GENERODB

- Wajahath Quraishi
- wajq@4js.com

Why upgrade an existing system?

- **Technology Advancement**
- **Business Needs (Organizational/ newer opportunities)**
- **Alternate Cost effective solutions**
- **External factors: Market trend / GOVT. regulations/ recession**

Nature of this Industry constantly demands changes. Else we would all be sitting in front of mainframes today

Technology around us is evolving, need to keep providing better services than our competitors and stay ahead in the race

Not always is Technology advancement a cost burden. Picking the right technology can help ROI

In today's Recession: Saves on your licensing and other cost

What do you look for in a database solution?

	GeneroDB
• Fully ACID compliance	YES
• Scalability and performance	YES
• Features like High availability, Backup and Restore, Disaster Recovery, Replication	YES
• Ease of use and easy migration	YES
• Meet ANSI standards and SQL complaint syntax support	YES
• Stored procedures and triggers	YES
• Performance tuning tools	YES

Risks/Challenges to adopt newer solutions (database technology)

- Syntax incompatible and need to modify code and rewrite stored procedures
- Training resources on the new database
- Migration can be tedious and needs good tools
- Normally any upgrade is burden on cost
- Oversight on features supported in the newer database
- More…

Uphill task: Can GeneroDB help !!

- Is moving to newer technology an uphill task?
- Lets learn how GeneroDB will help reduce/mitigate risks to achieve the goal

GeneroDB

- GeneroDB is …
  - A high-performance SQL relational database
  - Compatible with Informix, Oracle, SQL Server, Sybase (supporting their SQL, SPL, triggers)

Helps achieve..

  - Low cost solution
  - High performance achieved thru Minimal locking (much less than any other DB), compilation to machine language, advanced networking
  - Saves you money with fewer database servers (for same performance), fewer licenses, and cheaper hardware

GeneroDB: Performance advantages

- Completely lock-free data structures
- No read locking of rows, and almost no write locking
- GeneroDB Concurrency Engine
- Compiled (to machine language) SQL execution

The only Enterprise Database kernel built from the ground up in this century, architected to current hardware tradeoffs.

Performance advantages: Lock avoidance

- Lock-free datastructures
  - For example, any number of readers and/or writers can be concurrently in an Generodb index, with no waiting ever by anyone
  - All other DBs must walk a lock down the index tree, even for reading

GeneroDB: Lock avoidance..contd

- No read locks (MVCC)
- Writes *do not* exclusively lock the row
- Any number of concurrent pending updates to a row are OK, provided they are *commutative* (order independent)
- Examples:
  - Concurrent adds/subtracts to a number
  - Concurrent updates of distinct columns of a row
- Result will be same as with Oracle (another MVCC db), or Informix

Performance advantages:DB Concurrency Engine

- Extremely fast microthreading architecture that permits non-blocking execution of concurrent queries and provides full scalability
- Data structures and tables optimized for low latency main-memory data access
  - Performance of an “in-memory” db if all data can fit in memory, but …
  - Data can be larger (even much larger) than main memory. Currently unused data overflows to disk.

Performance advantages: Compilation to machine language

- Most db's parse SQL statements or stored procedures to an intermediate form, which is walked and interpreted at every query execution. But interpretation is expensive.
- GeneroDB instead includes an embedded, in-memory compiler which compiles SQL and stored procedures to *machine language* functions
  - Execution is much more efficient

Replication: High Availability (HA) and Disaster Recovery (D.R.)

