion rows was compared for Oracle and Sybase. The total mirrored space for Oracle was 1,052,447,153,398 bytes (over a terabytes) and that for Sybase was 511,337,680,618 (under half a terabyte). The products would not be so dramatically different for every database (but might be worse for some), however, the costs of storage ($0.5 million), operation, administrative times, and maintenance costs are quite real and must be considered. The largest production databases are of similar size when compared in terms of data supported, regardless of DBMS used and despite large differences in total space consumed.

When raw data is loaded into a database, the storage format of DBMS involves some overhead for each record, each page or block, and the table or file. In addition, the DBMS will store individual data fields in an encoded format. Typically, the data type of the field determines the particular encoding used and, therefore, the amount of space required. The ratio of the space required just for data (represented in its native encoding) to that required to store the data in DBMS format plus any DBMS imposed overhead is called *storage inflation.* Storage inflation can vary significantly among DBMS products, differing by a factor of as much as two or three. Thus, for example, one product might be able to store in 500 gigabytes the same amount of data that another product would store in a full terabyte <u>*simply due to storage inflation differences*</u>.

Perception 13: Data and index space support proves the ability of a DBMS to support large databases.

> ***Reality Synopsis 13:*** Temporary space (for sorting and reorganization), recovery or log space, redundancy for performance, and redundancy for availability are part of the reality of a production database. Even if all the other issues involved in supporting a large and growing amount of user data can be managed, there are a number of operational issues that are aggravated in a non-linear fashion by the growth. These include:
> - the difficulties of designing and controlling transaction isolation
> - read-only transaction management overhead (if read-consistency is required)
> - deadlock avoidance, detection, and resolution under increasing deadlock probabilities
> - allocation errors and recovery
> - space management and organizational complexity

In most cases, databases which vendors cited as "large" actually supported only a relatively small amount of user data. We found no DBMS that supported a terabyte or more of user data. Indeed, few supported more than a few hundred gigabytes of user data measured in native DBMS format. Overall, the VLDB reference sites of most products supported roughly the same amount of user data per platform, once the various amounts of storage overhead were taken into account. When comparing products, the evaluator should compare the space requirements for equal amounts of raw data. Lack of awareness of the impact of storage inflation is characterized by *Perception 13* above.

When determining if a product will support a large database, there must be sufficient storage addressability to support not only the user data given all the overhead, but also space required for operational and administrative tasks. Even if all the other issues involved in supporting a large and growing amount of user data can be managed, there are a number of operational issues that are aggravated in a non-linear fashion by the growth. These include:

- the difficulties of designing and controlling transaction isolation
- read-only transaction management overhead (if read-consistency is required)
- deadlock avoidance, detection, and resolution under increasing deadlock probabilities
- allocation errors and recovery
- space management and organizational complexity

Predicted database sizes vary as much as reported database sizes. It is difficult to determine what consensus, if any, exists within the analyst community regarding either existing or anticipated database sizes. The Meta Group correctly predicted that by 1996/1997, databases of 100GB-500GB would be common. They failed to clarify to which measure of database size this prediction referred. As is typical of the confusion among market analysts, the same group predicted that "… By 1996/97, 2TB+ will be common on SMP servers." Meta Group also defined the realm of databases approaching a terabyte of data as "ULDB" and stated that ULDB would "…remain the exclusive domain of specialized and expensive systems from Tandem and Teradata."

Gartner Group believes that the "goal" is 900 GB and 1000s of concurrent users, with some Gartner analysts predicting the need for petabyte databases. Again, there was no clarification as to which measure of database size the prediction referred. Gartner's Radcliffe (reference 26, August, 1996) "… Database sizes of up to 100 Gbytes now exist, but we would expect manageability problems beyond 50 Gbytes." Radcliffe (reference 13) defines VLDB as greater than 1000 users and greater than 200 GB. In 1996, Radcliffe once again redefines VLDB. This time (in Figure 1, reference 26) he defines *VLDB for OLTP as more than 1,500 concurrent users and more than 300 Gbytes database size* (not necessarily data). Jonathan Block (reference 22) along with Donald Feinberg, takes a much more aggressive view of the rise of VLDB: "By 2000, Gartner Group expects database sizes to approach *1,000 terabytes* (1 petabyte)."

In one sense, these predictions have become, and are becoming, true. However, whether databases are in the 200 – 2TB+ range depends upon what one considers to be the actual size of the database. As noted above, how one measures database size, and which DBMS product is being used, is crucial to an understanding of the market.

V. Evaluating Numbers of Users Supported

Scalability is also often measured by appealing to the numbers of users that a DBMS will support. Unfortunately, reported numbers of users can mean any of the following:

- *licensed users* - the maximum number of users the licensed configuration will legally allow
- *identified users* - the number of unique user identifiers that have been assigned in the database
- *user community* - the number of individuals that access the database at any time, through any means
- *connected users* - the number of simultaneously connected users, whether or not the DBMS is doing work on their behalf
- *concurrent users* - the number of users for which the DBMS is simultaneously doing work
- *indirect users* - the number of users that are indirectly connected to the DBMS (perhaps via an application server, TP monitor, or other means of multiplexing
- *concurrent transactions* - the number of simultaneously executing transactions, irrespective of whether multiple transactions are being processed on behalf of the same user or not

When comparing DBMS products with respect to the numbers of users supported, care needs to be given to make certain that the meaning of "user" is the same for all of the products. This is especially important when evaluating reference account or anecdotal data. It is not meaningful (as well as misleading and unfair) to compare systems in terms of connection scalability in which one uses a TP monitor and another does not. Ideally, both the number of concurrent transactions and the number of concurrent users will be identified, although this is difficult information to obtain in practice. This study found that the difference between concurrent users and connected users could differ by as much as a factor of 100 (*Perception 14*).

