nocoug JOURNAL

Official Publication of the Northern California Oracle Users Group

VOL. 15, No. 3 · AUGUST, 2001 $15

Behind the Scenes at NoCOUG

by Lisa Loper

If you are like many members of NoCOUG, you attend quarterly conferences, enjoy the day, learn something new, visit with old friends, and then head back to work until the next quarterly meeting. All the while, the members of the NoCOUG board of directors continue to plan upcoming events and focus on issues that help achieve the group's mission of educating and representing Oracle users. A lot of this work occurs at our board meetings where we tackle lots of issues and have some fun along the way.

All board members are unpaid volunteers who give of their time, starting with one Saturday meeting and one evening meeting per quarter. Our meetings begin once we have a quorum, and we follow Robert's Rules of Order in running our meetings. We always make time for catching up with the pals we make on the board, but when the meet-

(continued on page 4)

Don't Miss Our Summer Conference on August 23!

We've lined up some top speakers like Gaja Vaidyanatha, author of *Oracle Performance Tuning 101*.

Read about Gaja on page 4.

TABLE OF CONTENTS

–ADVERTISERS–

Thank You!!!

I'd like to thank everyone who helped in the production of this issue of the *NoCOUG Journal*. You helped make my job as editor much easier and helped make the Journal more valuable to our readers.

A special thank you to:

Colleen Childers for acting as assistant editor: writing articles, proofreading, and assisting in the Journal production and mailing.

And a big thank you to all the authors whose work is contained in this issue. The generosity of your time in preparing articles and your mission to share knowledge are appreciated.

—Lisa Loper
NoCOUG Journal Editor

Other User Groups

Local

NorCalOAUG – Northern California Oracle Applications Users Group

- **Contact:** Michael Capelle (650) 562-1167
- **Email:** capelle@tru-course.com
- **Website:** www.norcaloaug.org

Sacramento

SacOUG – The Sacramento Oracle User Group

- **Contact:** Ravi Verma (916) 705-3261
- **Email:** ravi.verma@telcommand.com
- **Website:** www.sacoug.org

International

IOUG-A – International Oracle Users Group of the Americas

- **Website:** www.ioug.org

U.S. Domestic

OAUG – Oracle Applications Users Group

- **Website:** www.oaug.org

ODTUG – Oracle Development Tools User Group

- **Website:** www.odtug.com

Canvassing calls by employment recruiters to local chapter contacts is strongly discouraged.

Publication and Submission Format

The NoCOUG Journal is published four times a year by the Northern California Oracle Users Group approximately two weeks prior to the quarterly regional meetings. Please send your questions, feedback and submissions to: Lisa Loper, NoCOUG Journal Editor, at journal@nocoug.org.

The submission deadline for the November issue is October 1st. Article submissions should be made in electronic format via email if possible. Word documents are preferred.

NoCOUG does not warrant the NoCOUG Journal to be error-free.

Thank You, Genentech!

Genentech has generously offered to host our Fall conference to be held November 15.

Mark Your Calendars! ▲

NOCOUG BOARD

ADVERTISING RATES

Hail to the Chief!

by Colleen Childers

Running around the soccer field at lunchtime helps Vilin Roufchaie, NoCOUG president, have the stamina needed to help run our organization. In addition to being the Unix and DBA Team Lead at Cingular Wireless in Pleasanton, Vilin has the role of being the "general manager" of the Northern California Oracle Users Group.

Vilin strikes a thinker's pose.

The Bay Area, considered the most beautiful area in the world by Vilin, became his home back in 1985. He took the opportunity to come out west from Lafayette, Louisiana when, during some tough times in the area, he found himself without a job. Although Vilin is a world traveler, the Bay Area is where his heart is. The first position he held in the Bay Area was as a C programmer.

Vilin is currently in his second term as NoCOUG president. He's been involved with the group for four years now. You might wonder what gets our board members involved. Well, upon meeting Joel Rosingana, NoCOUG's current vice-president, Vilin just knew that NoCOUG was the place for him and took on his first role—finding presenters for the technical sessions.

Vilin has always enjoyed organizing and managing. He especially gets a kick out of being at the quarterly meetings and seeing all the efforts of the volunteers and board members come together for a successful conference. He strongly believes in giving NoCOUG members the quality they deserve in all areas—the website, quarterly meeting, and Journal. Vilin hopes that, in his tenure as president, he'll make an impact in terms of what the leaders of NoCOUG deliver to the Oracle community.