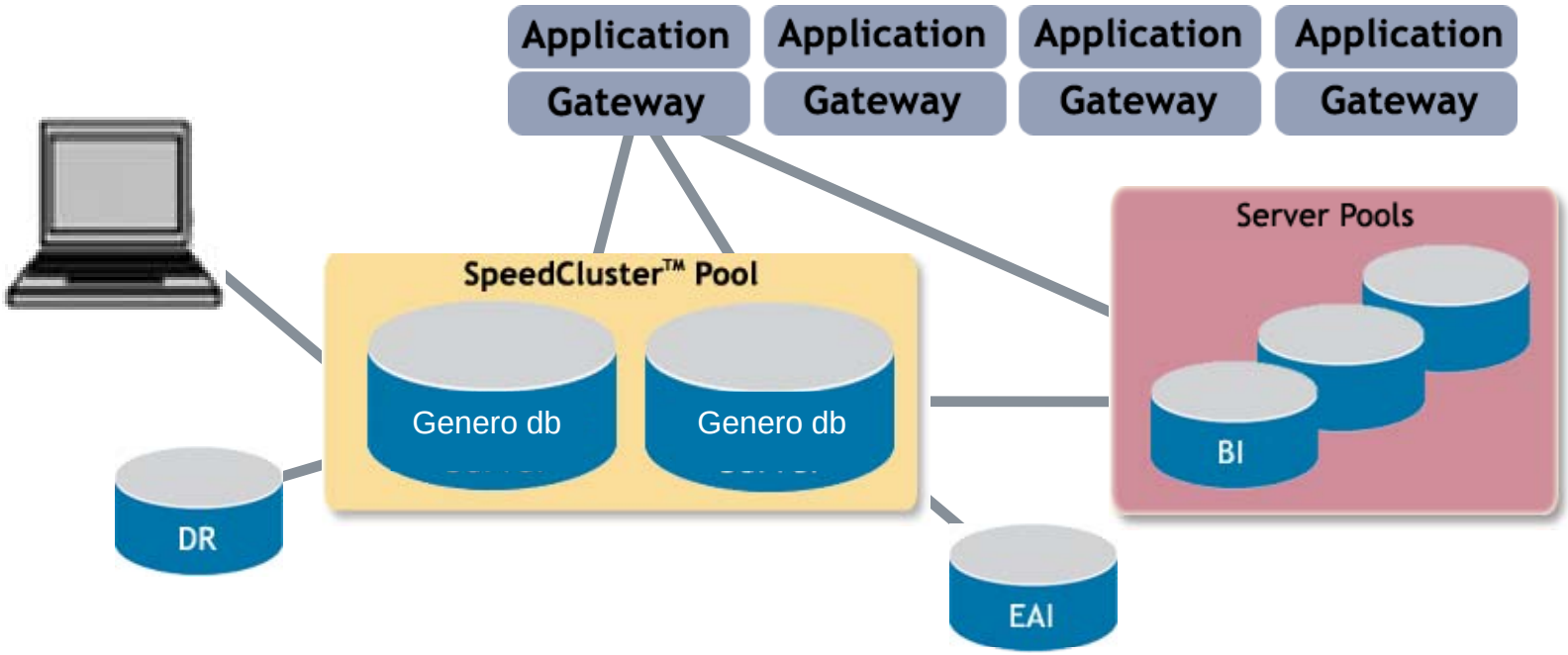
- Master/ Slave (Hot-standby)
- Speed clusters (read-only replicas)
- Log Miner (advanced replication options)

HA: Master/Slave - Hot-standby

- Very easy to set up: just a few configuration parameters
- Automatic failover of all connections if the master should fail
  - Very fast, in milliseconds
  - All committed transactions preserved
  - Uncommitted transactions fail – application replays them and then continues as if nothing happened
  - Otherwise, application is unaware of failure
- Recovery is automatic:
  - just restart failed server; resyncs and resumes replicated operation automatically
  - No rebuild of Master like Oracle

HA: Readable Replicas

- GeneroDB SpeedClusters™ – Readable Replicas:
  - Enables customers to deploy business intelligence, data warehousing, disaster recovery copies of database
  - Extract valuable competitive data, analyze customer trends, comply with disaster-recovery regulations, internal controls



Advance Replication: Event driven log mining with Log Miner

- Log Miner is a “C” API which lets you examine GeneroDB transaction log
- Application can run on any machine (need not be co-located with server)
- Data available as soon as transactions are committed
- Filters allow table selection; applications can also filter on any other criteria
- Replication to other GeneroDB or non-GeneroDB Servers
  - SQL statement to reproduce each change of interest can be supplied.