When evaluating a DBMS product's native capabilities with respect to *user scalability*, issues such as connection overhead, concurrent transaction overhead, and limitations on the numbers of identified users, concurrent users, connected users, and concurrent transactions all need to be taken into account. Although a DBMS may in some sense be "scalable", the cost per user in terms of memory or connect and disconnect times, and the difficulty in managing required resources and authorizations will determine its practical value. In addition, not all users consume equal resources, even if they are concurrent and executing similar transactions.

Perception 14: Numbers of users supported is a measure of scalability.

> ***Reality Synopsis 14:*** Reported numbers of users at reference and survey sites vary greatly in meaning. These include concurrent transactions, indirect users (via multiplexing), concurrent users, connected users, size of the user community, identified users, and licensed users. Of these, concurrent users and concurrent transactions are probably the most useful, and rarely exceed a few thousand. Also, the costs associated exclusively with connection overhead need to be taken into account. With a small connection overhead of 75KB per user, ten thousand (10,000) users require 750 MB of memory. With a more common connection overhead of 150KB, it becomes 1.5 GB.

Even a simplistic model of the cost of user connections demonstrates how important the memory cost per connection can be when today's large user communities are considered (see the 1997 VLDB Survey, Winter Corporation, www.wintercorp.com). Predicted numbers of user connections will push the total cost up even higher. With a small connection overhead of 75KB per user, ten thousand (10,000) users require 750 MB of memory. Forty thousand (40,000) users require 3 GB! With a more common connection overhead of 150KB, these numbers become 1.5 GB and 6 GB, respectively.

As with database size, understanding of both the actual and predicted numbers of users is confused and confusing. Analysts and researchers typically do not define what they mean by "users." There is, however, much more consensus on the absolute numbers. The Meta Group predicted that, by 1996/1997, although user counts would typically not grow beyond the current 200 or 300, a few sites would grow to roughly 1,000 users. They further predicted that TP monitors and distributed environments would dominant the region above 500GB and 1000 users. Gartner's Radcliffe (reference 26, August, 1996) "...largest user populations now surpass 1,000 connected users, usually in a two-tier C/S architecture, although very few systems have more than 400 concurrent users. …" In fact, the case studies supporting this report clearly showed that numbers of users reported by vendors for reference sites were almost exclusively "connected users" while customers were most often concerned about concurrent transactions.

VI. Database Partitioning

This study found that almost all databases in the 200 GB – 2TB+ realm have been partitioned. There are a variety of techniques used for partitioning. Contrary to common belief, partitioning is *not* implemented exclusively by dividing a logical database into multiple physical databases and attempting to provide some degree of transparency through application code. Among the techniques used to partition a database are:

- database partitions (dbchunks, partitions, tablespaces)
- multiple databases (units of recovery, name spaces, unit of transaction management, unit of connection, instances)

- multiple software servers (unit of connection, collection of databases)
- multiple hardware servers (clustering, shared nothing, federated, distributed, disconnected)

Whatever the technique for partitioning a database, some form of cohesiveness has been necessary. Cohesiveness can be provided by one or more of the following:

- cache coherency (there are many schemes, supporting varying degrees of transparency)
- shared disk
- shared memory
- distributed transactions
- design techniques (avoiding cross partition transactions or requests)
- partial backup and restore
- replication (asynchronous, copy management)
- application code
- TP monitor (commercial or custom)

The reasons for partitioning a database are far from obvious. Some analysts have simply assumed that databases are partitioned because the DBMS will not support a database of the needed size. They have characterized partitioning as a "workaround" for a lack of scalability in a centralized system, and selected certain methods of cohesion (such as distributed or replicated database strategies) as "artificial." Our case studies and surveys have shown that users have other, more important reasons, for partitioning. Indeed, no amount of technology in a centralized system, including parallel query and data partitioning) would meet these user's needs.

There are many reasons that users chose to partition databases. Surprisingly, we found that scalability was among the least of their concerns, with storage addressability and performance being two scalability reasons actually cited. The following is a sampling of other reasons that were more important to the user:

- Platform limitations preclude DBMS scalability (for example, Solaris has a 2GB file limitation).
- Additional partitioned systems can be added online with partitions more loosely coupled than in an integrated database.

- The impact on the business is minimized in the event of failures (reliabilty).
- Individual partitions correspond to distinct aspects of business processing.
- Individual partitions correspond to distinct projects.
- Existing stovepipe applications dictate that partitions correspond to business and political divisions.
- Faster, more granular and independent backup and recovery.
- Partitioning improves batch load restartability.
- Most joins occur within a partition, thereby reducing join cost.
- A partitioned database, with its own catalog, recovery facilities, and space management is easier to manage operationally as a unit.

