A Comparison of RTI-INGRES and SYBASE

-- Revision 1.0 --

Product Marketing
July 1987

marcom doc # MBW-033S-001

A COMPARISON OF RTI-INGRES AND SYBASE

Dr. Michael Stonebraker
Computer Science Division
University of California
Berkeley, Ca.

1. INTRODUCTION

This paper discusses the differences between the SYBASE data base system and RTI-INGRES. The major points of comparison concern performance and function. It will be seen that SYBASE has a small advantage in performance for certain queries that will go away over the next 12 to 18 months. Moreover, there are serious difficulties with their product that render it inadequate in several important environments. These points are elaborated in the sections that follow.

2. PERFORMANCE

Initial tests have been run by Sun Microsystems which compared RTI-INGRES and SYBASE on a benchmark called TP1 (essentially a transaction that cashes a check). It contains 6 SQL commands, each a which affects a single record. Sun also tried a collection of other "short queries" that are frequently used in Sun/Simplify. RTI-INGRES throughput was typically 0.6 to 0.8 of the throughput of SYBASE. (Throughputs reported by SYBASE are not consistent with the ones reported by Sun, and we report the Sun numbers here.) The only cases where a greater disparity was observed was when each data base system was severely overloaded. In such situations, response time seen by end users would be totally unacceptable, and a rational user would increase the speed of the platform being used. This option is readily available to an RTI-INGRES customer but not to a SYBASE customer as discussed in the next section.

The difference in transaction throughput is caused by two factors:

1) the SYBASE server architecture
2) the SYBASE use of stored procedures

These issues are discussed in turn below.

2.1. Architecture

A server architecture is probably 20 percent more efficient that a process-per-user architecture. For this reason, RTI-INGRES has switched architectures from a process per user (Release 5) to N servers (Release 6). N may be 1 (a single server) or a larger value (for example, one server per machine in a VAXcluster or other tightly coupled architecture). The single server architecture was rejected by RTI because of the necessity of running on the VAXcluster, and a single server makes this task impossible. This is a serious weakness of the SYBASE product as discussed in the next section.

In summary, the N server approach of RTI-INGRES, Release 6 gets the 20 percent performance benefit and can utilize the VAXcluster hardware. Hence, when Release 6 is compared to SYBASE, half of the performance difference should disappear. Moreover, SYBASE will have to be changed to the RTI-INGRES architecture to support the VAXcluster.

2.2. Stored Procedures

Some DBMSs allow a user to define a collection of commands as a procedure which is stored inside the data manager. An end user simply invokes the procedure with his run-time parameters, and a result is returned at the conclusion of the execution of the commands. Since the application program running on behalf of the user must execute in a separate address space from the DBMS for protection reasons, this mechanism allows one exchange of messages between the two address spaces per procedure invocation. If TP1 is defined as a stored procedure, then each transaction can be executed with one exchange of messages. On the other hand, if each of the 6 SQL commands that make up TP1 is individually supplied to the data manager by the application program, then 6 message exchanges are required. Since SYBASE supports stored procedures while RTI-INGRES does not, it benefits from this factor of 6 in messages, thereby explaining the remaining 20 percent performance difference.

Stored procedures result in higher performance only if the user is not required to take any action inside a transaction. User actions require additional crossing between the application and the data base system which remove portions of the 20 percent noted above. If every SQL command requires a user action, then stored procedures have no benefit.

In any case, RTI-INGRES is planning to insert stored procedures that can be written in ABF/OSL into Release 7 of the product. Such procedures are substantially more powerful than the TSQL procedures available from SYBASE. Not only will the remaining 20 percent performance difference disappear, but SYBASE will be at a disadvantage in transactions that require computation since ABF/OSL contains substantially more power than TSQL.

2.3. Summary

The bottom line is that RTI-INGRES will become performance competitive with or superior to SYBASE in transaction processing by the time Release 7 is available during the second half of next year.