GeneroDB: Scalability

- The GeneroDB scales almost linearly
  - Scalability limits are locking, HW cache coherency
  - GeneroDB Concurrency Engine avoids both problems:
    - *Much* less locking
    - Much less shared data →less cache coherency problems
- GeneroDB can easily handle thousands of connections, millions of users
  - Wireless Services: 16,000 connections

GeneroDB: Configuration settings

- Transaction Isolation level:
  - Read committed
  - Serializable
- Debug level : Logging level for debugging increases with 0,1 and 2
- Authentication modes: uses SRP (Stanford Remote password) RFC2945
  - Simple
  - Secure Password
  - Secure Username and Password
  - OS level authentication (like Informix)
- HA peer if master-slave

Broad SQL Compatibility

- Databases Supported
  - Informix
  - ORACLE
  - Microsoft SQL Server
  - Sybase
  - Oracle TimesTen
- Datatypes
- Stored Procedures
- Database Triggers

Migration Example
SQL-I (Informix) Language

```
create or replace procedure proc1(x int)
returning varchar, varchar;
define a varchar(100);
define b varchar(100);
define i int;
define x1,x2 varchar(100);
let i=0;
foreach select id,data into x1,x2 from t1
return x1,x2 with resume;
let i=i+1;
end foreach;
select reg_price,promo_price into a,b from
stock;
return a,b;
end procedure;
```

Ease of use: Developers/DBA

EASE of USE, Minimal/No retraining effort needed

For Developers:

- No retraining . Resources can continue with their current skills as we are SQL-92 and some of the SQL 99 standards.
- COMPATIBILITY_MODE : Support for multiple syntax of SPL and SQL (namely T-SQL, PL-SQL and I-SQL for SQLSERVER, Oracle and INFORMIX)

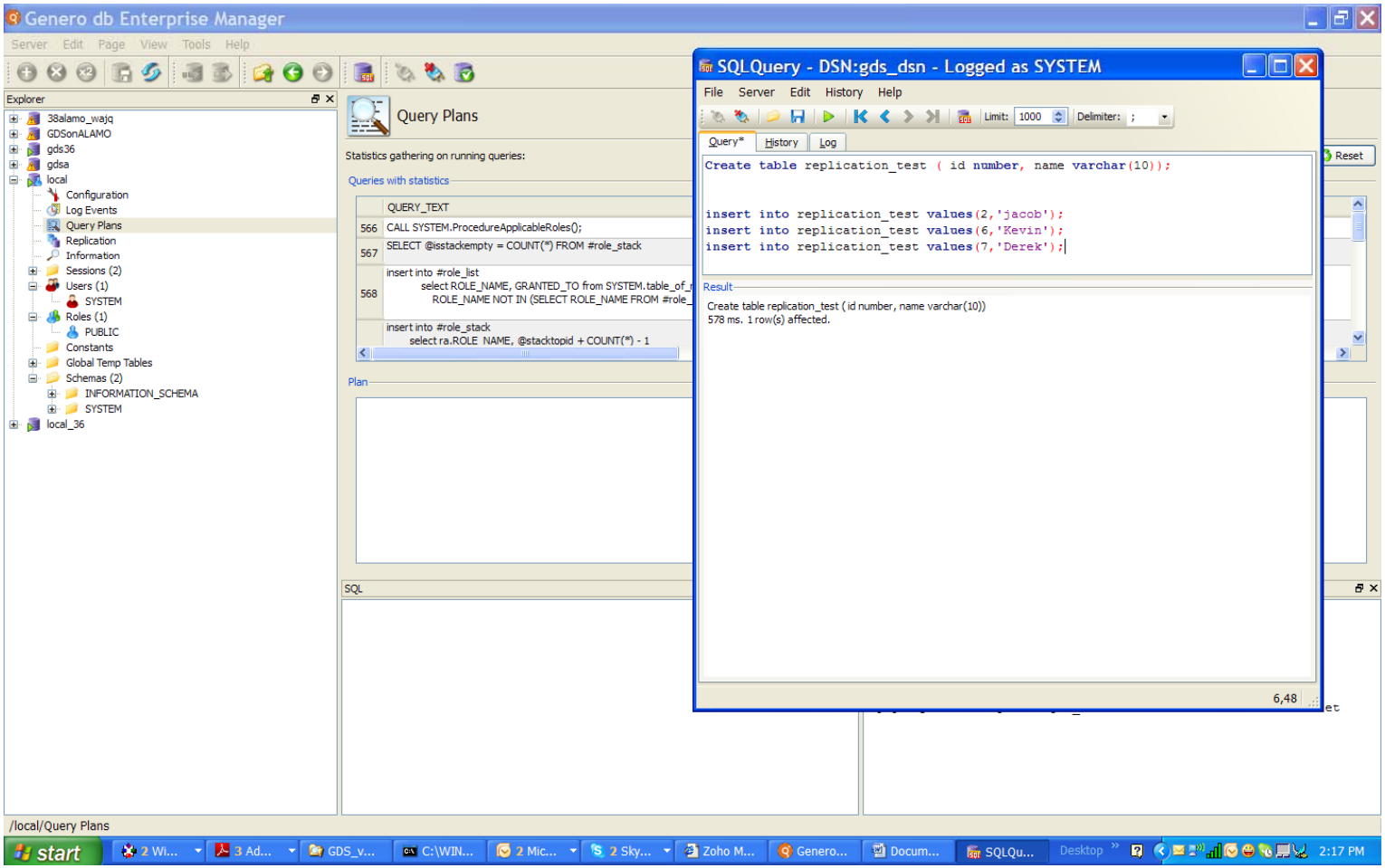
FOR DBA:

- DBA's can leverage the power of GUI based Enterprise Manager tool
- Overall much simpler and easier to manage and administer than other databases

Ease of use: Enterprise Manager

- Administration of many servers local or remote
- Instant view of database engine performance information
- Create/Alter/Remove tables/roles/users
- View columns and data types
- Query data and view results
- Manage HA configurations

Tools: Enterprise Manager



impexp: Migration tool

- An ODBC client Import/Export Utility is included with Genero db client install.
- Can facilitate migration of data between non-Genero db databases and Genero db.
- Currently extend impexp to load Informix IDS .unl exports created with dbexport or UNLOAD.
- It also will take a directory full of .unl files and load them all into the database.

Saving on Infrastructure Cost

- *IR – Greater than 100%*
- *For large number of Connections/ Users*

Sprint Together with NEXTEL	Other RDBMS	Genero db
License Cost	**~ $800K**	**~$112K**
Hardware Cost **(HP 4-way Xeon, 4GB)**	**$320K 16 servers**	**$40K 2 servers**
Improve your performance **(messages per second)**	320/sec	800/sec

GeneroDB 3.80: New Features

- **New Features in Log Miner**
- New Log Miner API functions have been added, including new functions to facilitate its use for replication.
- The Log Miner has an output log and a state file.
- The Log Miner has a Data Load Manager to facilitate propagation of existing tables to a target database.
- **Increased Compatibility**
- Literals can be either single-quoted or double-quoted in Informix or Sybase compatibility modes.
- Informix DATETIME and INTERVAL data types are supported.
- GLOBAL variables in Informix stored procedures .
- Public and private synonyms are supported (compatible with Informix and Oracle).
- UNIQUE is now a synonym for DISTINCT (for Informix compatibility).
- **Improved SQL Functionality**
- GRANT and REVOKE support the Informix-style system privileges, CONNECT, RESOURCE, and DBA.
- New system-level privileges have been added:
- CREATE ANY SYNONYM, CREATE PUBLIC SYNONYM

New features..contd

- **Genero db 3.80 New Features for Improved Manageability and Performance**
- Quality of Service (QoS) (role-based privileges), controls CPU allocation among competing users, to ensure quality of service to time-critical applications.
- User authentication modes are now configurable with parameter AUTHENTICATION_MODE
- SRP security for server logins (RFC2945) provided , both with/without user name encryption
- The performance of client failover in a replicated High Availability server configuration has been improved; now 200 milliseconds to failover.
- Readable replicas and Log Miners have improved fault tolerance when receiving from a replicated High Availability log forwarding server
- Readable replicas and Log Miners can continue to receive if either server of an High Availability log forwarding server pair fails
- The Import/Export utility uses Flood Insert by default
- In replicated environments, the configuration parameter, FAILOVER_TIMEOUT, has been replaced with a new parameter, REPLICATION_TIMEOUT, with improved functionality