All of the above denies the widely held *Perception 15*:

Perception 15: Databases are partitioned primarily in order to circumvent scalability limitations of particular products.

> ***Reality Synopsis 15:*** Case studies show that most ***large*** databases are partitioned***, regardless of the DBMS used.*** Surprisingly, we found that scalability was among the least of their concerns, with storage addressability and performance being two scalability reasons actually cited. Among the more important reasons were:
> - Platform limitations preclude DBMS scalability (e.g., 2 GB file limitations).
> - Additional partitioned systems can be added online.
> - The impact on the business is minimized in the event of failures (reliabilty).
> - Individual partitions correspond to distinct aspects of business processing.
> - Individual partitions correspond to distinct projects.
> - Existing stovepipe applications dictate that partitions correspond to business and political divisions.

Analysts claim that the methods of partitioning add unnecessary complexity, especially that "…data reorganization would cause significant disruption, if required" (reference 14). By contrast with database partitioning, table partitioning and shared nothing implementations have been treated as good, scalable solutions. In fact, this study found that difficulties with data reorganization in a table partitioned environment is one of the most significant reasons that shared nothing implementations fail to scale! Reorganization of table partitioning schemes is not only difficult and disrupts table access, but there are few guidelines for redesign.

By contrast with table partitioning and a shared nothing implementation, a partitioned database with replicated data as needed for consistency permits each replica to be reorganized as needed and independently of other replicas. In most cases, we found that replication is used because its loose consistency closely mirrors the business processes. These business processes are themselves tolerant of short term inconsistencies among the corresponding portions of the business, each portion modeled in part by a replica. Another important reason that replication is used is to integrate systems that could not otherwise be integrated, since they require a high degree of independence (for security or administrative reasons, for example).

Perception 16: Replicated databases are more difficult to reorganize than those supporting table partitioning.

> ***Reality Synopsis 16:*** Table partitioning and shared nothing implementations have been treated as good, scalable solutions. In fact, difficulties with data reorganization in a table partitioned environment are one of the most significant reasons that shared nothing implementations fail to scale. Reorganization of table partitioning schemes is not only difficult and disrupts table access, but there are few guidelines for redesign. By contrast, the loose coupling afforded by asynchronous replication permits relatively independent and non-disruptive reorganization.

Both Oracle and Sybase sites often partition the total problem set into a number of physical databases. If the degree of data coupling among these databases is moderate to low, these physical databases can belong to distinct physical databases on separate platforms, and users often use asynchronous replication to keep these databases in synch. Asynchronous replication technology was introduced first by Sybase, and later by Oracle and others. The ability to partition a logical database across multiple physical databases is used with asynchronous replication to address the differing storage, backup, and restore requirements of portions of the database, as well as for business reasons (*Perception 17).*

Perception 17: Asynchronous replication is used to counterbalance scalability limitations.

> ***Reality Synopsis 17:*** Asynchronous replication is used to provide cohesiveness and loose consistency among application systems that cannot be tightly integrated. Although some sites have mistakenly attempted to use replication to counterbalance scalability limitation of DBMS products, most sites learned very quickly that this is an inappropriate use of the technology and does not work.

Put another way, common update and administrative characteristics often dictate the partitioning of a database. For example, some data is read-only (such as a reference database), while other data is highly volatile (such as a transaction). This characteristic affects not only storage allocation requirements, but also how often backups are needed. It also affects how long backup and restore operations will take, as does the size of the partition. Addressing such issues in physical design are not simply workarounds for DBMS deficiencies (*Perception 18*), but involve the use of DBMS features and functionality to best meet business requirements.

Perception 18: Using multiple databases and/or servers and replication are workarounds for DBMS deficiencies.

> ***Reality Synopsis 18:*** Customers use multiple databases and/or servers and replications as particular methods to partition a database and maintain cohesiveness, most often because it fits the business model. They also use a number of other techniques to achieve the same end. For most customers pushing database limitations, a single, integrated database is impractical and would not meet business requirements even if the DBMS could support it.

If the degree of data coupling is moderate to high, these physical databases are often maintained on the same server platform. When run on an SMP platform, users can sometimes use Oracle's Parallel Server Option (OPS) to achieve a solution similar to that achieved with replication, albeit for slightly different problems, and one which requires strict attention to application partitioning. In particular, such "clustering" solutions are used primarily for high availability rather than scalability.

Perception 19: Clustering is an important scalability solution.

> ***Reality Synopsis 19:*** DBMS clustering solutions primarily provide, and are used for, high availability rather than for scalability. Due to limitations in each of the various DBMS clustering solutions, designers must exercise great care to obtain even moderate scaleup or speedup from cross-node cluster resources. These considerations cause a clustered database to be designed more like a federation of loosely coupled physical databases than like a single integrated database.

VII. Oracle

As everyone now knows, Oracle today is the RDBMS market leader. The company leads its competition by a wide margin, not only in terms of market share, but also revenues. Over the course of some ten years, it has continuously added features and functionality to its product and has marketed these features well. The current product (Oracle8) is purported to support thousands of users (perhaps tens of thousands), is designed for petabyte storage addressability, and lays claim to record setting benchmark transaction rates. Oracle makes a version of its product available on many platforms, from the desktop to the mainframe.