Moreover, RTI-INGRES and SYBASE have yet to be benchmarked on two or three way joins, or on ad-hoc queries. This is an area where RTI-INGRES is superb and SYBASE is apparently weak. We are anxious to observe the relative performance of the two products in this other important class of data base interactions.

3. FUNCTION

In this section we discuss several areas where the SYBASE product is severely lacking. We present only major difficulties rather than a much longer list of more minor items.

3.1. Support for the VAXcluster

Until SYBASE is converted to the RTI-INGRES Release 6 architecture, it cannot run on a VAXcluster. More precisely, if several application programs are running on different CPUs in a VAXcluster environment and want to access a shared data base, they can do so only by accessing a SYBASE server running on **exactly one** machine in the VAX-Cluster. Hence, application programs can reside on all machines but the server can reside only on a single machine. In the worst case, the remaining machines are idle.

Consequently, in a VAXcluster environment RTI-INGRES performance increases linearly with the number of clustered machines (because RTI-INGRES can run on each machine). SYBASE performance on a cluster is equal to its performance on a single machine. In effect, SYBASE is unusable in a VAXcluster environment until its architecture is corrected.

3.2. Platforms

At the moment, SYBASE runs only on Suns and on VAXes using VMS. They do not run in any of the following environments:

1) IBM
2) large UNIX platforms
3) personal computers

SYBASE claims to be hard at work on an IBM system; however RTI-INGRES has been available for nearly 2 years on VM/370 and will be out on MVS late this summer. Moreover, if additional throughput is required in UNIX environments an RTI customer can run on Elxsi, Sequent, Pyramid, Computer Consoles, and Arete platforms. Furthermore, an RTI-INGRES user can utilize Sun, IBM PC/RT or Apollo workstations rather than be constrained only to the Sun platforms supported by SYBASE. In all, RTI-INGRES is supported on 25 different UNIX platforms. This compares favorably with the one platform supported by SYBASE. Lastly, INGRES for PCs is probably the fastest PC product available for PC/DOS. Hence, any application written for RTI-INGRES can run on any PC with 640K of memory. Not only does SYBASE not have a PC product, but also they apparently have no immediate plans to build one. (The widely rumored arrangement with Microsoft is for an OS/2 product not a PC/DOS product.)

Consequently, if a customer is concerned about ability to run on a wide variety of platforms, SYBASE is not a good choice.

3.3. Distributed Data Bases

Although SYBASE talks about distributed data bases, they really only provide distributed access. On the other hand, RTI-INGRES supports both distributed access (through INGRES/NET) and distributed data bases (through INGRES/STAR).

To clearly distinguish the two concepts, a simple example is presented. Consider a user in New York who needs to interact with data at two locations. In particular, he requires access to the EMP relation

EMP (name, salary, age, dept, manager)

in San Francisco and the DEPT relation

DEPT (dname, floor)

in London. Such a user might want to find all the employees on the first floor which can be retrieved by running the following SQL query:

```
select name
from EMP
where dept in
     select dname
     from DEPT
     where floor = 1
```

This capability is available in the INGRES/STAR distributed data base system which allows a user to see data in multiple local data bases as a single "seamless" data base. On the other hand, SYBASE provides only distributed access, whereby a New York user can connect to London to find the first floor departments. Then he must copy them to San Francisco under program control. Lastly, he can connect to San Francisco to run the remainder of the query. This capability of remote access has been present in INGRES/NET for many years.

In summary, **remote access** allows access to a single remote data base, but joining tables in multiple data bases at different locations requires substantial programming. A **distributed DBMS** provides seamless interaction with multiple local data bases at multiple sites. SYBASE provides only remote access while RTI-INGRES provides both functions.

3.4. Application Generator

Applications for a data base system can be built either by embedding SQL statements in a conventional program (written in COBOL, PL/1, C, FORTRAN, etc.) or in a so-called fourth generation language (4GL). A 4GL typically contains a "what-you-see-is-what-you-get" screen painter, access to data base tools such as the report writer and the ad-hoc query system, and a means to specify operations in a very high level notation. Most RTI-INGRES users report an order of magnitude productivity gain when they move from embedded SQL to our 4GL (ABF).

