First Failure Fault Diagnosis

Oracle 11*g* Features and Novel Approaches

Jeremiah Wilton

ORA-600 Consulting

ORA-600 CONSULTING

Jeremiah Wilton

- Amazon.com's first DBA 1997-2004
- Working with Oracle since 1994
- Owner, ORA-600 Consulting

  www.ora-600.net

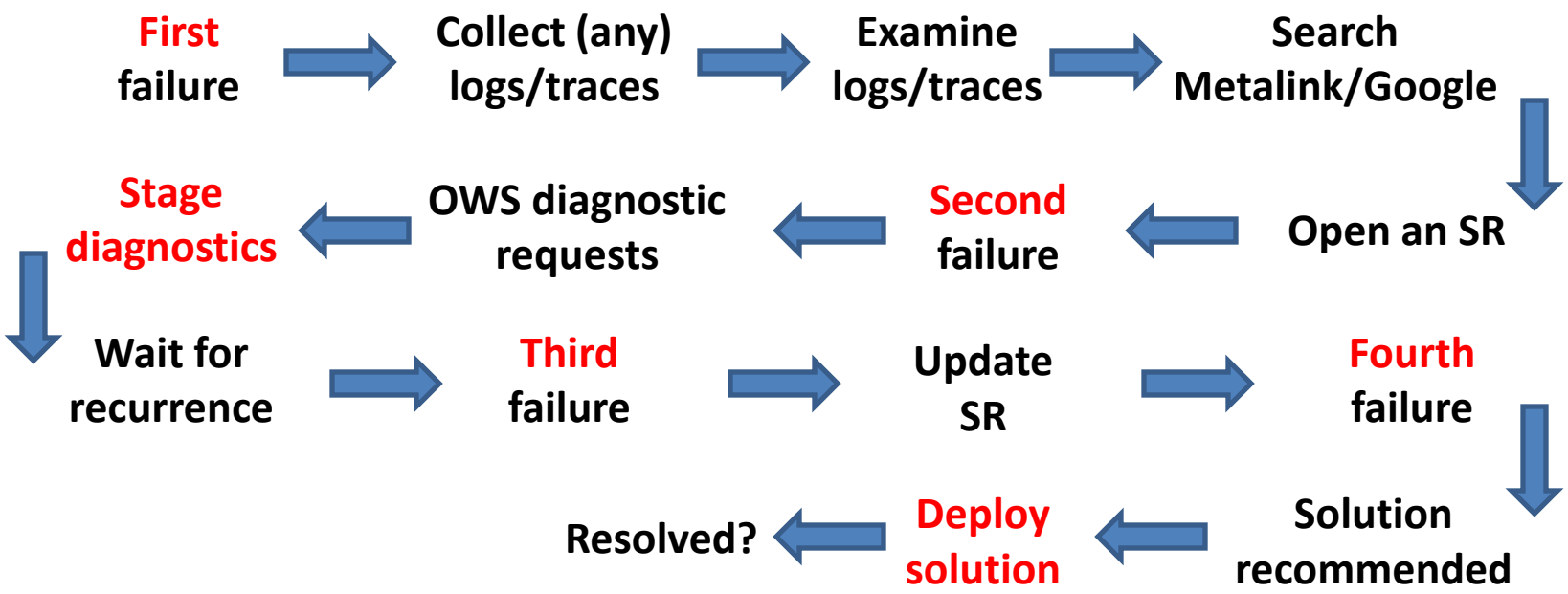
  – Architecture, scaling, performance
  – Availability, stability, complex recovery
  – Training, seminars, recruiting
- University of Washington lecturer
- Independent Researcher

ORA-600 CONSULTING

Something strange I learned at Amazon:

- Databases will inevitably fail
- There are two sides of the availability coin:
  - Availability technologies/strategies - prevent outages
    - RAC
    - Data Guard
    - Replication/Streams
    - Application design
  - Diagnostic capabilities - anticipate, diagnose & prevent *inevitable* outages
    - ASH/AWR
    - ADDM
    - Logs, dumps, traces
    - **11*g* Diagnosability**

Anatomy of a typical Amazon failure: Fourth (Fifth) Failure Diagnosis:



Oracle problems can seem chaotic

- Cryptic/alarming alert log messages
- Copious dump files, illegible content
- DBA inability to differentiate serious problems
- Lack of diagnostic speed, skill
- Pressure on DBAs to resolve problems fast
- Multiple problems in the same timeframe
- Increase in problems with scale
- Early feature/version adoption

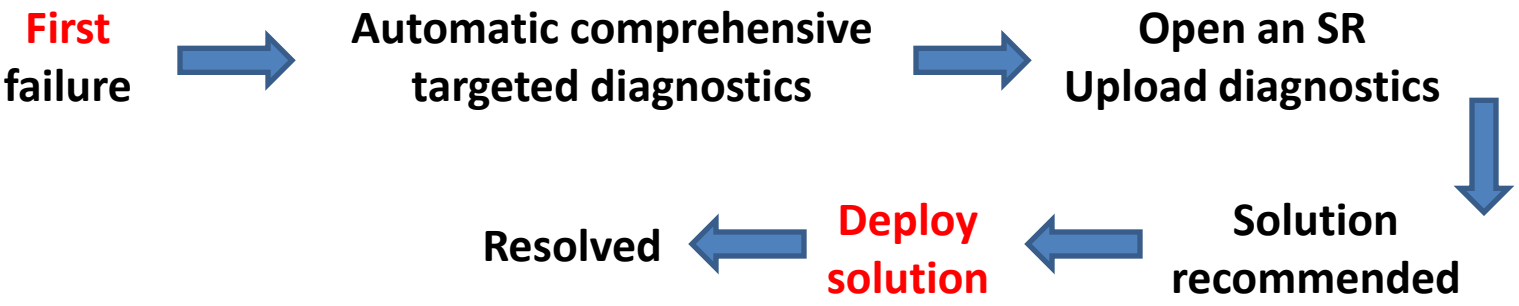
What about *Unbreakable*?

