

Exploring ADR in Oracle Database 11g



By Jeremiah Wilton, ORA-600 Consulting

It's inevitable that at some time failures will occur in a database, particularly when the database has issues with self-reporting of faults. DBAs often lack the necessary time, knowledge, and/or skills to resolve an outage. For this reason, Oracle has introduced the Automatic Diagnostic Repository (ADR) in Oracle Database 11g. Jeremiah Wilton provides an introduction to this exciting new feature.

Effective strategies for availability and stability are predicated on the knowledge that failures will inevitably occur. The art and science of stability involves crafting effective strategies for overcoming those inevitable failures. Even “unbreakable” Oracle generally refers not to an individual instance, but to Oracle's overall availability strategy, using multiple instances and multiple databases.

Over the life of the server product through Oracle Database 10g, Oracle has led the industry in providing rich instrumentation and diagnostic facilities. Chief among these are wait events, timed statistics, trace events, and dynamic performance views.

One area in which Oracle has not historically provided best-of-breed technology is in self-reporting of faults. For customers encountering faults such as hangs, dead processes and ORA-600/7445 issues, resolution has been a somewhat tedious process. The process often involves opening an SR with Oracle Support, running the Remote Diagnostic Assistant (RDA) to obtain information to send to Oracle, inspecting the alert log and any trace file, then uploading files one at a time to MetaLink.

The RDA/trace upload process has multiple inadequacies. RDA, rather than providing targeted diagnostics specifically tailored to the type of failure that occurred, creates a huge payload containing a wide variety of information, most of which is certain to be unrelated to the problem. Besides requiring extra time and space to provide to Oracle, the mere existence of all the irrelevant information in the RDA payload often leads to Oracle Support analysts getting sidetracked on unrelated or irrelevant issues. In addition, the RDA reporting is often done long after the conditions that led to the failure have passed, making the RDA output even less relevant to the resolution of the problem.

Through Oracle Database 10g, the execution of specific diagnostic events and traces in relation to a problem is up to the DBA or is performed at the direction of Oracle Support. In the heat of an unplanned outage, many DBAs do not have the time, knowledge, and skills to obtain traces and dumps relevant to the problem. For others, the event passes or is resolved by instance restart, and they never obtain the necessary diagnostics for root cause detection. Potentially, the same problem can plague a customer repeatedly, causing multiple unplanned outages, before assistance from Oracle support results in any resolution.

Introducing ADR

In part to address the lack of good self-reporting of faults, Oracle has introduced the Automatic Diagnostic Repository (ADR) in Oracle Database 11g. ADR consists centrally of

- A set of directories where trace information is automatically stored
- A new concept called an *incident*, which may consist of one or more traces
- A packaging service for generating a zip file to upload to Oracle Support
- A command line interpreter for managing incidents and packaging called *adrci*.

A Quick Tour of adrci

The apparent original intention of ADR was to do away completely with the background and user dump directories, and to allow access to trace and alert data only through the ADR command interpreter or Enterprise Manager. Early beta releases had only an XML-based alert log. However, thanks to Oracle's responsiveness to the comments of beta-program participants, the alert log has reappeared in plain text in a recent beta release.

If you prefer, you can still access the alert log, trace files, and listings of recent dump and trace incidents through the *adrci* tool. One nice feature of *adrci* is the ability to query the alert log using SQL-like predicates, which will be useful for the *grep*-impaired, or those on a windows platform:

```
adrci> show alert -p "message_text like '%alter%'"
2007-06-27 23:04:39.232000 -07:00
alter database open
2007-06-27 23:04:58.786000 -07:00
Completed: alter database open
2007-06-27 23:11:33.931000 -07:00
alter database open resetlogs
2007-06-27 23:19:57.954000 -07:00
Completed: alter database open resetlogs
```

The *adrci* utility also provides a facility for watching current messages in the alert log, like the Unix *tail* command. This will be particularly helpful for those on platforms like Windows that generally lack a utility like *tail* that constantly shows the most recent lines to be written to a text file as they are written:

continued on page 18

```
adrci> show alert -tail -f
2007-08-01 10:53:08.177000 -07:00
Thread 1 advanced to log sequence 134
  Current log# 4 seq# 134 mem# 0: /mnt/usb/oracle/oradata/DEV01/online/og/o1_
mf_4_386nv7j_.log
2007-08-01 11:20:38.289000 -07:00
Thread 1 advanced to log sequence 135
  Current log# 3 seq# 135 mem# 0: /mnt/usb/oracle/oradata/DEV01/online/og/o1_
mf_3_386nv4wn_.log
```

Incidents and Packages

With Oracle Database 11g ADR, when Oracle encounters an error that results in a trace or dump, a trace file is written to either the background or user dump destinations as in previous releases. However, it is a short file containing only summary information and automatic diagnostic logs. The full trace dump is written to the ADR incident-specific directory. In addition, ADR also:

- Records the occurrence as an incident in the ADR metadata directory. This metadata is stored externally to the database so that it is available even when the database is down or unavailable.
- Runs additional diagnostics specific to the type of failure that took place.

