Dynamic Database Solutions

Breakthrough Security and Performance Solutions from ActiveBase

Tony Cannizzo | Managing Partner
Dynamic Database Solutions

About ActiveBase

- Based in Israel
- Founded in 2002
- Released technology in 2004
- Experienced Oracle veterans
- Innovative technology protected by patents
- About 50 major production implementations primarily across Europe
- US Launch now underway

ActiveBase Innovation

- Actionable In-Line Proxy
- Intercepts and Evaluates all In-Bound SQL
- Dynamically Applies Solution
  - ? Match SQL
  - ! Take Action

? Match SQL

User
Program
IP Address
Time of Day
Syntax based statement
From Clause Object
Execution plan
Text string matching
PL/SQL function
Java API

ActiveBase Modules Overview

User

Application

activebase

ActiveBase in-line Proxy modules:

Performance Module

- Applies SQL **Hints** / **Rewrite** for improving performance
- **Redirects** report /ad-hoc to replication / history DB
- **Blocks** / **defers** 'request-from-hell' for safe guarding production

Security Module

- Masks personal information for outsourced support and IT
- **Blocks** offensive Requests / SQL injection / CPU risks
- **Scrambles** / **Encrypts** confidential & personal fields

Prioritization module (DB server)

- **Allocates** Database Server resources to processes according to transaction importance
- **Manual** Operator
- **Automatic** – Rules

Tuning Robot

ActiveBase Priority

activebase

Manage Server CPU and I/O resources During Peak Processing

High priority business transactions receive more server resources by throttling down resource allocation to lower priority transactions

DynamicDB

ActiveBase Performance

ActiveBase Security

Masking Rules:

Original SQL:
Select name,..from..

After Rule:
*Select substr(name,1,2)||'***'*

Result:

Name
Tiger
Nelson
Rogers
Rosen

Name
Ti***
Ne***
Ro***
Ro***

Scrambling Rules:

Original SQL:
Select name,..from..

After Rule:
Select scrmbl(name)..

Result:

Name
Tiger
Nelson
Rogers
Rosen

Name
Bell
Cave
Lennon
Lenin

Hiding Rules:

Original SQL:
Select name,..from..

After Rule:
select ..,'',..from..

Result:

Name
Tiger
Nelson
Rogers
Rosen

Name

Blocking Rules:

Original SQL:
Select name,..from..

After Rule:
Returned message:
You are not allowed to access this personal information!

Dynamic SQL Optimization

- ActiveBase Performance

When The Answer Is…Rewrite the SQL

activebase

- This is not as easy as it sounds
- Some of the most critical performance problems are in applications where you do not have source code:
  - <u>Proprietary Applications</u>
    - PeopleSoft, Siebel, Amdocs, ESRI, Oracle EBS, Cerner
  - <u>Development Tools</u>
    - Toad, DBArtisan, SQL*Plus
  - <u>End-User Query and Reporting</u>
    - Cognos, Business Objects, Crystal Reports
  - <u>Legacy Applications</u>
    - Long Development Cycles
    - Source Code not Readily Available

DynamicDB

Challenges with Packaged Applications

activebase

Vendor Challenges

- Broad variety of customers
- Different environments will generate different application hot-spots
- Often, problems cannot be duplicated in the lab
- Solutions for individual clients could not be guaranteed

Customer Challenges

- Needs product to work well in their environment
- Can't modify the source code or database
- "At the mercy" of the vendor
- Often, the customer knows exactly what the problem and the solution is

DynamicDB

Customer Success Example

activebase

- Major telecom running proprietary billing app
- Used monitor to diagnose problem
- Handed it to the vendor on a silver platter
  - The Bad SQL
  - Why it was Bad
  - What is Should Be
- Vendor Engineering Response
  - In the queue – 12 months to get to it
  - Estimated Special Fix Fee $500K–$800K
- Installed ActiveBase in the morning
  - By 2:00PM jobs that were taking over 14 hours completing in 20 minutes!
  - Didn't have to touch the source code or the database