- … means providing a rich toolset
  - Availability
  - Manageability
  - Scalability
  - Security
- Every version has provided better diagnostics
  - 6.x: dumps, traces, timed statistics
  - 7.x: wait events
  - 8.x: Statspack
  - 10*g*: ASH/AWR/ADDM
  - 11*g*: Self-diagnosability

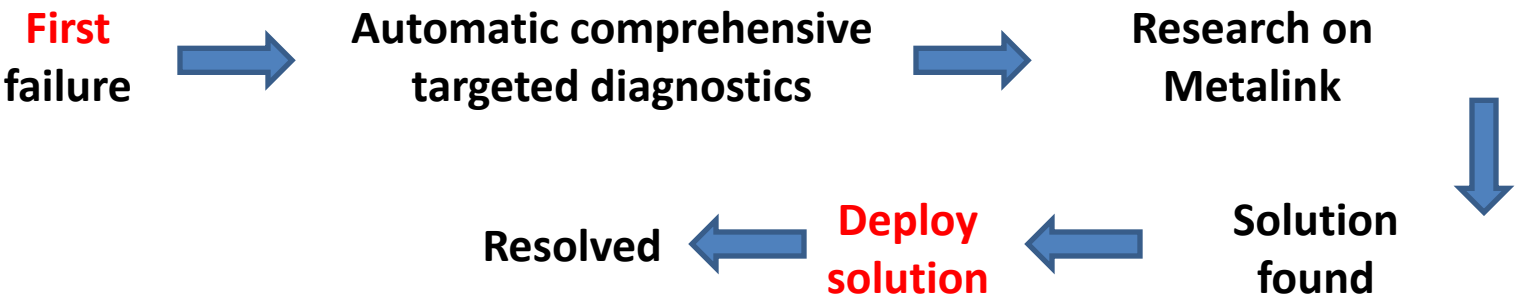
First Failure Diagnosis: Enabled by high-quality diagnostics

- Diagnose problems on first occurrence
- Targeted instead of general diagnostics
- Deploy solutions to problems before second occurrence
  - Avoid repeated events/outages
- Improve availability & stability
- Reduce DBA work / stress per incident
- Streamline the support process
- Proactively detect and diagnose problems before they become critical