A Bit of History

In the late 1980's, Oracle became the consummate combatant in the so-called "benchmark wars." By publishing unexplained benchmark results in advertisements and claiming that competitors were incapable of matching their performance results, Oracle succeeded in getting other vendors to play "follow the leader." This strategy continued into 1996 with full page advertisements and numerous white papers. As SMP support became important, TPC benchmark results were promoted as proof that Oracle was "scalable." While TPC benchmarks do contain information relevant to scalability, most of this information was neither published nor used in Oracle's positioning on scalability.

Prior to Version 6, Oracle implemented only table level and database locking. This strictly limited the number of concurrent users. With the introduction of row-level locking and multi-version read in Version 6, Oracle began promoting these as solutions to scalability. However, these solutions were of limited value on SMP platforms. Oracle addressed this problem with its cluster solution, OPS (the Oracle Parallel Server). With this move, Oracle entered (and somewhat created) the processor scalability wars.

Historically, it is interesting to note the concurrence of client/server database computing with Oracle's introduction of row-level locking. Oracle applications have traditionally been written to perform row-level, cursor-driven operations, similar to applications one might find on the mainframe. Such applications encounter unacceptable contention in a distributed environment (such as client/server) and require the row-level locking to support reasonable levels of concurrency. This fact may have been the strongest force driving the implementation of row-level locking in Oracle. Most pre-relational applications were written with the assumption of sequential record manipulation and record ownership. They are thus more readily re-deployed on an RDBMS that provides sequential row-level manipulation and row-level locking. It is to Oracle's credit that they were among the first to recognize the need for row-level locking with row-at-a-time processing, and to meet it.

Oracle also recognized that existing applications (and application developers) were familiar with application managed transaction isolation. As a result, RDBMS enforced serializability was unnecessary. Likewise, administrative functions depended on familiar file-level operating system utilities and facilities. The products process-based architecture required less operating system integration (by Oracle) and less learning by administrators and developers unfamiliar with multi-threading.

Oracle originally promoted MPP (massively parallel processor) platforms, such as N-Cube, as a solution for VLDB and high-volume OLTP. Oracle's implementation in a multi-processor environment involved few, if any, significant changes to its engine. This was accomplished by starting up a separate "instance" (set of processes) for each processor or node. The need to provide multiple processors with shared access to each instance's memory resulted in the need to solve the "cache coherency" problem. Conceptually, cache coherency is provided by tagging data blocks with transaction identifiers. A list of active transactions is maintained for each instance. Blocks owned by one instance are then transferred to another instance on request. This is "easily" transformed into either a shared disk or "cluster" solution (by transferring block ownership via shared disk) or into an MPP solution (by transferring block ownership via the processor interconnect).

As SMP platforms became more important, Oracle promoted its support of these platforms as a "scalable" DBMS solution. Because Oracle has a multi-processing based architecture and defers to the operating system for scheduling, it can readily to take advantage of multiple processors on an SMP platform (albeit with more overhead than a multi-threaded based architecture). From both it's MPP and it's cluster solution, Oracle began to obtain larger TPC benchmark numbers than competitors. These numbers were quickly associated with multiple processor support by Oracle, press, and analysts.

Most of the positioning was that Oracle could support more processors and provide larger transaction rates with each additional processor. Oracle did not identify the percentage scalability or percentage speedup, nor did it identify the amount of disk and memory resources that were added to support additional instances. (In fairness, neither did any other vendor.) This is an example of the misleading results that are obtained when "scalability" is identified with absolute numbers or when the range of scalability and cost are not taken into account. The analyst community identified (but has not analyzed) the impact of the starting point for this "scalability" being a lower level of performance and higher resource usage than that of competitors.

Recently, Oracle has stopped pushing scalability per se, although design support for database partitioning, larger databases, and larger numbers of connections are in Oracle 8. Oracle has won the scalability battle on the marketing front: the analyst community continues to tout Oracle as "scalable" and most decision makers have accepted this analysis. In positioning against competitors, Oracle points out that the product runs on systems with large numbers of processors and that they have a cluster solution while competitors do not. However, as noted by Gartner Group's Radcliffe (reference 12) "...it requires great care in placing the workload on the cluster. ..."

Oracle claims desktop to mainframe scalability. This claim is not entirely unbelieveable: Oracle presents its workgroup, SMP, cluster, and mainframe products as though they were identical. This capability might be called *platform scalability* were the products on each platform the same (they are not). The fact that Oracle does have an MVS version, while its main competitors do not, does make the argument more convincing. To date, we have been unable to find an Oracle for MVS site with mainframe class DBMS characteristics (in terms of database size, performance, or numbers of concurrent users).

Much to its credit, Oracle has long supported a number of key features that help promote administrative scalability, including backup and restore below the database level. Though care must be taken with some Oracle claims (for example, its "online backup" is available only for partial backup and row level locking depends on data block design[7]), the average Oracle customer obtains considerable value from the product.

[7] The number of locks pre-allocated to each block at table creation time (INITRANS and MAXTRANS) determines the number of concurrent transactions that can share ownership of the block.