Unfortunately only the embedded SQL level is available to a SYBASE customer, because they have no 4GL through which a user can obtain this productivity leverage. Since the total cost of any application is moving from "mostly hardware" in the 60s toward "mostly software" as machines become cheaper, this absence of function in the SYBASE product is a severe disadvantage and will result in substantially increased total application cost.

3.5. Non Standard SQL

Some users require the ability to write an application program which can be run with little change on the data managers of various vendors, (DB2, Oracle, Informix, etc.). 4GLs offer superior programmer productivity as noted above, but unfortunately, each 4GL is different. Hence, a user requiring portability must resort to embedded SQL. The programming language interface to DB2 has become a defacto standard in this area. This interface, supported by a preprocessor, contains specific commands to define cursors, open them and fetch records out of them.

Unfortunately, SYBASE has chosen a subroutine call interface that bears little resemblance to this standard. As a result, a program written for SYBASE stands no chance of vendor portability without substantial recoding. RTI-INGRES, on the other hand, has been extensively modified to conform to this standard interface and makes this transition painless.

The non-standard subroutine call interface is easy to construct but is reminiscent of the interfaces designed in the 1960s and early 1970s. Even Oracle discarded a similar interface more than 3 years ago.

A customer interested in application program portability should reject systems that do not conform to the standard.

4. SUMMARY

SYBASE has yet to prove that they can deliver quality technical support, documentation, timely maintenance releases, or a portable product that can run in many environments. Moreover, they have substantial shortcomings to correct in their product before they can match the functionality of RTI-INGRES. These problems will take SYBASE far longer to correct than it will take for RTI-INGRES to match SYBASE performance. Hence, choosing SYBASE over RTI-INGRES seems to offer little reward for the substantial risk taken.

SYBASE Product Summary

SYBASE IS AN EXPERIMENTAL SYSTEM

SYBASE has taken experimental DBMS concepts like triggers and referential integrity and implemented them in a production system without fully researching the consequences. For example, a referential integrity check involving a set (aggregate) function may be so CPU-intensive that it could bring the DBMS to its knees.

Relational Technology, like IBM, does its research in the lab on experimental systems and only introduces new capabilities into the production INGRES product after they have been proven in research protypes. In the same way that IBM is experimenting with referential integrity at its Santa Theresa software research lab, the founders of Relational Technology have been and are still investigating referential integrity, triggers, domains and user-defined datatypes, etc. in the widely-acclaimed research project called POSTGRES.

SYBASE IS AN UNPROVEN SYSTEM

No long-term production users. System and company not stress-tested. In fact, the only organizations we know of who have bought SYBASE are those who tend to dabble in new technology without need to deliver actual production systems--- note that these organizations were the first ones also to buy INGRES although that was back in 1980 - 81.

SYBASE IS WEAK IN APPLICATION DEVELOPMENT TOOLS

SYBASE forces programmers to code at the subroutine CALL-level for its DB-LIBRARY and APT-FORMS system. No 4GL at all. TRANSACT-SQL is minor, non-standard extensions to SQL. SYBASE's DataToolSet is evolving and not production-proven. Most organizations report an order of magnitude productivity gain when they move from embedded SQL to a full 4GL such as INGRES's Application-By-Forms.

SYBASE DOES NOT CONFORM TO THE ANSI OR DB2 SQL STANDARDS

ANSI and IBM have not yet adopted standards for the implementation of referential integrity, triggers, etc. And regarding SYBASE's "advanced" DBMS features--- if you use these features in SYBASE, you may well have to recode all your applications once the standards are adopted.

Relational Technology is participating in standards organizations (ANSI, ISO, MAP, RDA, etc.) and will implement features that conform to industry standards.

SYBASE IS A CLOSED, PROPRIETARY SYSTEM

SYBASE does not use the standard Sun UNIX operating system as supplied by Sun. This "hacked" UNIX kernel means the SYBASE user must allow for support issues, vendor-pointing, etc.