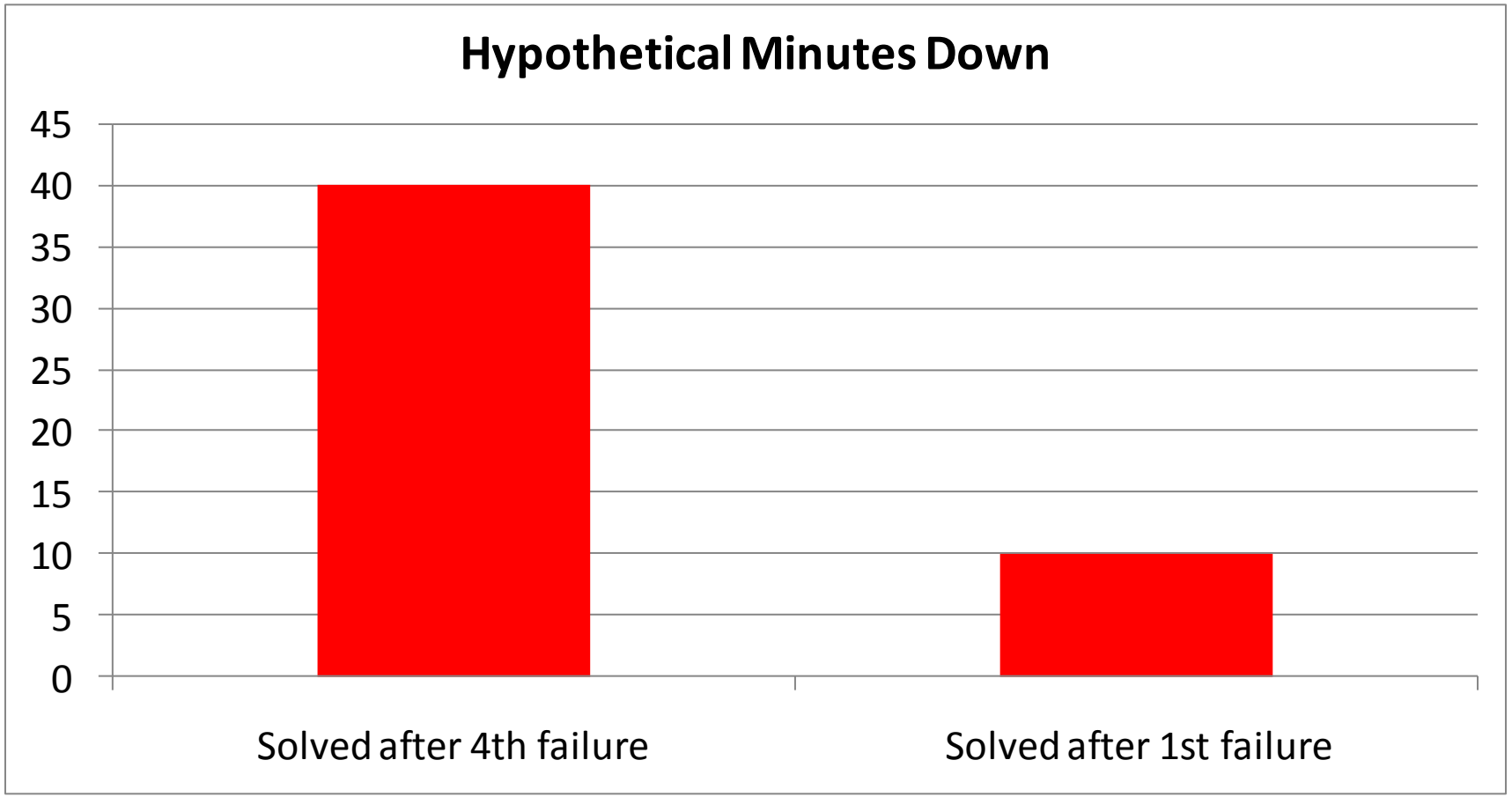
First Failure Diagnosis



First Failure Diagnosis



Problem downtime and FFD

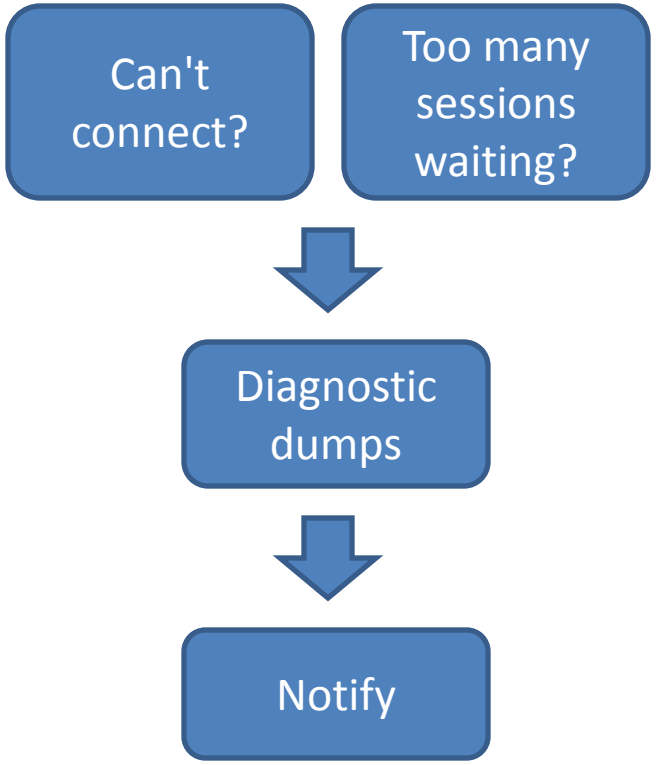


Rapid resolution induces a virtuous cycle

- More bugs found and addressed sooner
  - Even single occurrences
- Means 11.1.0.7 benefits from FFD
  - More bugs addressed through quality diagnostics
- More stability earlier in version cycle
- Increased customer confidence to upgrade
- Narrower version base (more customers 11*g*)
- Better dev and support ability to focus on 11*g*
- Higher quality of support and new versions

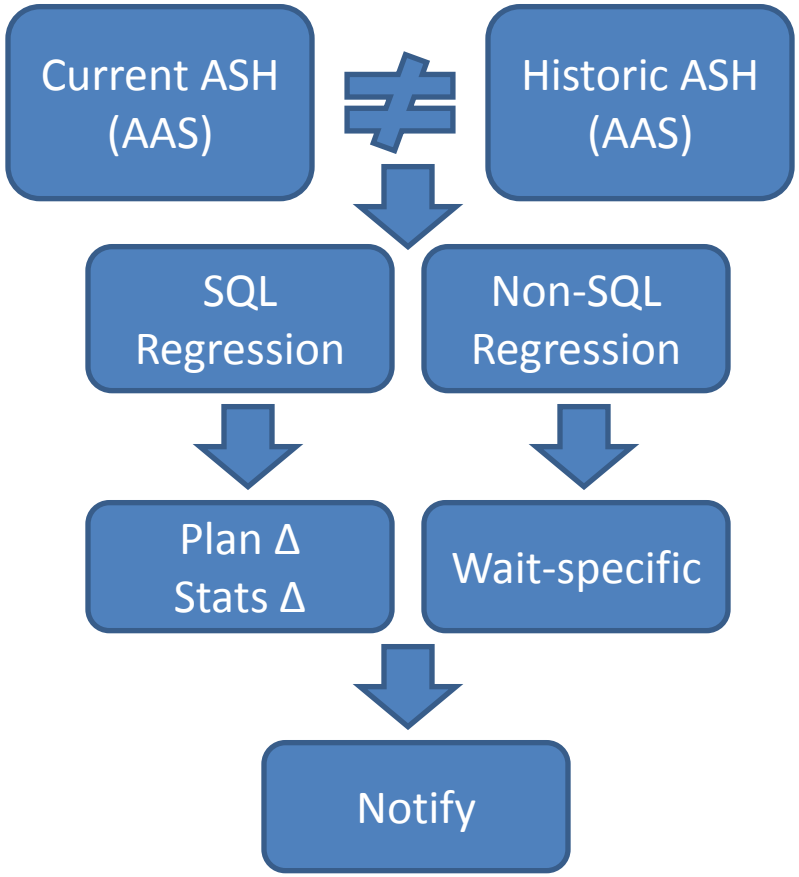
My client's pre-11*g* use of FFD: Instance hang detection