If a DBA decides to upload the information related to the incident to Oracle Support, the adrci command line provides easy commands that create a zip file with everything Support needs. Under ADR, this is referred to as a *package*:

```
adrci> show incident

ADR Home = /mnt/usb/oracle/diag/rdbms/dev01/dev01:
*****
INCIDENT_ID PROBLEM_KEY                CREATE_TIME
-----
13124      ORA 7445 [kksMapCursor()+287]         2007-07-31 14:40:15.940839 -07:00
11417      ORA 7445 [_d_l_sysinfo_int80()+2]     2007-07-27 14:12:47.352902 -07:00
11393      ORA 7445 [kgesiv()+21]                2007-07-02 10:20:55.785032 -07:00
11329      ORA 7445 [kgeadp()+171]              2007-07-01 20:03:04.304846 -07:00
```

Generating an uploadable zip file for Oracle support is trivial with adrci. First, create a “logical” package, which is just a set of metadata listing all the files that will be part of the incident package:

```
adrci> ips create package incident 13124
Created package 1 based on incident id 13124, correlation level typical
```

Then, generate the “physical” package, a zip file containing all the files relating to the incident:

```
adrci> ips generate package 1 in /tmp
Generated package 1 in file /tmp/ORA7445kk_20070801135443_COM_1.zip, mode
complete
```

The resulting zip file for this core dump incident is about 16M in size. Examination of the resulting zip file shows that it contains the XML alert log, the trace file from the core dump, the summary trace from the background_dump_dest, trace files from *all* background processes that happened to

generate trace data around the time of the incident, all of the incident metadata, and a manifest. Notably, it is possible to add additional files as needed to the package using adrci prior to generating the physical package.

Reactive First-Time Diagnostics

An important feature of ADR, automatic diagnostics, is meant to reduce diagnostic and resolution time. When encountering a new problem, especially in a production environment, time is of the essence. The DBA can ill afford to spend several minutes collecting diagnostics while the application is unavailable or impaired. Most of the DBA's effort must be directed toward resolving the problem as quickly as possible. Often, such as in the case of restarting Oracle, resolving the problem leaves the DBA with no good data for determining root cause. For this reason, good data on faults is often not obtained until a second or third occurrence.

The documentation states that on encountering a critical error in Oracle Database 11g, the kernel will try to obtain additional diagnostics. Currently, the types of diagnostics taken as a result of certain events are not yet documented, but we can presume that in general, they are specific to the type of fault. All resulting traces and dumps are also included in the incident and any packages the DBA might create based on that incident.

Hang Detection

Two new background processes, DIAG and DIA0 are responsible for automatic diagnostics and hang detection. Upon detection of a hang, the Hang Manager runs a node dump, as well as a systemstate dump. By default, the levels of these dumps are set at 0 and 1 respectively. The dump level is adjustable via the `_hm_analysis_oradebug_node_dump_level` and `_hm_analysis_oradebug_sys_level` hidden parameters.

Hanging situations often involve long chains of waits, in which several sessions are waiting on each other for a variety of resources, such as latches. A common occurrence in busy environments is a pileup on library cache resources. Such chains of waiting sessions often were challenging for a DBA to diagnose or resolve, because mapping the chains of resource holders required taking a systemstate or hanganalyze dump. Once dumped, the DBA had to interpret the trace output.

In Oracle Database 11g, not only are hangs automatically detected, but the diagnosis and mapping of long chains of waits is simplified. A new view, `GV$WAIT_CHAINS`, allows easy and simplified mapping of wait/resource dependencies. Hang Manager uses the data exposed in this view as one of the criteria for detecting hangs.

Other Features

Other new features that interact with the ADR, but are not central to the feature, are

- Trace format standardization
- A variety of new diagnostic reports and tools, called Health Checks
- Data Recovery Advisor for recommending repair strategies after a problem
- SQL Test Case Builder, for obtaining extensive information that could help reproduce a fault.

The Future

In ADR, we see a feature that holds out immense promise for eliminating days and weeks of diminished functionality at customer sites when a bug or other fault is encountered. When this feature reaches full maturity, it should:

- Allow user specification of what diagnostics to run upon encountering specific faults
- Effectively detect a variety of hanging situations
- Provide an easy way to upload ADR packages to Oracle Support without OEM or a Web browser.

ADR-Related Views

The following ADR-related views might be of interest:

- v\$wait_chains: Provides a mapping of dependent waits among sessions
- v\$hm_check: A list of available health checks
- v\$hm_check_param: A listing of the parameters set via DBMS_HM for the various health checks
- v\$hm_run: Records on the time, mode, cause, and status of each run of a health check
- v\$hm_finding: Findings from health check runs
- v\$hm_recommendation: Recommendations based on health check findings
- v\$diag_info: The locations of ADR files on disk
- v\$database_block_corruption: Block corruptions found by health checks that detect block corruptions
- v\$corrupt_xid_list: A list of transactions IDs found corrupt by health checks.

ADR-Related Packages

In addition, there are two ADR related packages:

- dbms_hm: Runs health checks and displays resulting reports
- dbms_sqldiag: The API to the SQL Test Case Builder and Diagnosability features.

Conclusion

In every major database release, Oracle has been adding many useful features. In Oracle Database 11g, we believe that ADR will prove to be one such very practical and useful feature.

■ ■ ■ About the Author

Jeremiah Wilton was Amazon.com's first DBA. He is currently principal consultant at ORA-600 Consulting in Seattle, where he provides novel solutions to problems of scalability and availability. A specialist in complex recoveries, he enjoys contributing to the Oracle user group community by writing articles and presenting at conferences.