DynamicDB

Layer 1: Switching/Routing Rule

Define Profiles

- Users
- Applications
- Databases
- Associate Rule Sets to be used

Layer 2: Statement Processing Rules

activebase

- **Match** incoming statements and **apply actions** on them (e.g., rewrite/block etc.)
- **Matcher types** include:
  - Syntax based statement
  - From Clause Object
  - Execution plan subset
  - Execution plan step
  - Partition range
  - Text string matching
  - PL/SQL function
  - Time based matching
  - Oracle cost matching
- **Actions** include:
  - Rewrite
  - Add hint
  - Search and replace
  - Define symbol

DW Knowledge Pack

activebase

ActiveKnowledge Rule Editor

File Action

Rules
- Rules applied when DB in high load
- Speed performance rules
- Partition_limit
- ETL Prioritization and user guidance
- Automatic Routing rules to History D
- Logging

Name: Rules applied when DB in high load

Description

Matcher

Identify statement by: PL/SQL Function

Function name: ak_high_load

- [ ] Include Oracle Username
- [ ] Include OS User
- [ ] Include Client Host
- [x] Include Program Name
- [x] Include Statement
- [ ] Include text value

Value to be used:

- [ ] Keep matcher result

Then try to match once every 0 seconds per session

Rule Processing method: Stop if applied

DynamicDB

Rule Tree

activebase

ActiveKnowledge Rule Editor

File Action

- Rules
  - SQL Syntax - Rewrite
  - EMP Rules
  - BIG Table Test
    - Big Table Explain Plan
      - INDEX (FAST FULL SCAN) (
    - Big Table Exec
  - Sysdate Symbol Exchange Examp
  - DW Knowledge Pack
    - Rules applied when DB in hig
      - Block long queries
      - Identify high cost querie
      - Add hint no parallel to lo
    - Speed performance rules
      - Report Centric
        - Add Hint based on E
        - Allocate more paralle
      - Object Centric, for ad-h
        - Contract_Status
        - og_goods_by_NUME
        - Converting char con
    - Partition_limit
      - Substitute SYSDATE
      - DW_FACT_USAGE range
      - DW_FACT_USAGE range
      - DW_FACT_SUB_CDR rar
    - ETL Prioritization and user gu
      - Notification wile ETL not
      - Set Low Prioritization
    - Automatic Routing rules to H
      - check_ic_date
      - check_og_date
    - Logging

Name: Big Table Exec

Description:

Matcher

Identify statement by: Oracle Cost

Minimum Cost: 2000

Keep matcher result

Then try to match once every 3600 seconds per session

Action

Action applied when identification occurs: Block

Error message returned

Oracle cost too high!!!

DynamicDB

Rewrite Date Processing

activebase

ActiveKnowledge Rule Editor

File Action

Rules
- SQL Syntax - Rewrite
- EMP Rules
- BIG Table Test
- Sysdate Symbol Exchange Example
- DW Knowledge Pack
  - Rules applied when DB in high lo
    - Block long queries
    - Identify high cost queries
    - Add hint no parallel to low c
  - Speed performance rules
    - Report Centric
      - Add Hint based on Expl
      - Allocate more parallel se
    - Object Centric, for ad-hoc a
      - Contract_Status
      - og_goods_by_NUMBER
      - Converting char conditi
  - Partition_limit
    - Substitute SYSDATE
    - DW_FACT_USAGE range 3 r
    - DW_FACT_USAGE range 3 r
    - DW_FACT_SUB_CDR range
  - ETL Prioritization and user guida
    - Notification wile ETL not con
    - Set Low Prioritization
  - Automatic Routing rules to Histo
    - check_jc_date
    - check_og_date
  - Logging
    - log_many_partitions_OC

Edit Rule

Name: erting char condition into date condition

Description:
select * from cust_order where str_date_dmn.month_number = '2007/10'

Matcher

Identify statement by: Text

Text:
(.*)str_date_dmn.month_number = '(.*?)'(.*)

Identification method: String / Wildcard / Regular Expression

Case sensitive

Keep matcher result

Then try to match once every 0 seconds per session

Action

Action applied when identification occurs: Rewrite

Alternate statement:
\(1)str_month_dmn.date between to_date('\(2)','yyyy/mm') and last_day(to_date('\(2)','yyyy/mm')) \(3)

Rule Processing method: Stop if applied

Log when rule is applied

Ok Cancel

DynamicDB

Testing Rule Results

activebase

ActiveKnowledge Rule Editor

File Action

Rules

Test Rules

Parsing schema SCOTT

Enter a SQL statement:

```
select * from str_date_dmn.month_number = '2007/10'
```