- External job determines if instance is hanging by:
  - Ability to connect/run
  - Percent of sessions
    - Waiting on non-idle waits
    - In non-instrumented waits
    - Waiting compared to previous ASH periods
- On detecting a hang:
  - Connect prelim/direct_access
  - Spool ASH data out
  - Diagnostic dumps
    - hanganalyze
    - systemstate
    - processstate
  - Alert with relevant dumps



My client's 10*g* extension of FFD: Database performance problem

- Job compares current ASH data to ASH data from past AWR time periods
- On detecting statistical difference:
  - Finds culprit SQL or event
  - Drills in with changed plan or event details
  - Alerts with relevant diagnostic data



Drilling in to specific waits

- Enqueues (blocking locks)
  - Blockers & waiters
  - Blocker state and further drilldown
- Internal resources (latches, library cache)
  - hanganalyze
  - `v$wait_chains`
  - Head-of-chain state and further drilldown

FFD Opportunities Outside the Diagnosability Infrastructure

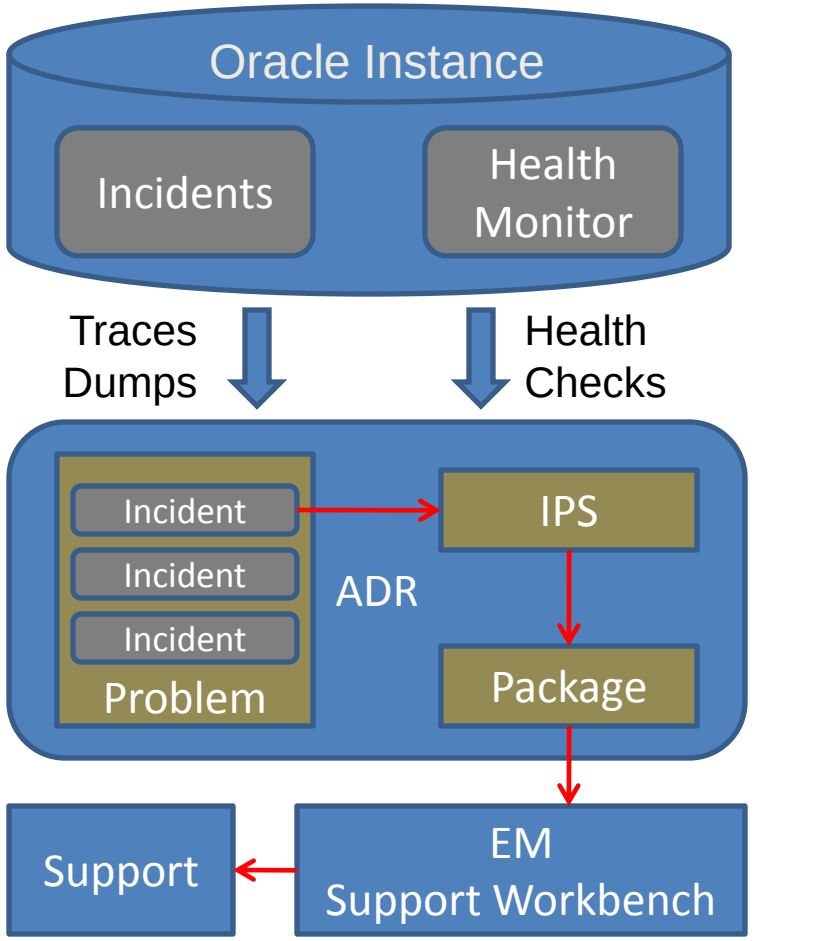
- Oracle-supplied
  - Alert log
  - Other logs
  - On-demand dumps/traces
  - On-error dumps/cores
  - RDA
  - Waits/ASH/AWR/ADDM
  - (g)v$ views
  - Prelim/direct_access
  - Some reactive/targeted
    - ORA-60
    - ORA-1555
  - Support
- Customer-supplied
  - Crash packs
    - Scripts to obtain set diagnostics before emergency restart
    - Systemstate x 3
    - Hanganalyze
  - DBA skill
  - FFD-aware monitoring
  - Self-collected history
    - Timed statistics
    - Session waits
  - Simulated ASH
    - www.ashmasters.com
  - Direct-attach performance tools (history)

11*g* Diagnosability Infrastructure

- Background processes/threads `DIAG`, `DIA0`
  - New in 9*i* and 10*g* RAC; 11*g* single-instance also
- Improved, standardized, targeted dumps & traces
- Automatic Diagnostic Repository
- Incident Packaging Service
- Support Workbench
- Hang detection
- Health checks
- Data Recovery Advisor
- SQL Test case builder
- Incident flood control

