

This presentation focuses on the convergence of root cause analysis and Oracle internals.

A few months ago, I was listening to a local radio show that deals with automotive problems. A caller had a very perplexing issue and the hosts recommended that he take the care in so that they could diagnose the problem. The key point that I recall is the discussion of “diagnostic routines” and how they make decisions based on the results of simple tests and gathering of data. About the same time, a colleague asked me to diagnose a sql issue and we started talking about the steps I used to identify and resolve the issue. I realized that I was using my own diagnostic routines inside my head...but I never had formalized the process.

Documenting the processes I use to diagnose problems is quite a challenge. We all know how to do things...it is often difficult to explain and document the reasons why we do what we do.

One of the issues in Oracle Administration (and IT in general) is a lack of discipline in diagnosing issues. Last year I read a paper that Robyn Sands was working on regarding using an industrial engineering approach to managing Oracle databases. It was a revelation.



Things to Know

- Presentation *not* on CD
 - www.optimaldba.com/papers.html
 - Paper is on CD and covers other items
- Not really technical
 - Some discussion of tools
 - Goal is to start you on the way to developing your own routines
- Questions or Comments?
 - Don't wait until the end
 - daniel.fink@optimaldba.com

We All Diagnose Problems

- Every day issues
 - Car won't start
 - House too cold
 - Gravy too thin
- Learned routines
 - Symptom(s)/Decision(s)/Solution(s)
 - Adapt for conditions/situations

We all diagnose problems in every day life, problems that have nothing to do with computers.

Think about what you do when your car won't start. What process do you use? What questions do you ask? Tests do you perform?

You also adapt for conditions. For example, if you just filled up the car with gas, you probably won't ask the question "Am I out of gas?" Not that you know for certain that the engine is getting fuel.

ABORS*

- Guess and Grimace
- Groundhog Day
- Whack-A-Mole
- Panic and Point
- IAMM

OptimalIDBA.com

Stop Chasing Your Tail

Slide 4 of 26

ABORS stands for A Bunch Of Random Stuff

The current state of Oracle diagnosis is less than disciplined. While we have made great strides in the last decade using approaches such as response time optimization, we are still not

Guess and Grimace This term was coined by Mogens Norgaard to describe the actions of people who engage in troubleshooting exercises using random ideas (brainstorming) instead of taking a more disciplined scientific approach. I often see this when

Groundhog Day Another Mogens term, this describes repetitive efforts that are not producing lasting results.

Whack-A-Mole This one is from Robyn Sands and describes the situation where the same situation keeps coming up and you never really solve it.

Panic and Point identify performance problems as originating anywhere except in your area of responsibility

IAMM How often do you hear the terms 'Firefighting', 'Tiger Team' or 'SWAT Team'? This is "Inappropriate Allocation of Macho Metaphor". One of my issues is that these teams practice/practice/practice

What Makes A Good Routine?

- Relevant
- Comprehensive
- Efficient
- Deterministic

Relevant – the routine is relevant to the application, system, version, situation. Using an 7.3 diagnostic routine for a 10.2.0.4 database is not necessarily relevant. If you find that you are skipping tests/questions because they no longer apply, it is time to review and update the routine.

Comprehensive – it needs to be able to cover almost every likely scenario. If you find yourself skipping steps because they are no longer relevant, it is likely that you are also missing steps.

Efficient – a routine that wastes time to arrive at the root cause is not going to be all that useful. The idea behind a routine is to have a process you can follow while under pressure.

Deterministic – if the same root cause with the same symptoms results in different diagnosis, then you have to question the routine.



Developing a Routine

- Your Tasks

- Know Your System
- Tool Proficiency
- Question/Answer or Test/Result

- Your Knowledge

- Root Cause Analysis Process
- Oracle Architecture

This is the basic process for developing routines.

Know Your System

- Oracle
 - Architecture
 - Query Processing
 - Timed Events and Statistics
- 'Your' System
 - Connections
 - Business Use

One of the main foundations of developing a routine is to know your system. How can you diagnose a problem with a system if you don't know the system? This requires that your knowledge remain relevant and comprehensive. If your knowledge stopped at 7.3.4, you cannot really create a diagnostic routine for 10.2.0.4.

Tool Proficiency

- What tools are at your disposal?
 - Strengths/Weaknesses
 - Blind Spots
- What situation calls for what tool?
 - Systemic
 - Session
 - Statement

A tool is only as good as the developer. And it will only show you what the developer considers to be important. One great example is that the Oracle instrumentation shows a great deal of information about physical reads, but almost no information about logical reads.