User Experience

Despite claims to the contrary (and analyst acceptance of those claims), referenceable Oracle sites supporting more than a few hundred gigabytes of raw data are hard to find. There are several reasons for this. First, Oracle has tight account control in place regarding VLDB sites. Even key Oracle partners apparently cannot circumvent this mechanism: when VLDB sites were approached by the Oracle partners for this study, Oracle account managers immediately halted the investigation even though it was approved by an Oracle VP and the customer had no objection. Lists of customers obtained from Oracle partners depicted a large number of customers in the over 500 GB total space range. Preliminary investigations by telephone and electronic mail differentiated the actual size from the sizes depicted. There are a tremendous number of VLDB sites which are currently being populated and have been designed to accommodate over 500 GB of storage. Some of the largest Oracle sites have been within the telecommunications industry.

The end result of this initial investigation yielded some important data. First, most of the Oracle sites that are represented as “VLDB” are works in progress and not fully populated with data. Second, most Oracle VLDB sites are in the data warehouse / decision support arena and are therefore read-intensive. Third, end users are not eager to disclose the status of their projects (whether successes or failures) while in the “data populating” process. Fourth, most VLDB sites support amounts of user data similar to that of the largest Sybase sites found.

Most recently Oracle has helped make contact with Lucent Technologies. Lucent Technologies is cooperating with the study. They are expected to provide sites for the study in the near future. In addition, Winter Corporation is collaborating with Alternative Technologies, and has provided information collected through Winter Corporation’s VLDB Survey. Additional case studies are expected to develop based on this information.

Oracle characteristics sited by Gartner include few Oracle7 databases with over 500 concurrent users. These exceptions are based on various architectures and reach up to 1,800 concurrent users.” “...benchmark claims of up to 100 Gbytes per hour, using third-party products such as Epoch and Legato, from Oracle’s SMTI program.” “Challenges ...[include] greater than 1,000 users, greater than 200 Gbytes...”

Gartner cites four Oracle references for large numbers of users, although only one might be characterized as VLDB. These include a U.K. insurance company supporting 600 concurrent users and an 80 GB database on CS Pyramid, a pan-European car rental with peak load of 1800 users and an 80 GB database on a Sequent cluster, and a Midwest mail order with 600 users and a 300 GB database on SPARCsystem. All are OLTP mission critical applications.

Preliminary Studies: Summary Results

To the present, preliminary investigations of two telecommunications sites, two retail sites, and two financial institutions have completed. Of these, only one has been willing to consider participating in a case study. Details, to the degree that they can be disclosed, appear in Appendix A.

The following summarizes the key conclusions that have resulted from the study so far:

- ***database size*** - Only one site located to date confirmed an working application (data warehouse) with an allocated database size of over a terabyte. The raw data supported and the platforms being used could not be confirmed. The allocated database size of two other sites (also data warehouses) were reputed to be over a terabyte, but have not been confirmed. Two other sites support data warehouses with allocated database sizes in the 700 gigabytes range. One of these supports raw data in database format of approximately 400 gigabytes corresponding to 200-300 gigabytes of raw data in native format. Initial reports of database size were actually planned allocated space.

- ***platforms*** - None of the sites confirmed a single platform database. Most large database or high concurrency sites were confirmed to be clusters or partitioned databases.

- ***database partitioning*** - Several sites that ran on clusters reported that they had partitioned their databases to minimize "pinging." Others pointed out that they had partitioned the database so that it ran under multiple instances to minimize administration costs (for example, backup, restore, and index times).

- ***storage inflation*** – The total storage inflation ratio (total allocation to raw data) may be as high as 10-15 when indexes, temporary storage, recovery space, and mirroring are included.

- ***clusters*** - Sites using clusters reported that they did so primarily to improve availability and only secondarily for cross-node access. Otherwise, the cluster database might have been divided among multiple platforms and databases, and would not have been as large.

- ***database type*** - Most databases over 400 gigabytes in allocated size support read-intensive applications. Most OLTP databases are under 300 gigabytes in allocated size.

- ***administration*** - Multiple sites reported that they were having difficulties scaling the database into the near terabyte range or beyond. Problems included the time required to perform administrative functions, the amount of storage required as the database grew, and robustness concerns as the database grew (for example, extent allocation errors, insufficient support for large numbers of files, and index build errors).

- ***consulting support*** - Two of the largest sites reported that they were extremely dependent on free consulting support from Oracle, without which they would not have continued with the product.

- ***scalability*** - Although all the sites investigated considered scalability of the product important, they had not qualitatively experienced a high degree of processor scalability or speedup. Furthermore, those that had experienced extensive database growth had found that considerable effort on the physical database design (sometimes complete re-design) was required to maintain acceptable performance.

- *common problems seen*

1. Poor database design, leading to wasted space and relatively inefficient processing
2. Poor process design, leading to transaction rates and response times that could easily be improved.
3. Non-relational approach to problems.
4. Difficulties with administration as the database grew.
5. Lack of system architecture and design.

- *common successes*
  1. Rapid application implementation.
  2. Easy process migration from legacy systems.
  3. Easy integration with non-DBMS systems and third party tools.

Benchmark Data

An analysis of TPC Benchmark C full disclosure reports and result summaries disclosed several important facts relevant to Oracle scalability:

- Oracle holds the record for the largest number of processors used in a TPC-C benchmark with forty-eight (48).

- Processor scalability is apparently less than 70% -- even below sixteen (16) processors. This number is seriously degraded in a cluster configuration.