Core ADR Concepts

- Incident
  - Single occurrence
  - Critical error
    - Ex: `ORA-07445, ORA-00600`
  - Health check findings
    - Dictionary, block corruption
- Problem
  - All occurrences of a given issue
    - Ex: `ORA-600[keltnfy-ldminit]`
- Package
  - All material related to a particular incident
    - Trace files, logs
    - Reports and diagnostics
  - Ready to submit to support



ADR vs. RDA

- ADR is a big improvement over RDA
  - RDA was not targeted: kitchen sink approach
  - Large payload, much material unrelated to issue
  - Runs long after an issue occurs/resolves (irrelevant)
  - Leaves payload on DB server - issue for high-security
- ADR and 11*g* Diagnosability improve on this by:
  - Obtaining targeted diagnostics on a problem
  - Packaging diagnostics for upload
    - Everything Support needs
    - Less scrambling for the DBA
  - Providing a secure path for upload from DB host (EM)

Interacting with ADR

- ADRCI
  - `% $ORACLE_HOME/bin/adrci help`
- EM Support Workbench
  - Database Instance > Support Workbench

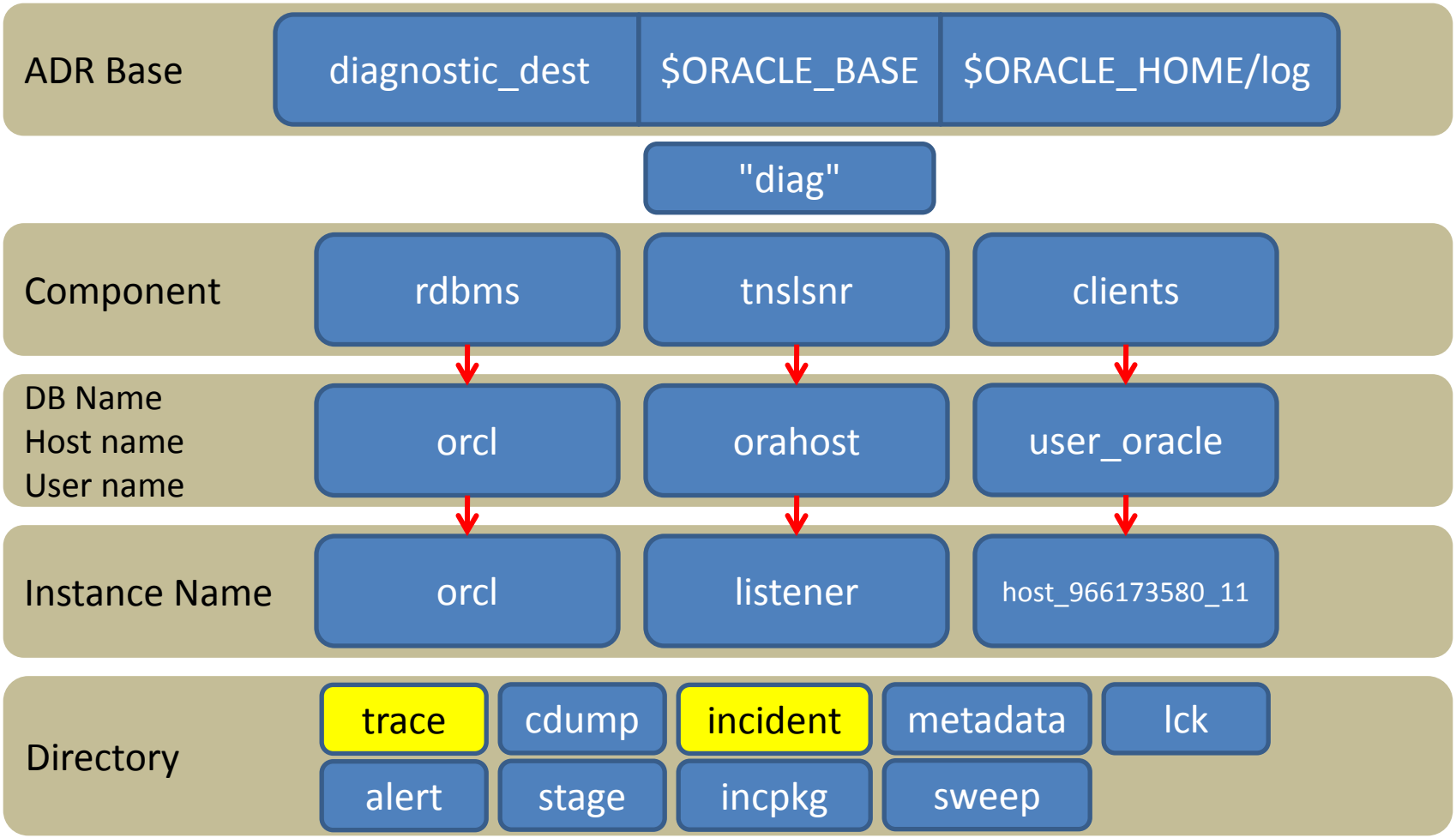
- Manage all aspects of ADR
  - Space management
  - Repository selection
  - Alert log query/watching
  - Health Check results
  - Incident and package management

Which errors trigger an incident?

