Custom Monitoring your Database with PL/SQL

William Schott

NYOUG December 8, 2009

Know Your Audience

Let’s see who's in the audience today.

- How many are DBAs? Developers?
- How many have written PL/SQL?
- How about a package and body?
- How many have created a DBMS_JOB?
- Who knows what SQL/PL is?

Why This Presentation?

- Repetitive tasks are tedious and boring.
- Not a good use of your time
- Yet are often essential in certain circumstances
  - Watching for a problem
  - Monitoring for a condition or event
  - Troubleshooting

So What’s the Solution?

Automation via DBMS_JOB

- Runs whenever database is open
- Access to PL/SQL procedures for logic

Not without its difficulties:

- Input/output difficult
- Job can “break”

Design Concepts

PL/SQL package implementation

- Compiled code for rapid execution
- public and private procedures
- Multiple procedures in one block of code
- Output from DBMS_Job can be difficult
    - Resolved by private “sendmail” procedure”

Design Concepts

Table driven “sendmail” procedure is essential

- Monitoring useless without notification
- Email is ubiquitous - not site specific
- Can also talk to most pagers and cell phones
- Worker procedures call private Sendmail with Message_type

Sendmail Details

- Implemented as a private procedure inside body
- Controlled by 3 column table

```
create table {unique name to package}
   (email_address     varchar2(50) not null,
   message_type       varchar2(8) not null,
   message_body       varchar2(250) not null) ;
```

- Supports multiple email addresses per message_type
- Message_body contains fixed text

Sendmail Details

- Opens cursor based on Message_type
- Has internal mail_line procedure to write 1 line
- Simple version in Appendix 2 of paper
    - Hardcoded subject and message body
- More complex, multi-line version in Appendix 1
    - Accepts an array for the message body
- Well tested - just use “as is”

Case Studies

Why use Case Studies?

- Problem/Solution format
- Meant to be examples, not finished modules
- Intended to be thought provoking
- Good teaching technique

Case Studies Covered

1. Row Locking and Waiters
    a. Row locks on a table
    b. Enqueue waits
2. Watch for an individual SQL statement
3. Standby Database Log Shipping
4. Detecting Missing Standby Logs

Monitoring Row Locking

Problem:

- Observed a condition where Websphere transaction was updating MATUSETRANS table but not issuing Commit and going idle
- Infrequent occurrence and hard to track to log files unless detected quickly
- Needed accurate session information

Monitor Row Locking

Solution:

- Create a package and body to watch for idle sessions with lock on this table
- Create a DBMS_JOB to call it
- Prcedure to send email when situation detected

Monitor Lock Waiters

Problem:

- Occasional transaction backlog due to record locking
- Often caused by manual script with no COMMIT
- Required prompt identification to avoid major impact on application response time

Monitor Lock Waiters

Solution:

- Create a DBMS_JOB to run every minute, looking for Sessions waiting on an ENQUEUE
- Send email with data about blocking session, current event, and what locks it holds

Watching for SQL Stmt

Problem:

- A user is running a poorly performing statement that is tying up DB resources and timing out
- Need to quickly locate session data when it starts
- Causes poor JVM performance until killed

Watching for SQL Stmt

Solution:

- Create a DBMS_JOB to run every 5 minutes, a session running a SQL with specific HASH value
- Send email with data about that session so source can be traced in the log files.

Standby Log Shipping

Problem:

- Firewall timeouts throwing away IP traffic
- Slow internet connection to Standby
- Running “Mandatory” fixes firewall but puts Primary DB at risk of stalling due to lack of available redo logs

Standby Log Shipping

Solution:

- Set Log_Archive_Dest_2 = Mandatory
- A PL/SQL job to run every 5 minutes to watch “archived” status of redo logs. Possible actions:
    - Change destination to Optional if nearly full
    - Defer destination if < 2 available redo logs
    - Re-enable destination if Defer’d and caught up

Missing Standby Logs

Problem:

- When Standby “Optional” archive log may be skipped due to network error
- RMAN purges archive logs once backed up
- Result: Standby unable to FAL missing archive log

Missing Standby Logs

Solution:

- Monitor v$archived_log for redo logs sent to Dest 1, but not Dest 2
- Send email if archive log has not been sent to standby in over “n” minutes”

Concluding Comments

- I hope you found the use of Case Studies useful
- PL/SQL packages can do a lot!
- Does not take sophisticated programming
- Use my code as a template for your jobs
- Did this inspire some use cases in you job?

Contact Information

It’s been my pleasure to share this technique with you.

Bill Schott

DTE Energy

schottw@dteenergy.com