- The only TPC-C benchmark result from which large processor scalability could be inferred gives a very poor 30% speedup (HP, one processor “scaling” to forty-eight processors in a cluster).

- The overhead in terms of required memory and disk space is high compared to some competitors.

When TPC benchmark results are quoted in Oracle advertisements as evidence of scalability, they are correctly taken from the same “platform”, but are not really comparable since the number of CPUs, the clock rates, or even the OS versions may differ. In at least one case, a higher benchmark number by a competitor on identically the same platform was simply ignored by Oracle and a lower number on a lower clock rate system was quoted instead with a claim of better performance.

To Oracle’s credit, upper management within the company cooperated fully with this investigation. However, they were largely unsuccessful at being able to supply customer contacts despite using the company’s VLDB customer lists. In general, account managers and the sales force were uncooperative even when customers were clearly eager to participate. Information cited here in case studies is primarily from other sources, including that obtained through Alternative Technologies’ clients, publicly available case studies and benchmark data, and participants in Winter Corporation’s 1997 VLDB Survey.

VIII. Sybase

Sybase has gone from the number two position in RDBMS market leadership to number three or four in terms of market share and revenues. From its inception until approximately four years ago, the company continuously introduced important new technology for distributed computing. Its support for key administrative scalability features included online backup and better index management than most companies. Initially it provided competitive support for processor scalability. A lag in support was addressed with System 11. It is weak in platform scalability and has serious limitations on storage addressability (for which it compensates through storage efficiency). High transaction rates, low connection overhead, a multi-threaded architecture, strong integration and connectivity have been strong arguments for the product. With recent announcements from competitors, the company must work hard to keep the product competitive. It has never been able to capitalize on its industry "firsts". Its current financial position has lead many decision makers to avoid selecting the product.

A Bit of History

Sybase was the first RDBMS product designed specifically for SMP environments: Version 4.8 introduced VSA (the virtual server architecture). The product was also the first to have a multi-threaded based architecture. Although the product had limitations (uneven load balancing, bottlenecks, and single-point vulnerability), it did provide low overhead processor scalability over a short range. Some introductory marketing of the product was done, but it was ineffective and there was little follow-up. Part of this was due to the fact that 4.8 was built on an old code baseline. Users were forced to choose between old functionality and SMP support. In addition, Sybase was unable to convey the value of the architecture to users and analysts. Sybase presented VSA as an option rather than significant architectural innovation. [8]

With the introduction of 4.9, Sybase finally integrated VSA with its baseline code. Although this removed functionality issues for users, it did not address product limitations. Sybase begin to feel the pinch of the scalability wars, especially from Oracle's TPC numbers on clusters. It began marketing its TPC numbers as representing scalable performance. Unfortunately, this left it wide open for attacks on VSA bottlenecks, ultimately resulting in Sybase being characterized as being unable to take advantage of more than four processors. Sybase seems to have made no attempt at a clear public statement regarding this issue, instead choosing to attack the problem (however slowly) in engineering.

Sybase failed to introduce and maintain a cluster capability. As early as 1990, however, Sybase provided a warm standby high availability solution with Companion Server. It also took severe criticism during the 1990s for its failure to introduce row-level locking. This led to poor scalability in some "benchmarks" with applications written for row-level locking, but did not generally affect benchmark performance results. Indeed, applications written to perform set processing operations have demonstrated very competitive benchmark results with reasonable

[8] By contrast, note the much more recent introduction by Informix of Informix OnLine DSA (Dynamic Server Architecture), which seems to have deliberately mimicked VSA (even in name), and made a significant point of the architecture by using DSA in the product name.

levels of concurrency even though page-level locking was used. Likewise, Sybase has ultimately done well with respect to TPC Benchmark C results. Although Sybase played in the TPC benchmark wars, its advertising and marketing campaigns were not nearly as effective as those of Oracle in this area.

With System 10, the Sybase scalability barrier was moved in the minds of analysts to six processors. However, this improvement was lost in the mire of System 10 bugs and support problems. This caused Sybase to lose the second battle in the scalability wars, leaving Informix an opportunity to catch up and placing Sybase as third on analysts' lists for DBMS scalability. Although Sybase made a valiant effort to meet non-scalability criticisms with the introduction of System 11, the message was seriously degraded by Sybase’s fragmented DBMS market strategy. This strategy, which attempted to justify the lack of an integrated DBMS by claiming that special markets require specialized products, only highlighted the fact that users could not obtain advertised Sybase functionality “out of the box” (integration was required) and that Sybase itself was very fragmented.

Analysts were critical of both the company and product line weaknesses, ignoring the scalability improvements made by Sybase with System 11. Sybase failed to deliver the System 11 scalability message, even though a demonstration of operation on 20 processors was given in the System 11 introduction. In particular, Sybase has never managed to deliver supporting details. Some advertisements were used to dispute the Oracle TPC advertisements, but these were not strong enough and were not continued. Sybase does not appear to make analyst presentations that deliver the kind of powerful scalability message that competitors deliver.

Sybase did not learn early enough how to deliver product functionality that has high marketing value. For example, Sybase has long ignored the market value of row level locking, clustering, relatively unlimited storage addressability, and the need for backup and recovery below the database level.