The SYBASE architecture also has impact on its VAXcluster capabilities. SYBASE cannot run more than one DataServer (DBMS back-end) in a cluster, while INGRES can run any number of INGRES back-ends on a VAXcluster and take full advantage of cluster processing. More precisely, if several applications are running on different CPUs in a VAXcluster and want to access a shared database, they can only do so by accessing a SYBASE server running on EXACTLY one machine in the VAXcluster.

SYBASE PERFORMANCE ADVANTAGE IS TENUOUS

TP1 performance tests run by Sun Microsystems show that INGRES 5.0's throughput was typically 0.6 to 0.8 of the throughput of SYBASE. This difference in transaction rates is due to SYBASE's server architecture and SYBASE's stored procedures capability.

INGRES 6.0 (in beta now) provides such a server architecture; INGRES 7.0 (planned for 1988) will provide production-level stored SQL procedures. In short, INGRES will become performance competitive with or superior to SYBASE in 1988. Moreover, INGRES and SYBASE have yet to be benchmarked on 2- or 3-way joins, or on ad hoc queries. This is an area where INGRES is superb and SYBASE is apparently weak.

On VAXclusters, INGRES performance increases linearly with the number of clustered machines (because INGRES can run on each machine); SYBASE performance on a VAXcluster is equal to its performance on a single machine.

SYBASE IS NOT A MATURE COMPANY--- HAS LIMITED RESOURCES

SYBASE is small (25 installations, no real production sites, < $2M revenues), may not stay around, and has to cater to customer-investors such as Apple and TRW while ignoring other customers. In fact, from what early user's are experiencing they definitely have quality problems in support and training which is not uncommon for technology start-ups.

SYBASE, due to its start-up nature, cannot provide its product for all key platforms (MVS, VM, VAX/VMS, UNIX and PC-DOS). Nor have they even commited to doing more than Sun UNIX, VAX/VMS and OS/2 (PC-DOS for 386).

And SYBASE's architecture for distributed DBMS is lacking. They claim to have DDBMS, yet all they actually have is the same architecture as INGRES (front-end/back-end) with 2 phase commit for locking. INGRES/STAR is production quality DDBMS available now--- with a full architecture for future enhancements.

SYBASE Product Summary

- An experimental system
- An unproven system
- Weak in application development tools--- no 4GL
- Does not conform to the ANSI or DB2 SQL standards
- A closed, proprietary system
- Performance advantage is tenuous and temporary
- Not a mature company--- has limited resources

Competing Against SYBASE

- SYBASE is not a "mini" version of ORACLE

- Certain current sales tactics successfully used against ORACLE may not work against SYBASE

  - Performance and technology are effectively used against ORACLE

  - These areas are being emphasized by SYBASE

Competing Against SYBASE

- Other unique INGRES selling advantages used successfully against ORACLE are just as effective against SYBASE
  - A real 4GL---SYBASE has CALL-level "C" subroutines
  - Distributed DBMS---we have architecture, product and track record
  - Proven, production-oriented RDBMS and company
- Consider SYBASE as you would ADABAS or TERADATA
  - A niche performance-oriented product

SYBASE Niche Targets

. Performance "zealots"

- Any site where performance is sole criteria at expense of everything else ("online transaction processing")

. SUN workstation "purists"

- Users who desire pure, window-oriented pre-packaged system

. "Technology research" chartered organizations

- Organizations that buy one of every new technology to test

. Advanced DBMS technology "enthusiasts "

Dealing With Performance "Zealots"

- Don't get pulled into benchmarks with SYBASE if sole evaluation criteria is performance

- Watch out for heavily networked applications where stored queries provide possible boost to SYBASE performance

- If VAX, do inform the prospect of SYBASE's inability to run DataServer on more than one node in VAXcluster, i.e. no parallel database processing

- Discuss prospect's performance requirements

  - Many times if you can get them to describe their expectations and agree that if INGRES meets this level of performance...

