TMS TEXAS MEMORY SYSTEMS

The Myth of Database Independence

Mike Ault

Oracle Guru

Texas Memory Systems

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Michael R. Ault
Oracle Guru

- Nuclear Navy 6 years
- Nuclear Chemist/Programmer 10 years
- Kennedy Western University Graduate
- Bachelors Degree Computer Science
- Certified in all Oracle Versions Since 6
- Oracle DBA, author, since 1990

ORACLE CERTIFIED PROFESSIONAL

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Books by Michael R. Ault

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Introduction

- A new project starts
- Time and energy is spent on selecting the right database system
- Hardware is bought
- Specifications are developed and then the bomb from on high goes off…

"This must be database independent"

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

The Grail of Database Independence

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Database Independence

- Great from management point of view
- Build once, swap out DB as needed!
- No DB dependence
- All logic in application tier

A Texas Memory Systems Presentation

A Simile

Replace this…

With this…

A Texas Memory Systems Presentation

Comparison

- Maserati
  - Expensive
  - Hard to maintain
  - Complex
- Yugo
  - Cheap
  - Easy to maintain
  - Simple

A Texas Memory Systems Presentation

Performance

- Maserati – 0-60 5.1 seconds
- Yugo – 0-60 30 seconds

A Texas Memory Systems Presentation

But we are talking databases

- Moving code to the application tier hurts performance
- Not using features usually means more coding
- You have to code to the Least common denominators (LCD)

A Texas Memory Systems Presentation

What the heck are LCDs?

- If a feature isn't on all databases, it isn't an LCD
- For example, SQL Server, Oracle, MySQL all have mechanisms to populate a numeric key, SQL Server and MySQL have an auto increment feature, Oracle uses a sequence generator, therefore there would be no common function between the three for populating a number based key.

A Texas Memory Systems Presentation

Basic Limits

- Pure ANSI SQL
- No special structures
  - No partitioning
  - No special indexes
  - No DB specific languages

A Texas Memory Systems Presentation

Moving Application to Middle Tier

- No PL/SQL, T-SQL, etc
- No triggers – triggers work off of internal PL/SQL or T-SQL logic
- No replication – since it can't be done independent of database code
- No referential integrity – again, not all databases support this logic

A Texas Memory Systems Presentation

Moving Application to Middle Tier

- No packages, procedures, Java in the database
- No advanced queueing
- No virtual private databases
- No types, objects, methods, materialized views, summaries, dimensions, etc- not all databases support these constructs

A Texas Memory Systems Presentation

Moving Application to Middle Tier

- All internalized code in items 1-7 now must be written in external routines
- The coders must recreate the wheels of database development for each application.

A Texas Memory Systems Presentation

Use No Database Specific Feature

- Bitmap Indexes
- Bitmap join indexes
- Function based indexes
- Advanced text based indexing
- Reverse key indexes
- Hash, list, composite partitions
- No PL/SQL triggers, functions, procedures or packages

A Texas Memory Systems Presentation

Use no Database Specific Feature

- No types, methods, etc.
- Materialized views, summaries, dimensions
- No Java in the database
- No XML in the database (at least no parsing)
- Advanced security features (password checking, aging, logon restrictions, etc)
- Real Application Clusters
- Oracle standby features

A Texas Memory Systems Presentation

Use No Database Specific Language

```
CREATE OR REPLACE TRIGGER exam_pkey
BEFORE INSERT ON exam FOR EACH ROW
BEGIN
SELECT exam_seq.NEXTVAL
INTO :NEW.exam_pk
FROM DUAL;
END;
```

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Becomes (in external language)

1. **Send Select to database to get ID for row containing our owner, table, column name and last value**
2. **Implicitly lock row**
3. **Add 1 to the last value**
4. **Insert row with new value into table**
5. **Update row in sequence table**
6. **Release lock on sequence table**

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

So?