Rule test results:

Rule used: [Converting char condition into date condition, Object Centric, for ad-hoc and reports, Speed performance rules, DW Knowledge Pack]

```
select * from str_month_dmn.date between to_date('2007/10','yyyy/mm') and last_day(to_date('2007/10','yyyy/mm'))
```

DynamicDB

Rule: Hint based on Execution Plan

activebase

ActiveKnowledge Rule Editor

File Action

Rules
- General
- Report Centric_ShortTerm
  - Peter
  - Resource investigation
- Object Centric
  - Contract_Status
    - Contract_status$UK
      - PARTITION RANGE (ITERATOR
        - TABLE ACCESS (BY LOCAL
          - INDEX (RANGE SCAN)
- Access Control
- Logging

Name: Contract_status$UK

Description:

Matcher

Identify statement by: Execution Plan Subset

Minimum Cost: 0

Action

Action applied when identification occurs: Add Hint

Hint text:

```
/*+ NO_INDEX(CONTRACT CONTRACT$XX) AKRULE[Contract$XX] */
```

When Hint already exist: Replace to the new Hint

☐ Check Excution Plan changed

☐ Check Oracle cost is not increased

DynamicDB

Enforce Dev Tool Usage Policies

activebase

Toad, DBArtisan, SQL*Plus, etc.

- Restrict parallel load:
  - allow up to four parallel servers for all Toad requests
  - or dynamically remove the parallelism from the request

- Block specific DB activities from either authorized or unauthorized users:

  `locks, drop table, drop synonym, drop grant`

- Selectively preventing DML, DCL, DDL commands from unauthorized users

- Automatically redirect requests to the REPORT DB when request includes certain conditions

DynamicDB

Rule: Disable Parallel for Toad

Rule Editor - SQL

File Action Help

- SQL
  - Audit
  - Masking examples
    - Mask Credit Cards
  - Prevent general account usage
    - Stop apps usage
  - Identify and control offensive requests
    - Identify and control offensive requests
    - SQL Injection
      - Audit SQL Injection using DML/DCL/DDL Commands
  - Control Privileged Users
    - Apply Alter Session For Toad (No Parallel)
    - Identify DCL commands
    - Identify DDL Commands
    - Identify DML Commands

Rule Name: oad (No Parallel)

Description

Toad is running the following SQL command during the toad initialtion:
begin sys.dbms_application_info.set_module('TOAD', null); end;
ActiveBase rule rewrites these command to add an any 'alter session' command you require. The exmple includes changing session with the following command:
execute immediate 'ALTER SESSION DISABLE PARALLEL QUERY'

Matcher

Matching Method: Text

Text: begin sys.dbms_application_info.set_module('TOAD%

Identification method: String / Wilcard / Regular Expression

Case sensitive

Keep matcher result

Then try to match once every 0 seconds per session

Action

Action Type: Rewrite

Alternate Statement: begin sys.dbms_application_info.set_module('TOAD Freeware 8.5.0.50', null); execute immediat 'ALTER SESSION DISABLE PARALLEL QUERY';end;

Whenever this rule is matched then: Stop if applied

Log when rule is applied

Rule: Identify DCL Commands

activebase

Rule Editor - SQL

File Action Help

- SQL
  - Audit
  - Masking examples
    - Mask Credit Cards
  - Prevent general account usage
    - Stop apps usage
  - Identify and control offensive requests
    - Identify and control offensive requests
    - SQL Injection
      - Audit SQL Injection using DML/DCL/DDL Commands
  - Control Privileged Users
    - Apply Alter Session For Toad (No Parallel)
    - Identify DCL commands
    - Identify DDL Commands
    - Identify DML Commands

Rule Name: DCL commands

Description: DCL commands. You do not need to change the list.

Matcher

Matching Method: Text

Text:

```
(/\*[^\*]*(\*[^/][^\*]*)*\*/)?\s*((GRANT)|(REVOKE)).*
```

Identification method: String / Wilcard / Regular Expression

Case sensitive

Keep matcher result

Then try to match once every 0 seconds per session

Action

Action Type: Nothing

Whenever this rule is matched then: Stop if applied

Log when rule is applied

DynamicDB

Rule: Identify DDL Commands

Rule: Identify DML Commands

Rule: Block Unauthorized DDL

activebase

?