Dealing With Performance "Zealots"

. Remind prospect also of need for access to that data with 4GL high productivity tools, i.e. programmers are more expensive than machine cycles and disk I/Os

. Remind prospect of Relational Technology's performance commitment

- SYBASE is tuned for short transactions only; INGRES's design point is for short transactions and complex queries

- INGRES has been performance leader for 7 years; SYBASE only has a tenuous, temporary advantage that will go away soon

  -- In 1987, INGRES gets multi-server architecture

  -- In 1988 (or sooner), INGRES gets stored queries

Dealing With SUN Workstation "Purists"

. Inform prospect of joint SUN/Relational Techology development project for SIMPLIFY and what this partnership means

  - INGRES front-ends will be designed with SUN workstation human engineering in mind by SUN and INGRES experts

  - INGRES will continue to evolve as the premier SUN RDBMS product given both companies commitment to price/performance

. Inform prospect about ease of using SUNView to get same net effect today with INGRES as SYBASE's windowed front-ends

Dealing with SUN Workstation "Purists"

- Inform prospect of need to allow for UNIX support issues due to SYBASE's "hacked" UNIX kernel

- Remind prospect of probable future need to network with non-SUN systems such as IBM mainframes, other UNIX platforms, IBM PCs, etc.

Dealing With "Technology Research" Chartered Organizations

- If they are buying one of every major new technology, why haven't they bought a copy of INGRES/STAR yet?

- Introduce INGRES "futures" such as POSTGRES papers to reinforce our technology advantage and enthusiasm for triggers, object-oriented databases, abstract data types, etc.

- Remind prospect that these same features are controversial as to whether they should exist and execute in the DBMS back-end or the application front-ends

  - Many of these features are implemented in INGRES's front-ends (domains, defaults, etc.) and not in its back-end (yet)

Dealing With
Advanced DBMS Technology "Enthusiasts"

. Introduce INGRES "futures" such as POSTGRES papers to reinforce our technology advantage and enthusiasm for triggers, object-oriented databases, abstract data types, etc.

. *If necessary, point out to enthusiast's management of* need to do testing of new DBMS features in research labs such as UC Berkeley and not with your production databases

. *Remind prospect that many of these features are curio*sities that reflect the undue influence of SYBASE's few early customers

Dealing with Advanced DBMS Technology "Enthusiasts"

. Remind prospect that these same features are controversial as to whether they should exist and execute in the DBMS back-end or the application front-ends

- Many of these features are implemented in INGRES's front-ends (domains, defaults, etc.) and not in its back-end (yet)

. Remind prospect that SYBASE's claimed advanced architecture (front-ends and back-ends) has been integral to INGRES's architecture for years

π VAX version by the end of year
π

π Ship on Sun in Sept.
π OEM deal w/ [redacted] Sun
π Unseat Oracle @ Stratus
π

<u>Sybase:</u>

π Full documentation and product spec.
π 3 Key product features
  - Performance - 70 xacts/sec. = 70 appends/sec.
  - Entire backend in 35,000 lines of code -
  - Embed rules in the database
  - Non-stop transaction processing - 24 hours/day
    backups while still - 7 days/week
  - Bit-map displays

π 2nd time database users - ⓪
π Much cash - #6MM in bank - #300-400K/month -
π 55 people - 45 ~~all~~ developers, 10 ~~dev~~ others;

π SQL/Star -
π SQL/Net
π SQL/CONNECT   DB2, SQL/DS, ORACLE

π DOS/VSE version

INGRES AND THE FINANCIAL MARKET

"PREREQUISITES FOR SUCCESS"

Submitted By: Tom Lutz
District Manager:
Financial District
November 17, 1987

INGRES AND THE FINANCIAL MARKET

"PREREQUISITES TO SUCCESS!"