See 11.1.0.7 v$diag_critical_errors

ORA-00227: corrupt block detected in control file: (block %s, # blocks %s)
ORA-00239: timeout waiting for control file enqueue: held by '%s' for more than %s seconds
ORA-00240: control file enqueue held for more than %s seconds
ORA-00255: error archiving log %s of thread %s, sequence # %s
ORA-00353: log corruption near block %s change %s time %s
ORA-00355: change numbers out of order
ORA-00356: inconsistent lengths in change description
ORA-00494: enqueue%s held for too long%s by '%s'
ORA-00600: internal error code, arguments: [%s], [%s], [%s], [%s], [%s], [%s], [%s], [%s]
ORA-01578: ORACLE data block corrupted (file # %s, block # %s)
ORA-03137: TTC protocol internal error : [%s] [%s] [%s] [%s] [%s] [%s] [%s] [%s]
ORA-04030: out of process memory when trying to allocate %s bytes (%s,%s)
ORA-04031: unable to allocate %s bytes of shared memory (\"%s\",\"%s\",\"%s\",\"%s\")
ORA-07445: exception encountered: core dump [%s] [%s] [%s] [%s] [%s] [%s]
ORA-29740: evicted by member %s, group incarnation %s
ORA-32701: Hang detected (11.1.0.7)
ORA-32703: deadlock detected: %s (11.1.0.7)
OCI-03106: fatal two-task communication protocol error
OCI-03113: end-of-file on communication channel
OCI-03135: connection lost contact

ADR Directories



ADR Space Management

- Incident trace files are purged after 30 days &
- Incident metadata is purged after 1 year

  unless:
  - Purge disabled for the incident
  - Packaged incident uploaded < 30 days ago
- Reduces bloat from many trace files
  - /bin/rm: Argument list too long
  - shell, kernel limits from huge numbers of files

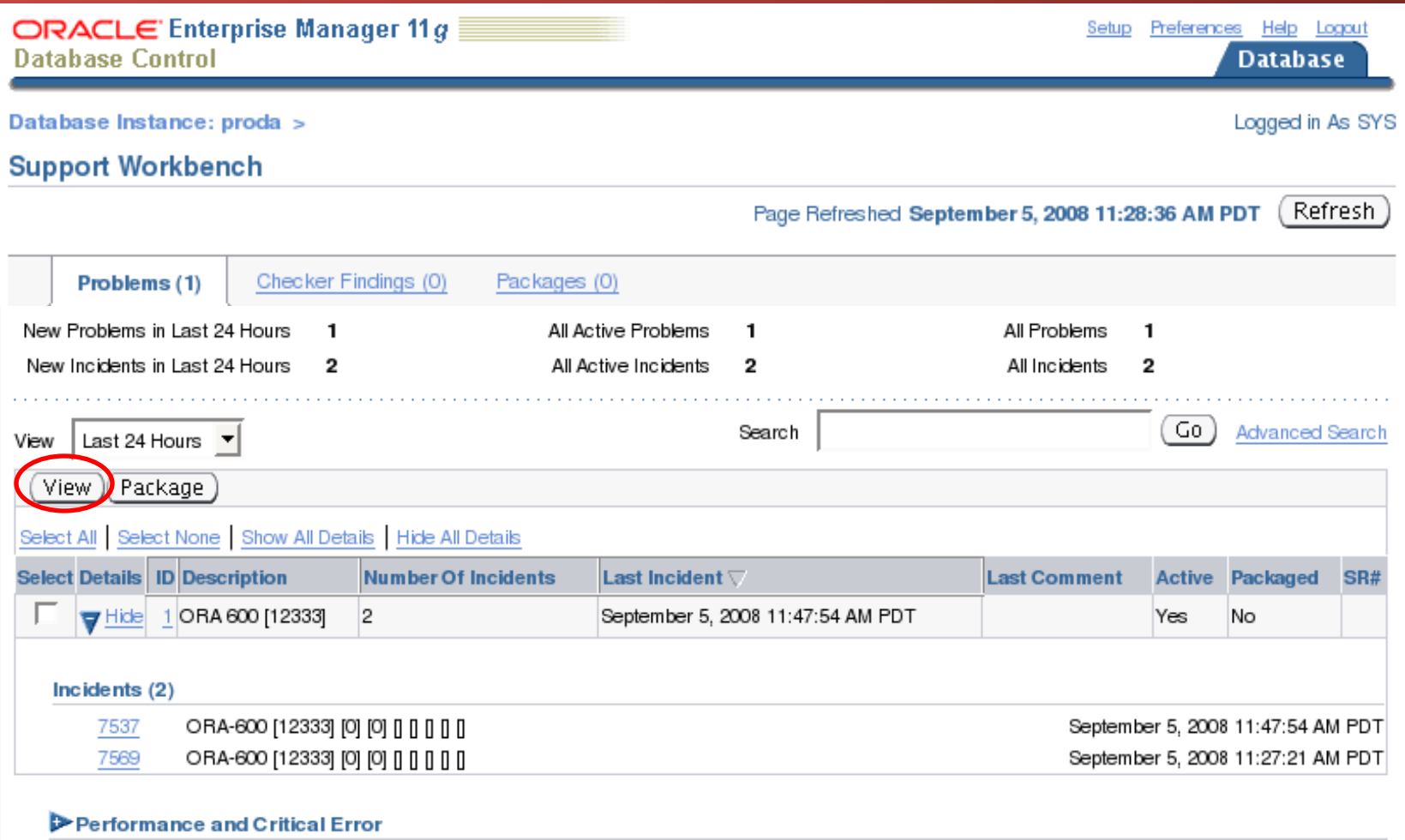
ADR Flood Control

- Five dumps per problem per hour
- Incidents continue to be recorded unless:
  - many incidents in a short period
- Reduces load from excessive dumping
- Preserves space for further diagnostics