!

Edit Rule

Name: Control DDL for these users

Description: Developers are not allowed to issue DBA Commands

Matcher

Identify statement by: Text

Text:

```
((ALTER)|(DROP)|(CREATE)|(ANALYZE)|(ASSOCIATE STATISTICS)|(AUDIT)|(COMMENT)|(DISASSOCIATE STATISTICS)|(FLASHBACK)|(NOAUDIT)|(PURGE)|(RENAME)|(REVOKE)|(TRUNCATE)).*
```

Identification method: ○ String ○ Wildcard ◉ Regular Expression

☐ Case sensitive

☐ Keep matcher result

Then try to match once every 3600 seconds per session

Action

Action applied when identification occurs: Block

Error message returned:

You are not allowed to run DDL commands. Please contact DBA team.

DynamicDB

Privileged User Control

activebase

TOAD for Oracle Freeware - [SYSTEM@ORCL SQL Editor (<No name>)]

File Edit Grid SQL Editor Create Database Tools View DBA Window Help

SYSTEM@ORCL

Set Schema SYSTEM

<No name>

```
select sal from scott.emp
```

Warning

Line 1, statement 1 of 1

SQL Error

ORA-00900: Accessing the employees table is not allowed!

select sal from scott.emp

Continue

Cancel

Ignore All

Data Grid

Data Grid | Script Output | Explain Plan

1: 1 SYSTEM@ORCL Modified

AutoCommit is OFF CAPS NUM INS

DynamicDB

Deployment Strategies

Dynamic Resource Prioritization

- ActiveBase Priority

First challenge – Guarantee and improve service levels (QoS)

activebase

- Exponential data growth and growing application usage cause resource-stress and peak-times
- Improving performance during peak-times and cycle run-time when Quality of Service deteriorates
- **Improving performance while saving and deferring hardware purchases and software upgrades**

DynamicDB

ActiveBase Priority Architecture

activebase

ERP, CRM etc. Database servers:

ActiveBase Priority Server

Rule Engine

Identification functions: Server CPU and I/O, Instance and session CPU, Session info, SQL syntax patterns, time of day, define user grouping

Priority Actions: Reduce process resources, Limit process resource consumption

Oracle Monitor and OS Monitor

Process Manager (throttling processes)

DynamicDB

Session Rules

activebase

Active sessions are collected (every 15 sec.) and processed according to rule flow (top to bottom):

Operator dashboard enables TO MANAGE AND CONTROL RESOURCE USAGE in a single mouse click

Immediate effect

activebase

```
oracle@isolinux01:/oradata/flash_recovery_area/
Fichier  Édition  Affichage  Terminal  Onglets  Aide
top - 00:12:02 up 12:48,  7 users,  load average: 2
Tasks: 129 total,   5 running, 122 sleeping,   0 st
Cpu(s): 96.5% us,  2.4% sy,  0.0% ni,  0.8% id,  0.
Mem:    513772k total,   497976k used,    15796k fr
Swap:  3068404k total,   111484k used,  2956920k fr

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM
 6210 oracle    25   0  370m  50m  47m R   87  0.1
 6473 oracle    25   0  368m  30m  28m R   58  6.1
 6386 oracle    25   0  369m  25m  23m R   48  5.1
 5638 active    15   0 22112 4552 3860 S    8  0.9
 3924 root      15   0 98[illegible]m 9200 5306 S    4  1.8
 5726 [illegible]                                0 S    1  3.0
 5941 [illegible]
 6484 [illegible]
    1 [illegible]                                4 S    0  0.1
    2 [illegible]                                0 S    0  0.0
    3 [illegible]                                0 S    0  0.0
```