I. THE FINANCIAL MARKET

II. A MORE "FINANCIAL" INGRES

1. Backend Speed "TP"
2. Back-up and Recovery
3. Redundancy
4. 6.0 INGRES/Star
5. The Unix Workstation Explosion
6. Technical Support
7. Consultants
8. "Financial" Gateways
9. Vertical Marketing

III. Summary

I. THE FINANCIAL MARKET

In order for RTI to be successful in proving ourselves in the Financial Market, we need to recognize that Financial organizations have a different business environment than our more traditional client base. The differences translate predominantly into features and support procedures which INGRES and RTI do not currently address.

The Financial "Transaction Processing" environment, with its' huge databases accessed by hundreds or even thousands of users has two major needs. Performance and Availability i.e. uptime of the database, are critical issues in this market.

Performance is a major issue because high transaction rates are neccessary to give this market the ability to have up to the minute information (data) so that they can make accurate predictions and sound business decisions. Performance or speed relates directly to profitability and a competitive edge.

High Availability in all environments in which online recovery can take place is absolutely vital (especially in trading and other brokerage-type applications). Downtime costs can be staggering; especially to brokerage accounts.

If we are to actively pursue this vertical market we must take a very close look at our current offering and enhance it to meet the markets' needs. The following highlights what I personally feel needs to be accomplished in order for RTI to be a viable contender in this extremely lucrative market.

II. A MORE "FINANCIAL" INGRES

1.) Backend Speed - In order to gain a larger share of the Financial Market we need to improve on the performance of our backend. We stress what our frontends can do but in the eyes of this market, speed is the bottom-line. In short, the faster the transactions processing, the more trades booked, the more profit made by an investment bank, the better their bottom-line. We can either learn from our experiences recently at Sanford Bernstein & Morgan Stanley, or get burned again.

2.) Back-up & Recovery - If a production database or production host goes down, the user needs to be backed up ASAP. Downtime is $$$ to the Financial industry. I've heard from users that our recovery & backing out of a transactions is too time consuming. There is also a serious performance problem when one enables journaling on a database. Fast and efficient backup and recovery is essential.

3.) Redundancy - In the event of disk problems or any other problem that would make a user unable to access data, we need to provide the ability to mirror databases on all platforms. This is not an operating system problem, it's ours. Once again, a database down is money lost; especially to a brokerage house.

4.) 6.0 INGRES/Star - We need a serverized 6.0 that supports networking and has great performance now. The current timeframes for development, which are somehwhat unknown, are unacceptable if we are to be successful in this market.

5.) <u>The Unix Workstation Explosion</u> : We need to pay more attention to our product and support in this area. This is the main OS in the workstation environment and workstations are hot on Wall Street! By more attention I mean higher transactions, Rel. 6.0, online recovery, disk-shadowing, frontends that support X-Windows interface, and the other goodies we are building into our VMS product. I have heard from several Eastern Region players that UNIX Tech Support needs some beefing up. Some of the issues I’ve heard include: Novice Unix background, Novice with Work Station, and total lack of financial industry expertise. We need to provide the SUN and Apollo customers among others with improved UNIX committment & support. If not, we could be in for a long and very bumpy ride.

6.) <u>Technical Support</u> - We need to execute a plan to give different levels of technical support. We need to provide special attention to production banking/brokerage applications. Once again, downtime is money, and these firms will gladly invest in more support dollars to insure less downtime.

7.) <u>Consultants</u> - Systems and development departments for the brokerage houses are mostly made up of consultants. We need to be able to fill consulting positions with quality people. Obviously, this is an opportunity to increase revenue and insure greater success with INGRES.

8.) Financial "Gateways" : We could create a differential advantage in this market if we were to provide "Gateways" to other products that support banking and brokerage. An example would be a Rueters Gateway to collect digitized Rueters data. IDC or Telerate gateways may also want to be considered.

9.) Vertical Marketing : Presentations, demos, seminars, newsletters, brochures etc. all need to be targeted at the financial community. We need to show them that we are serious about their business and truly understand it.