FFD and 11*g* Diagnosability

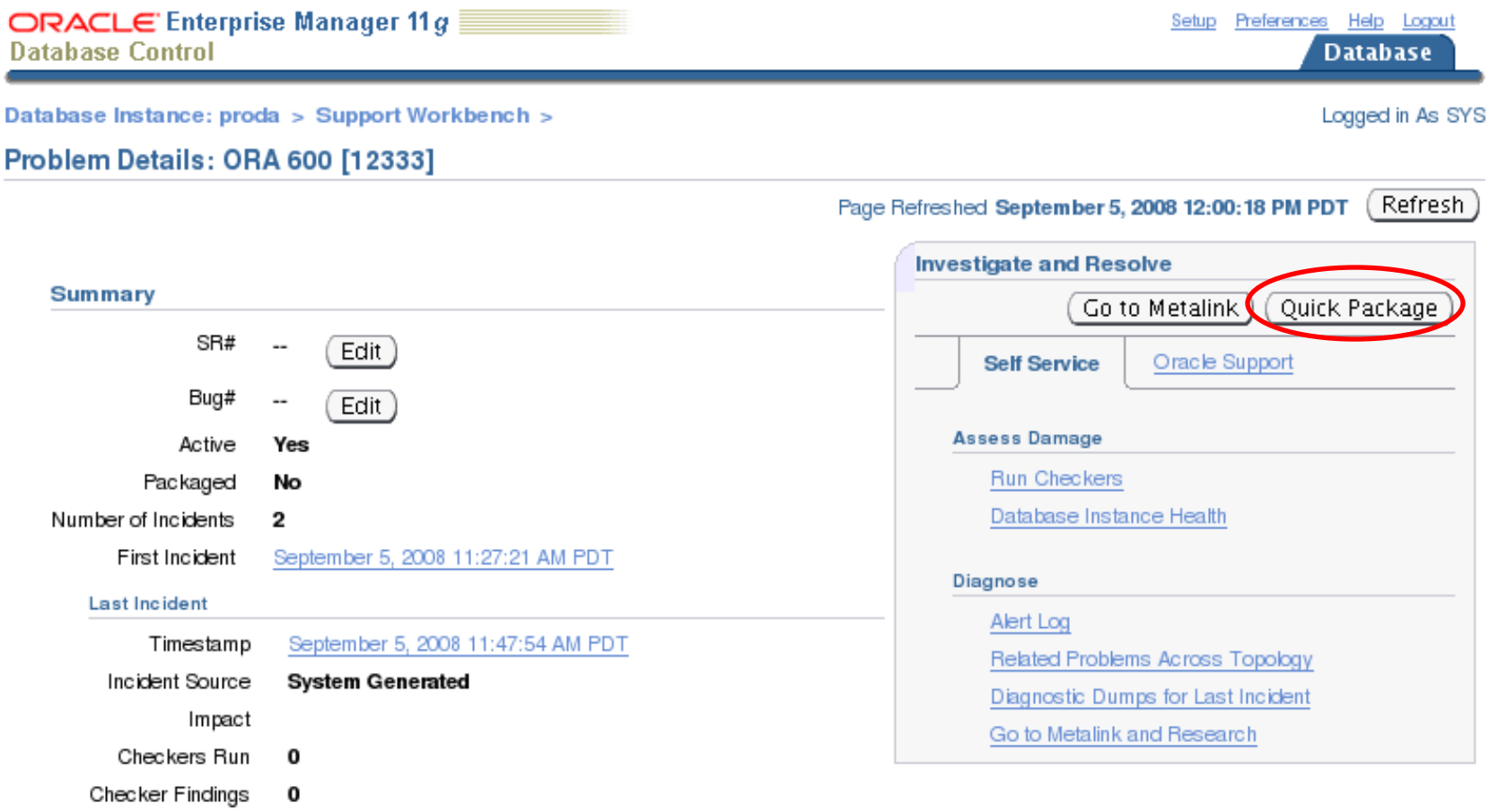
- Hang/deadlock detection
  - Certain hang scenarios
  - Hang chains
  - Inter-component deadlocks
- Support Workbench
  - Manage/package incidents and problems
  - Upload to support
  - Initiate problems/incidents on demand
- Health checks
  - Mostly to scope corruption currently
  - Extensible architecture
- ADR/dumps
  - Trace improvements
    - Targeted
  - Better trace organization
  - Shorter time to submit
  - Tracking of problems

Support Workbench:
All problems/incidents at a glance



ORA-600 CONSULTING

Support Workbench: Packaging and submitting a problem



ORA-600 CONSULTING

Automatic hang detection

- 11.1.0.7: CLUSTER_DATABASE=TRUE (RAC) only
- Messages in DIA0 trace:

```
*** 2008-09-07 15:51:33.336
About to produce dumps
```

- DIA0 sub-trace: (<sid>_dia0_<pid>_1.trc)

```
A possible hang has been detected in your system…
*** 2008-09-07 15:51:33.339
Verified Hangs in the System
   Hang_ID Type Status   Root_cnode Root_sess #hung_sess
   ------- ---- -------- ---------- --------- ----------
           Cnode SessId OSPID Event
           ----- ------ ----- -----
         1 HANG VICSELTD           0        88          3
               0     89 17435 library cache lock
               0     86 17378 enq: TM - contention
               0     88 17176 SQL*Net message from client
Victim Information
  Hang ID = 1, Victim (cnode:sessid:ospid) = 0:88:17176 (non-fatal process)
```