User Experience

As with Oracle, referenceable Sybase sites supporting more than a few hundred gigabytes of user data are hard to find. There are several reasons for this. First, Sybase customers often consider themselves to be pursuing “leading edge” solutions and do not wish to share this information with competitors. Second, the Sybase solution as implemented at such sites is not obviously compatible with the traditional understanding of a database (although several large single database sites were located).

Customers with applications supporting several hundreds of gigabytes (total database size) in Sybase databases were not unusual. For reasons apparently related to the methods that Sybase uses to store and manage data, Sybase customers are generally aware of the difference between allocated and used data space, and are often aware of the amount of space used for user data. The sites were surprisingly large in terms of the amount of raw data supported. Some of the largest Sybase sites have been within the telecommunications and financial industries.

To Sybase's credit, the company cooperated fully with this investigation and was able to supply the customer contacts cited here in case studies. Other information was obtained through Alternative Technologies clients.

The end result of this investigation yielded some important data. First, most of the "large" Sybase sites are still growing, but have been operational for some time. Second, large sites are split more or less fifty-fifty between OLTP and data warehouse / decision support applications. Third, the amount of raw data represented was relatively high and overhead relatively low. Fourth, very few sites (including all sources of information) were encroaching scalability ceiling or were restrained by any database scalability limitations.

Gartner Group cited its own awareness of Sybase accounts. J. Radcliffe (reference 14) stated that "... Few sites have more than 400 concurrent users on a single SQL Server and the maximum is about 800 concurrent users." Included are a credit card company supported 800 concurrent users but with a relatively low transaction rate of 10 tps. They also referred to database applications with 30 and 80 GB of data. These are small compared to the sites examined in this study.

To paraphrase a Gartner analyst, with a process-based architecture, users are less likely to hit a scalability ceiling than thread-based, multithreaded architectures; but, conversely, users are more likely to need that scalability insurance.

Case Studies: Summary Results

To the present, investigations of three telecommunications sites, two insurance sites (one health), one clinical laboratory, and one utility site have completed. A health insurance company is not discussed here because of the small amount of information that we were able to obtain.[9] A number of others declined to be included in the study, either because of the time commitment or because they feared disclosure of proprietary and confidential information. Details of the case studies, to the degree that they can be disclosed, appear in Appendix B.

The following summarizes the key conclusions that have resulted from the study so far:

- ***database size*** – Database sizes included 204 (single database), 300 (multiple), 300 (multiple), 600 (multiple), 360-480 (single database), 100 (single database), 180 (single database), and 20 (single database). These values included user data, summary data, and indexes, but excluded free space, log space, and temporary space.

- ***platforms*** - None of the sites investigated used a cluster. However, sites using more than one database on a single platform were not uncommon. Three sites used a custom TP monitor to divide the database across multiple platforms.

- ***database partitioning*** - With the exception of four databases, all others had partitioned the application data so that it resided in multiple databases. Although this had the effect of minimizing administration costs (for example, backup, restore, update statistics, integrity

[9] We did confirm that it supported a 180 GB database for the site.

checks, and index times), we were surprised to discover that the primary reasons were business related. One of the telecommunications sites required high platform reliability while minimizing the possible risk to the business overall (they were experiencing one platform failure about every ten days). As a result, application data was divided into fifty-gigabyte segments per database.

- ***storage inflation*** – The total storage inflation ratio (total allocation to raw data) may be as high as 5-7 when indexes, temporary storage, recovery space, and mirroring are included.

- ***clusters*** - Sites stated that they desired cluster support from Sybase to improve availability, but not scalability.

- ***database type*** - Most databases over 200 gigabytes in allocated size support read-intensive applications. All OLTP databases are under 510 gigabytes in allocated size, though some were close.

- ***administration*** – More than one site reported that they were having difficulties scaling the database into the one-half terabyte range or beyond. The primary problem was the time required to perform administrative functions. The addressability limitations of a single database server to 512 gigabytes on 32-bit platforms (this expands to 4 terabytes on 64-bit platforms) were considered serious if encountered (only one site had).

- ***loyalty*** - Most sites were committed to Sybase. Several sites (and many customers that chose not to participate in the study) were concerned about the financial viability of the company. Three sites were actively involved in proving that the product could handle the next generation of growth.

- ***scalability*** - Sites using System 10 were not overly impressed with processor scalability, but were generally pleased with performance relative to other products they had tried. By contrast, those that had used or tested System 11 were very impressed with processor scalability.

- *common problems seen*
  1. Poor database design, leading to wasted space and relatively inefficient processing
  2. Poor process design, leading to transaction rates and response times that could easily be improved.
  3. Inefficient or inadequate use of the vendor’s tools.

- *common successes*
  1. Relatively easy data and process migration from legacy systems.
  2. Easy to manage database partitions (once set up) and replication.
  3. Powerful customization of connectivity and integration services (Open Server based).

Benchmark Data

An analysis of TPC Benchmark C full disclosure reports and result summaries disclosed several important facts relevant to Sybase scalability:

- Processor scalability is apparently better than 70%.
- Sybase performance on a single SMP platform was more efficient than Oracle on a 4-way cluster.
- The overhead in terms of required memory and disk space is low compared to some competitors.