SUMMARY

The Financial Market is one that could prove to be highly profitable to RTI. Because of improved performance of RDBMS, the blessing of a relational DB2, and increased price-performance of hardware, the market is beginning to look towards relational systems to handle core applications previously implemented in networked and hierarchial systems.

It's clear that both Oracle and Sybase are strategizing to address this growing need. Regardless of whether or not Oracles new TPS announcements are more marketing hype than breakthrough technology, the market is perceiving them to be moving in a positive direction. Sybase is currently recognized throughout the Financial Market as the player to beat. WE NEED TO RECOGNIZE THE IMPORTANCE OF A HIGH PERFOMRANCE OFFERING (coupled with my other suggestions) OR WE WILL MISS THE BOAT IN THE SAME WAY WE ALMOST DID WITH QUEL VS. SQL. Whether we like it or not, RDBMS' are now moving into the TP arena and we need to be there to be competitive.

The window of opportunity is very short. RTI management needs to give the suggestions outlined in this overview serious consideration immediately. Although I like to consider myself an eternal optimist, if these issues are not to be addressed ASAP, then I would suggest that we do not attack this market from a vertical standpoint and continue to address it as we do most other commercial accounts.

Thank you for your time and consideration and I look forward to discussing these "prerequisites for success" in more detail at your earliest convenience.

Bob Epstein - Bearn-Stearns financial forum (Jay Hoag) 12/7/87

π TPS -
10-50
100 users w/ <1 sec

"Sybase = the next Britton-Lee"

π Second wave of Rdbms
1st

π 2-5% of Ⓘ, Ⓞ, Inf. - are trans. proc.
95-98% are decision support -

π Current applications won't be re-written
New application

π Bear-Stearns
30% - 40% Banking
20% CIM
20% Telecomm.
25% Gov't

π Ⓞ, Ⓘ have to rewrite their s/w -

π ASP ≅ $100,000 -

π Plan @ $6 Million -
Actual @ $7 Million
} Shipments for 7 mos.

π 50% SUN
50% VAX
revenues

π <u>Tools</u> -

π IBM PC version? - not delivered

π VAX

π UNIX

<u>Server</u>

- Sun, VAX, Stratus
- VM some
- MVS is dead

π A lot of machine dependent code

π Competition at Sun

- Factory floor -
- Customer service

π Benchmarks [redacted] @ Sun

Ⓢ	6 TPS	6 TPS	6 TPS
Ⓘ	3 TPS	1.5 TPS	didn't work
	3 users	25 users	26 users

π <u>Five differentiators</u> -

- high volume performance -
- DBMS enforced data integrity -
- High availability -
- Distributed data mgt.
- Window-based tools

π "Object-oriented
↳ Stored procedures"

π K-P is funding Ontologic -

π Hambrecht is lead investor at Sybase
TRW, K-P, Charles River, OAK, Apple

- Tools -
- IBM PC version? - not delivered
- VAX
- UNIX

Server

- Sun, VAX, Stratus
- VM some
- MVS is dead

- A lot of machine dependent code
- Competition at Sun
  - Factory floor -
  - Customer service
- Benchmarks ~~[illegible]~~ @ Sun

S	6 TPS	6 TPS	6 TPS
I	3 TPS	1.5 TPS	didn't work
	3 users	25 users	26 users

- Five differentiators -
  - high volume performance -
  - DBMS enforced data integrity -
  - High availability -
  - Distributed data mgt.
  - Window-based tools

- "Object-oriented
  - "Stored procedures"
- K-P is funding Ontologic -
- Hambrecht is lead investor at Sybase
  TRW, K-P, Charles River, OAK, Apple

Strategic Relationships:

- TRW for a secure server -
- Stratus
- Pyramid
- Metaphor - (also an Oracle & B-L VAR)
- Apple

"Sybase is a database competitor, not a tools company"

Not

-DBMS enforced integrity is the

17 salesmen

FY1988 plan	FY89	FY90	FY91
was $18 MM now $22 MM	$40 MM	$68 MM	$112 MM