- "Victim" is actually blocker

Health Checks: 11.1.0.6

- On-demand
  - dbms_hm.run_check()
  - OEM
- Reactive
  - Result of an incident
  - Corruption or similar
- Results in:
  - gv$hm_run
  - gv$hm_finding
  - gv$hm_recommendation
  - dbms_hm.get_run_report

```
SESS1> select name from v$hm_check;
NAME
--------------------------------
HM Test Check
DB Structure Integrity Check
Data Block Integrity Check
Redo Integrity Check
Logical Block Check
Transaction Integrity Check
Undo Segment Integrity Check
All Control Files Check
CF Member Check
All Datafiles Check
Single Datafile Check
Log Group Check
Log Group Member Check
Archived Log Check
Redo Revalidation Check
IO Revalidation Check
Block IO Revalidation Check
Txn Revalidation Check
Failure Simulation Check
Dictionary Integrity Check
```

Reactive health check example

```
ORA-01578: ORACLE data block corrupted (file # 5, block # 12)
Bad header found during buffer read

DDE rules only execution for: ORA 1110
----- START Event Driven Actions Dump ----
---- END Event Driven Actions Dump ----
----- START DDE Actions Dump -----
----- DDE Action: 'DB_STRUCTURE_INTEGRITY_CHECK' (Async) -----
Successfully dispatched
----- (Action duration in csec: 0) -----
----- END DDE Actions Dump -----

SQL> select name, check_name from v$hm_run where src_incident=42253;
NAME                             CHECK_NAME
-------------------------------- --------------------------------
HM_RUN_321                       Data Block Integrity Check
```

Novel failure detection: Beyond 11*g* Diagnosability

- User-built adaptive monitoring can:
  - Broaden scope of problems
    - Use your own SRs as a guide to what you should be detecting
    - Not just errors and corruptions
      - Abnormal activity
      - Abnormal waits
      - Based on statistical analysis of AWR
  - Broaden scope of hangs
    - Include very slow progress, not only true hangs
    - Include non-chained (statistical and signature-based hangs)
    - Use ASH data through prior incidents as guide (hang signature)
  - Allow user-defined actions/diagnostics
    - Your own SQL reports
    - Dump ASH buffer via prelim/oradebug direct_access

Diagnosability at ORA-600 client sites

- Out of 18 active clients:
  - Nine have 11*g* in testing for production in Q109
  - Nine are using some sort of test to scale
- In 11*g* test sites Diagnosability features have:
  - Helped reduce time to issue resolution
  - Reduced testing cycle timeline
  - Integrated seamlessly with existing monitors
    - *_dump_dest still populated values
    - text alert log still in background_dump_dest

Real Application Testing + Diagnosability Customer 10.2.0.4 to 11.1.0.6 upgrade example

- Major EU online retailer
  - 25 production databases, 3 RAC clusters
  - RHEL4, 64-bit, ASM, Streams,
  - DataGuard (Logical & Physical)
  - Heavy OLTP, low downtime tolerance
- 90-day test to prod. timeline (pre-holiday)
  - Applying Database Replay of captured workload to test clones
  - Uncovered many issues
  - ADR enabled:
    - Repeated incident to patch time of under 2 hours
    - Incident to bug to patch cycle under one week
    - Many problems solved by customer instead of raising an SR

Real Application Testing + Diagnosability
Customer 10.2.0.4 to 11.1.0.6 upgrade example

- Database replay encountered bug 6073325:

```
SQL> select 1 from sys.table_privileges tp, user_objects uo
     where tp.grantee in
        (select 1 from sys.dba_role_privs
         connect by prior granted_role = grantee
         start with  grantee = 'SCOTT');

ERROR at line 1:
ORA-00600: internal error code, arguments: [kkqcbydrv:1], [], [], [], []

Thu Sep 18 11:48:28 2008
Errors in file /opt/oracle/diag/rdbms/test02/test02/trace/test02_ora_4219.trc  (incident=48653):
ORA-00600: internal error code, arguments: [kkqcbydrv:1], [], [], [], [], [], [], []
Incident details in: /opt/oracle/diag/rdbms/test02/test02/incident/incdir_48653/test02_ora_4219_i48653.trc
Thu Sep 18 11:48:42 2008
Trace dumping is performing id=[cdmp_20080918114842]
Thu Sep 18 11:48:44 2008
Sweep Incident[48653]: completed
```

- Packaged/uploaded to new SR in two minutes
- Received response in less than 30 minutes
  - Known bug
  - Available workaround

Risks/Rewards

- I advise my clients to leverage this fully
- No major structural/code differences
  - Largely a set of augmentations to the code
  - Based on many preexisting structures
- Highly stable under overwhelming load
  - Improves failure characteristics
  - Flood control allows compromised system to continue
- Represents a low level of risk

Conclusions

- 11*g* Diagnosability improves the chances of early diagnosis
  - New tools and automatic capabilities
  - Faster more effective problem resolution
  - A framework for even better diagnostics in future
- Customers can take their cue from Oracle
  - Build their own diagnostics on same principle
  - Leverage ADR, IPS and health checks in their own monitoring
- Creates beneficial environment for
  - Availability
  - Stability
  - Oracle service quality
  - Customer confidence