Process ID 6210
Before AB:
87% CPU

```
oracle@isolinux01:/oradata/flash_recovery_area/S
Fichier  Édition  Affichage  Terminal  Onglets  Aide
top - 00:33:40 up 13:10,  8 users,  load average: 1.
Tasks: 134 total,   1 running, 127 sleeping,   2 sto
Cpu(s): 12.1% us,  3.5% sy,  0.0% ni, 84.1% id,  0.0
Mem:    513772k total,   498168k used,    15604k fre
Swap:  3068404k total,   139008k used,  2929396k fre

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM
 5638 active    15   0 22112 4552 3860 S   15  0.9
 6386 oracle    15   0  369m  25m  23m T    6  5.1
 6210 oracle    15   0  370m  50m  47m [illegible]    6  0.1
 3924 root      15   0 99.0m 9596 5656 S    4  1.9
 5726 [illegible]                                    3.1
 5941 [illegible]                                    0.2
    [illegible]                                      0.2
    [illegible]                                      0.1
    2 [illegible]                                    0.0
    3 [illegible]                                    0.0
    4 [illegible]                                    0.0
```

Process ID 6210 with
AB*Priority Rules:
6% CPU

- The results: important SQL requests run 2 times faster

DynamicDB

Live Demo – Q&A

activebase

THANK YOU!

Tony Cannizzo | Managing Partner

Dynamic Database Solutions

DynamicDB

tony@dynamic-db.com

404 580 3451

DynamicDB

Finding Problems

- SQL Expert and Tuning Robot

Optional Slides

Define target database

> Define database parameters for connecting to the database. ActiveBase Expert opens a jdbc connection to analyze and run serially (benchmark) the different execution alternatives.

> To analyze different statements, DBA user name requires the following pri

grant select any table to user_name

grant select any dictionary to user_name

grant alter session to user_name

grant resumable to user_name

grant become user to user_name

ActiveBase Ltd. All Rights reserved

Analyze possible alternatives

- Copy and paste your long running ‘Select’ SQL request into the text field or click ‘Find Hotspots’ to copy and paste a ‘Select’ request from Oracle SGA.
- Then click on ‘Find Alternatives’ button.

Click on ‘’ to view execution plan

Click on ‘Options’ to set the **benchmark execution parameters**

ActiveBase Ltd. All Rights reserved

Define Benchmark Parameters

> The Option parameters are used in the benchmark execution when finding the best hint.

> Options enable to set the number of times each alternative will be tested and total benchmark time.

> It is always recommended to set the automatic execution canceling for stopping alternatives running more that X% of the original time.

ActiveBase Ltd. All Rights reserved

Run benchmark

> Click on ‘Select all’ to mark alternatives for execution.

> Click on the ‘Cost’ column for sorting execution order based on increasing Oracle cost value. Click ‘run’ to start the benchmark which executes the checked alternatives.

Completed executed alternatives are colored in **green**.
Automatic cancelled alternatives are colored in **red**.
The best time alternative is colored in **blue**.

ActiveBase Ltd. All Rights reserved

Compare execution plan of alternatives

Click on ‘Compare’ for comparing execution plans of different SQL ‘Hint’ alternatives. Choose the ‘Hint’ to compare

Parsing schema abdemo | Session Parameters

SQL for analyzing:

```
SELECT  CUSTOMERS.CUSTOMER_NAME Customer,  CONTRACT.CONTRACT_CODE Contract, ACTIVITY.ACTIVITY_CODE
Activity,  ACTUAL_WORK.WORK_MONTH Month, ACTUAL_WORK.WORK_MAN_MONTH Man_month,  ACTUAL_WORK.
type, BUD_WORK_PLAN.WORK_PLAN_MONTH, BUD_WORK_PLAN.WORK_PLAN_MAN_MONTH , CUSTOMERS.CUSTOM
FROM
CUSTOMERS,
CONTRACT,
ACTIVITY,
ACTUAL_WORK,
BUD_WORK_PLAN
WHERE
```

Find Hotspots | Analyzer Levels: 3 | Find Alternatives

No.	Exec.	Hint	Cost	Plan	Elapsed	Buffer ...	Disk Re...	Disk W
1	☑	Original	78		0.156	25985	0	0
2	☑	RULE	0		1.047	117270	1	0
3	☑	USE_NL(B...	81		0.156	30115	0	0
4	☑	ORDERED	80		0.148	30908	0	0
5	☑	INDEX(AC...	417		0.152	31966	0	0
6	☑	LEADING(...	81		0.156	32766	0	0
7	☑	INDEX(BU...	417		0.16	34321	0	0
8	☑	INDEX(BU...	258		0.16	35722	0	0
9	☑	FULL(CUS...	80		0.14	36520	0	0
10	☑	INDEX(CO...	77		0.152	37324	0	0
11	☑	LEADING(...	80		0.148	38122	0	0