- Most steps require at least one round trip to the database.
- While the table row for sequence table is locked, no one can add rows to the base table.
- We have to send:
  - 1 Select
  - 1 lock
  - 1 Insert
  - 1 Update
  - 1 release lock
- All require SQLNet roundtrips.
- This was what we did in early versions of Oracle!
- At least 10 network roundtrips
- Plus the recursive SQL

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Problems

- Scalability
- Dependent on network latency
- Performance

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

The Mystery

- Why chose Oracle in the first place?
- Rather like our car example
- If you want Yugo performance, buy a Yugo!

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

The Solution

- Building a database independent application will take longer, be more complex and the resulting application will perform worse than if the strengths of the underlying database are utilized.
- The number of times a major application has been moved from a more capable to a less capable database, are relatively few and the result is never as good as hoped.

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

The Issues

- The issues that drive toward database independence are usually a result of:
  - Buying too much database to begin with
  - Hiring inexperienced developers
  - Developers unwilling to learn database specific skills
  - Being unable or unwilling to get needed skills
  - Throne decisions

A Texas Memory Systems Presentation

Developers

- Cheaper to hire developers who have a complete understanding of your database or to train your existing developers
- The internalized database feature will perform better than anything you develop to run externally, across the network, in a separate tier.

A Texas Memory Systems Presentation

The Issues with Database Independent Applications

1. True portability is almost impossible
2. It is nearly impossible to achieve acceptable scalability
3. It is nearly impossible to achieve acceptable performance
4. Maintenance
5. Retention of skill sets and knowledge

A Texas Memory Systems Presentation

Achieving True Portability

- All database and application logic must be contained within the application itself
- Absolutely no database specific feature can be used.
- This just can't be done except with the most basic of applications.
- All cross-database vendors have database specific routines that interface to the underlying databases, they also have database specific installation scripts for schemas and structures

A Texas Memory Systems Presentation

Scalability

- "network round trip" negates scalability
- The more round trips an application requires the less it will scale.
- More calls to the database increases locks, latches and recursive SQL reducing scalability.
- The more round trips you have in a transaction, the less dependent you are on the server and database for scalability
- You move the dependence to the network which now becomes the Achilles heel of your system.

A Texas Memory Systems Presentation

Performance

- Code contained in the database means database does the work internally
- An example would be the internalization and optimization of referential integrity triggers in Oracle
- Another would be the external compilation of PL/SQL by Oracle11g
- Internal code is also pre-loaded after the first execution and has already been parsed and optimized on subsequent executions.

A Texas Memory Systems Presentation

Performance

- External code means Oracle has to do soft or hard parses depending on how well the code is written
- External code cannot use:
  - Optimized internal routines for common functions such as sequences and referential integrity.
  - Results in more latching, locking, and recursive SQL generating more load on the database server, and, more load on the application tier servers as well.
- Even a perfectly written database independent application cannot out perform a perfectly written database specific application.
- Database independence is at the cost of efficiency and performance.

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Maintenance

- Database independence means re-writing a lot of code that already exists
- Database independent applications must be maintained by the group that writes them.
- Much of the code base is being written using offshore resources
- Your application may be dependent on support once or twice removed
- Utilizing database specific features, languages and capabilities greatly reduces the maintenance needs for your staff

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Retention of Knowledge and Skill Sets

- 5 years down the road you will have the expertise to maintain it?
- Companies where paying Cobol developers to come out of retirement to fix the Y2K problems.
- There will always be PL/SQL and Oracle experts
- People move on, get promoted, get fired, skill sets atrophy or are considered obsolete.
- Developing your application in Y+ might seem a bargain, but what about 5 years down the road when Z# is the skill set offered
- Take the long look ahead instead of to the next quarter.

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Summary

- Database independence is a myth, it doesn't exist.
- Database independence will always mean poorer scalability and poorer performance than with an application that takes advantage of the strengths of its underlying database.
- Independent code increases maintenance, decreases reliability and increases dependence on your development teams (in house or off shore.)
- Just say no to database independence!

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Questions/Comments?

A Texas Memory Systems Presentation

TMS TEXAS MEMORY SYSTEMS

Thank You!

Mike Ault
Mike.ault@texmemsys.com

A Texas Memory Systems Presentation