Asking Questions

- 5 Whys
- Decision Tree
 - Question/Test
 - Find answer/result
 - Next step
- The Devil is in the Details
 - Properly phrased question
 - Script/Tool steps for Answer
 - Expected Output

The 5 Whys are a method for arriving at the root cause of a problem. The idea is that you keep asking questions until you arrive at the final answer. Sometimes you arrive at the answer in 3 questions (or less) or it may take you 10 questions (or more). When you can't ask any more relevant questions, you should have arrived at the root cause.

Your decision tree is really just 3 components. Some of the questions/tests have yes/no answers, but there are going to be some that have multiple values, for example a set of events and the next step for each event.

Diagnostic Routine Components

- Symptom Description
- Question/Test
- Answer/Result
- Decision
- Determination
- Resolution
- Verification

A common problem is that the initial issue is described as a problem or solution instead of a symptom.

The Question/Test, Answer/Result, Decision process needs to be repeated until you arrive at a root cause.

Determination is arriving at the root cause.

Resolution and Verification are the act of fixing the problem and verifying that the fix is proper. While these are not strictly part of the diagnostic routine, they are key components to validating that the routine is working and the problem is actually solved.

Diagnostic Routine Examples

- Just examples...not total coverage
 - Simple examples
 - Not all tests/results/decisions are covered
- Use the examples to build your own

These examples are just a couple that are easy to demonstrate. They are not complete, but are intended to give you an insight into some basic routines and how to start developing them yourself.

System Diagnostic Routine

■ Tools

□ O/S

- Process Monitor
- System Overview

□ Oracle

- Views
- EM/Grid Control

Multiple User Complaints

- “sessions are hung”
 - Multiple reports within a few minutes
- Cannot create new sessions
 - Actual error message not returned

This may start as a single session or statement diagnosis. Be prepared to change the routine if you find out that your initial assessment is not correct.

Is the Instance Up?

- Yes, the required background processes running
 - Can you log in as a regular user?
 - No, you get the error ORA-00257 archiver error. Connect internal only, until freed.
 - Check archive destination and free space if full
 - Compare normal redo generation with current generation
 - Close to normal then check backup/archive log purge processing
 - In excess of normal then check for cause of excess redo generation
- No, the required background processes are not running
 - Check the alert log

This is an actual situation that has occurred several times.

Session Diagnostic Routine

- What are the possible session states?
 - On CPU
 - Waiting on CPU
 - Instrumented Event
 - Uninstrumented Event
- What do certain events mean?
 - SQL*Net Message
 - TM Enqueue
 - db file sequential read

One key to diagnosing session issues is to understand the session's state. And what each of those states means. For example, a query that is spending a lot of time On CPU, the likely cause is a lot of logical i/o. If the event is db file sequential read, then it is doing a lot of physical i/o.