Compare | Select All | Deselect All | Selected 448/448 | Options

Compare Execution Plans

1. Original

- Execution Plan
  - SELECT STATEMENT, COST: 78
    - HASH JOIN, COST: 78
      - HASH JOIN, COST: 49
        - TABLE ACCESS (FULL) OF ACTUAL_WORK, COST: 28
        - HASH JOIN, COST: 20
          - NESTED LOOPS, COST: 4
            - TABLE ACCESS (FULL) OF CONTRACT, COST: 3
            - TABLE ACCESS (BY INDEX ROWID) OF CUSTOMERS,
              - INDEX (UNIQUE SCAN) OF SYS_C005624
          - TABLE ACCESS (FULL) OF ACTIVITY, COST: 15
      - TABLE ACCESS (FULL) OF BUD_WORK_PLAN, COST: 28

9. FULL(CUSTOMERS)

- Execution Plan
  - SELECT STATEMENT, COST: 80
    - HASH JOIN, COST: 80
      - HASH JOIN, COST: 51
        - TABLE ACCESS (FULL) OF CUSTOMERS, COST: 3
        - HASH JOIN, COST: 48
          - TABLE ACCESS (FULL) OF CONTRACT, COST: 3
          - HASH JOIN, COST: 44
            - TABLE ACCESS (FULL) OF ACTUAL_WORK, COST: 28
            - TABLE ACCESS (FULL) OF ACTIVITY, COST: 15
      - TABLE ACCESS (FULL) OF BUD_WORK_PLAN, COST: 28

Close

ActiveBase Ltd. All Rights reserved

Benchmark result audit trail for best Hint

Open the [installation directory] \ActiveBase Expert\log directory
View the file: expert-[date/time].out

For the original statement and for each alternative:
1. SQL text including the Oracle Hint
2. Full SQL execution statistics
3. Execution plan

```
Wed Jul 15 00:00:32 IDT 2009 Executing: /*Original*/SELECT  CUSTOMERS.CUSTOMER_NAME Customer,  CONTRAC
Wed Jul 15 00:00:33 IDT 2009
Original
SELECT  CUSTOMERS.CUSTOMER_NAME Customer,  CONTRACT.CONTRACT_CODE Contract, ACTIVITY.ACTIVITY_CODE □Ac
normalizedTime: 156
elapseTime: 156
cpuTime: 12640
logicalReads: 25985
physicalReads: 0
sorts: 529
rows: 4999
rowsUsed: 274563
parallelDegree: 0
Statistics: {db block changes=103836, parse count (hard)=6329, commit immediate requested=2108, physic
Events: {SQL*Net message to client=10, CPU=227, SQL*Net message from client=340}

PLAN: SELECT STATEMENT, COST: 78
PLAN:  HASH JOIN, COST: 78
PLAN:   HASH JOIN, COST: 49
PLAN:    TABLE ACCESS (FULL) OF ACTUAL_WORK, COST: 28
PLAN:    HASH JOIN, COST: 20
PLAN:     NESTED LOOPS, COST: 4
PLAN:      TABLE ACCESS (FULL) OF CONTRACT, COST: 3
PLAN:      TABLE ACCESS (BY INDEX ROWID) OF CUSTOMERS, COST: 1
PLAN:       INDEX (UNIQUE SCAN) OF SYS_C005624
PLAN:     TABLE ACCESS (FULL) OF ACTIVITY, COST: 15
PLAN:   TABLE ACCESS (FULL) OF BUD_WORK_PLAN, COST: 28

Wed Jul 15 00:00:33 IDT 2009 Executing: /*RULE*/ SELECT   /*+ RULE */  "CUSTOMERS"."CUSTOMER_NAME" "CU
Wed Jul 15 00:00:34 IDT 2009 Canceled
```

ActiveBase Ltd. All Rights reserved

Get best alternative

- To get the best alternative SQL text, go to the 'Best alternative' line colored in blue and double click on the 'Hint' column.
- A pop-up window is presented. Copy the ‘Hint’ text and paste it to the beginning of your original SQL text.

ActiveBase Ltd. All Rights reserved