IX. TPC Comparison

Oracle and Sybase TPC-C full disclosures were obtained for both the DEC Alpha 8400/300 and 8400/350. These have been analyzed and are compared in Table I and Table II below. Both show that Sybase System 11 outperforms Oracle 7.3 and at a lower cost. In addition, System 11 uses less memory and less disk space, and both data space and log space grow at a slower rate that it does for Oracle 7.3. This fact represents lower costs for database physical design, implementation, and maintenance for Sybase than Oracle.

TABLE I

Alpha 8400 5/350	Sybase 11	Oracle 7.3
Number of CPUs	10	8
TPM	14,176.61	11,456.13
$ / TPM	198.37	286
memory	6 GB (2.6 for server cache?)	8 GB
concurrent users	290	600
simulated (TPC) users	9,100	9.,500
drives	146 / 214	161 / 195
RAID	9	10
client memory	128 MB	512 MB
total cost	$2.812 million	$3.27 million
total disk	1,033.4 GB	1,176.6 GB
dynamic	24,038,165 KB	23,604,774 KB
static	69,929,989 KB	87,711,202 KB
growth	4,543,729 KB	4,537,484 KB
180 day	846.67 GB	862.56 GB
log/TPM	3.86 KB	13.37 KB

Beyond these scalability issues, Oracle enforces the required serializable transaction isolation through application code with detection in the DBMS, whereas Sybase enforces it centrally in the DBMS. The Oracle approach places a burden on the developer and means that a different mix of transactions might fail the isolation tests or corrupt the database. We are eager to see comparable TPC benchmarks for Oracle8 and Sybase Adaptive Server 11.5.

NOTES:. Oracle reports storage requirements in 2 KB blocks, while Sybase reports 1 KB usage. Because this confused the comparison, the numbers were normalized to 1 KB units here.

TABLE II

Alpha 8400 5/300	Sybase	Oracle
Number and type of CPUs	10	8
TPM	11,014.1	9,414.06
$/TPM	222.02	316
memory	6 GB (2.6 for server cache?)	8 GB
concurrent users	200	400 sessions
terminals (TPC users)	9,100	8,000
drives	190 / 84	161 / 132
RAID	8	9
client memory	128 MB	128 MB
total cost	$2.42 million	$2.972 million
total disk	735 GB	866.1 GB
dynamic	19,110,380 KB	23,764,674 KB
static	55,554,121 KB	87,711,202 KB
growth	3,530,092 KB	3,728,714 KB
180 day	658.96	723.72
log/TPM	3.832 KB	13.37 KB

Acknowledgements

We wish to acknowledge the help of Richard Winter of Winter Corporation (Boston, MA), as well as Oracle, Sybase, and Richard Skrinde of Strategic Partners (Alameda, CA) in obtaining the data for this study.

X. References

Meta Group References

1. File 470, Oracle Jumps on Server Suite Bandwagon, 2/29/96, Meta Group
2. File 503, Databases for the Workgroup, 11/13/95, Meta Group
3. File 444, Architecting Database Infrastructures, 11/08/95, Meta Group
4. File 440, Database Processing for Scale, 9/25/95, Meta Group
5. File 419, Client/Server Performance, 6/16/95, Meta Group
6. File 400, RDBMS Selection, 3/27/95, Meta Group
7. File 372, Database Size: How Big is Big?, 11/28/94, Meta Group

Gartner Group References

8. K. Strange (The DBMS Market: A New Ranking Has Emerged, 3/8/96
9. K. Strange, The DBMS Market: A New Ranking Has Emerged, 3/8/96
10. A. Percy, The Relational DBMS Market - On Main Street, 2/27/96
11. D. Feinberg, A. Percy, Sybase - The Rocky Road Ahead, 7/6/94
12. A. Percy, D. Feinberg, Sybase: Architected for Change?, 5/17/95
13. J. Radcliffe, RDBMS Scalability, 12/28/95
14. J. Radcliffe, Evaluating RDBMS Scalability in OLTP System, 9/25/95
15. J. Radcliffe, Sybase SQL Server 10, 4/14/95
16. J. Radcliffe, RDBMS Road Map for Windows NT Clusters - Availability, 1/24/96
17. J. Radcliffe, SDM1 Q&A, 7/28/95
18. J. Radcliffe, Relational Databases: How Well Do They Scale Up?, 7/20/94
19. The Relational DBMS Market - On Main Street, 2/27/96
20. J. Radcliffe, Oracle7 - One of the "Alternative Mainframe" Leaders, 12/21/94
21. Beth Enslow, Sybase's Wake-Up Call, analyst: Donald Feinberg, 4/19/95
22. Beth Enslow, Relational Databases: Tracking the Market Leaders, 11/29/95
23. Beth Enslow, Sybase's Wake-Up Call, 4/19/95
24. Jonathan Block, The Future of Databases, 8/9/95
25. A. Percy, Oracle's Impressive 1996 Financial Results, 7/24/96
26. J. Radcliffe, RDBMS Scalability for OLTP Systems: A Status Update, 8/21/96
27. B. Burton, Sybase: Substituting Size for Focus, 7/24/96
28. B. Burton, K. Strange, Follow-Up Questions and Answers on Sybase, 8/21/96
29. J. Radcliffe, Informix: Growing Up, but Losing Focus, 7/24/96
30. B. Burton, Primary-Site Data Replication, 7/18/96
31. B. Burton, et al, The Big Five DBMS Vendor's Market Race Update, 8/12/96