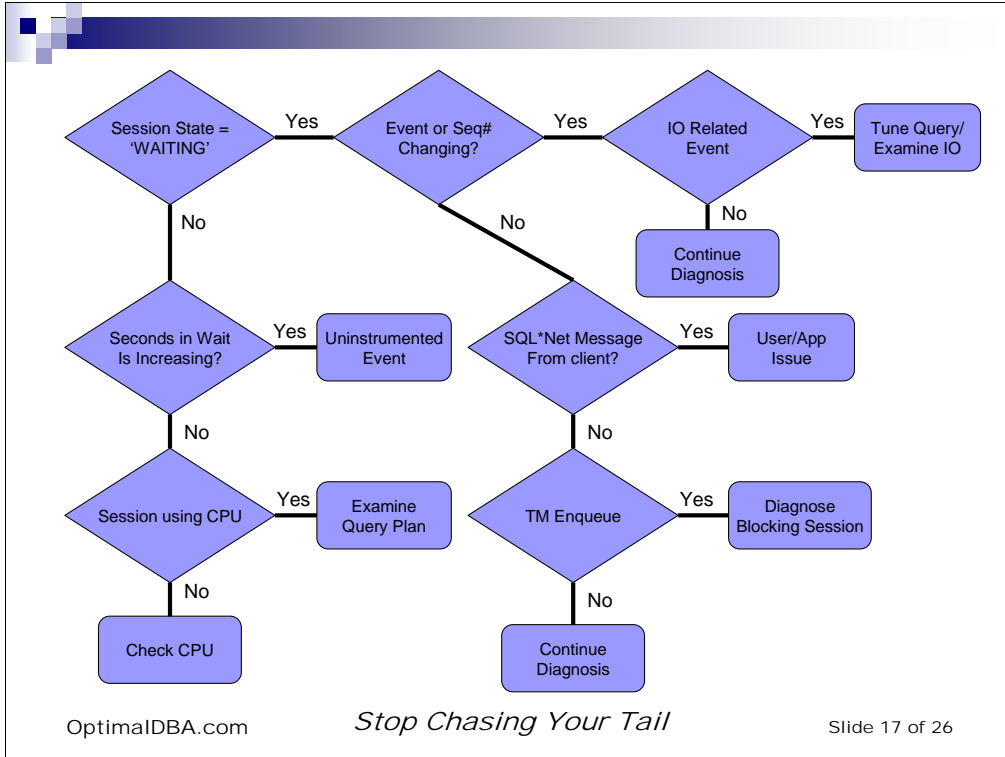
Tools

■ Views

- v\$session
 - v\$session_event
- v\$session_wait_history
- v\$sesstat
 - v\$statname

■ Profile

- dbms_xplan





Statement Diagnostic Routine

- Locate inefficiencies
 - Secondary Throwaway
 - Primary Throwaway
- Isolate operations that consume the most
 - Time
 - Resource



Tools

- sql trace and tkprof
- dbms_xplan
- Autotrace
- dbms_profiler
- Events



Routine

- What step(s) show the most throwaway?
- Is there an unindexed predicate?
 - Yes – create index for predicate
 - No – Is the predicate construct able to use the index?
 - Yes – diagnose optimization decision
 - No – alter predicate or create function-based index



Secondary Throwaway

- Easy to spot
 - Parent Row Source Operation significantly less than Child RSO
 - Filtering occurs after data read
 - Unindexed Predicate
 - Predicate unable to use index
 - Optimizer choosing not to use index

Secondary Throwing

- Easy to spot

Rows	Row Source Operation
798	SORT ORDER BY
309849	NESTED LOOPS
798	TABLE ACCESS BY INDEX ROWID
798	INDEX UNIQUE SCAN
309849	TABLE ACCESS BY INDEX ROWID
16083334	INDEX RANGE SCAN

16,083,334 -> 309,849
98.1% Throwing



Primary Throwing

- Not as easy to spot
- Compare Reads with Rows
- Look at
 - Longest Event(s)
 - Operations Related to the Event

Call & Event Information

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	332	0.01	0.03	0	0	0	0
Fetch	1138	13.15	857.26	111818	363714	0	806
total	1471	13.16	857.30	111818	363714	0	806

Elapsed times include waiting on following events:

Event waited on	Times Waited	Max. Wait	Total Waited
db file sequential read	111817	1.02	847.70
read by other session	2	0.01	0.01

- Most time spent reading data
- 111,818 disk/physical reads

Execution Plan

Rows	Row Source Operation
25	NESTED LOOPS (cr=319622 pr=110886 pw=0 time=845897463 us)
101584	MERGE JOIN CARTESIAN (cr=54023 pr=45915 pw=0 time=367527375 us)
253	TABLE ACCESS BY INDEX ROWID PR (cr=2406 pr=0 pw=0 time=20183 us)
1882	INDEX RANGE SCAN PRPRC_X (cr=999 pr=0 pw=0 time=5333 us)
101584	BUFFER SORT (cr=51617 pr=45915 pw=0 time=367464991 us)
99585	TABLE ACCESS BY INDEX ROWID OPR (cr=51617 pr=45915 pw=0 time=439842435 us)
99585	INDEX RANGE SCAN ORPR_CPR_X (cr=1525 pr=772 pw=0 time=7535234 us)
25	TABLE ACCESS BY INDEX ROWID ORD (cr=265599 pr=64971 pw=0 time=478493151 us)
58724	INDEX UNIQUE SCAN XPKORD (cr=203662 pr=28716 pw=0 time=147352908 us)

45915 + 64971
110886 physical reads
99.17% of statement I/O



Review

- Know The Systems
- Learn Tools
 - Know how to use the free and low level tools
 - Focus on efficiency
- Combine into Routines
 - Focus first on easiest and most painful issues
 - Keep refining