A resident of Dublin, Vilin has three daughters: ages 5, 9, and 14. When he has a few spare moments (which must be few and far between), he'll make a move on his computer chessboard. He also plays indoor soccer in Livermore in addition to the pick-up games that form during lunch.

Vilin has an affinity for world history. He has traveled extensively throughout Europe and Asia and loves the classical cities of Salzburg, Rome, Paris, Florence and Athens. Vilin enjoys spending time in museums and looks forward to visiting Egypt someday. He has also seen a fair amount of the U.S. When he came from Iran in his late teens, he landed in Washington, DC. He's also resided in Okalahoma and Tennessee and Georgia—he got his MBA at Georgia Tech.

He's also worked for Sun Microsystems in the Solaris kernel group and worked at Oracle for four years before taking a position at Cingular as a DBA manager. In addition to his MBA, Vilin has a civil engineering degree and has had some interesting jobs. Be sure to ask him about his work developing simulators for crane operators!

So, when you see him at the meeting running things without a hitch, be sure to introduce yourself. And if you need some travel advice in planning your own summer vacation, it sounds like Vilin is the right person to ask! ▲

Don't Miss Our Next Meeting!

The NoCOUG Board has planned a great conference that you won't want to miss. It's taking place on Thursday, August 23 in San Ramon. Sponsored by Chevron, the one-day conference will have three parallel tracks:

- DBA
- Data Warehouse
- Development

Registration begins at 8:00 a.m. For more details, see our website at www.nocoug.org. ▲

Don't Miss Our Summer Conference!

Gaja Vaidyanatha, a top-notch presenter, will be giving a two-hour session on optimizing and fine-tuning your Oracle system's overall performance.

Learn to implement tuning solutions using the proven methodology found inside his book *Oracle Performance Tuning 101*. Gaja will share step-by-step practical advice for removing bottlenecks, minimizing downtime, and increasing overall system performance. You'll also get tips for identifying Oracle's current bottlenecks, prioritizing and performing tuning operations, writing optimal SQL, interpreting statistics, and much more.

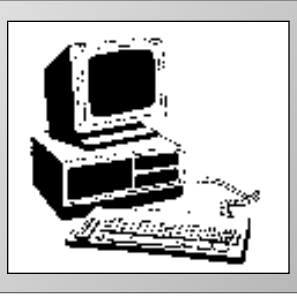
Gaja will also provide a valuable overview to help you:

- Understand the key components of a tuning methodology
- Set performance objectives, establish goals, and cease efforts when goals are met
- Learn how to generate well-written SQL commands — and know how to recognize poorly written SQL

Gaja Vaidyanatha

- Tune and manage various components of the Oracle Instance
- Use meaningful Oracle initialization parameter settings
- Implement proactive management of space in the Oracle database
- Understand I/O tuning with extensive coverage on implementing RAID on Oracle
- Get details on operating system tuning for Solaris, HP-UX, AIX, and Windows NT

For complete details about the conference on August 23, please see the back cover. ▲



Do you have a favorite Oracle reference? Let us know at journal@nocoug.org

TECH TIPS

Locally Managed Tablespaces in Oracle8i

Oracle8i introduced a new type of tablespace called Locally Managed Tablespaces. Though Oracle and many articles on Database Administration recommend the usage of Locally Managed Tablespaces, a DBA has to be aware of some considerations and caveats in the usage of Locally Managed Tablespaces. Download this informative white paper by **Raghav Vinjamuri** of Fujitsu Siemens Computers from the NoCOUG website at: http://www.nocoug.org/presentations.htm.

Behind the Scenes

(continued from page 1)

ing starts we "get down to business." During a typical three-hour meeting, we're planning for future quarterly conferences, scheduling speakers and topics, reviewing our budget, listening to proposals for changes by the board, and voting on issues in order to best represent the NoCOUG members.

We have a nice meal at a restaurant at our evening meetings. We have our quarterly bagel eating contest at our Saturday morning meetings. (I won't say who wins!)

We also frequently invite members of NoCOUG who are interested in volunteering to attend our meetings. If you are interested in learning more about NoCOUG and what happens behind the scenes, talk to a board member. If you're an Oracle user, you already have a lot to offer at our board meetings. Come share your thoughts about what the organization needs, what we can improve upon, and most importantly, what you'd like to help us with! Chat with us during a quarterly meeting or write to us at **board@nocoug.org**. We look forward to hearing from you. ▲

A Trip Down Memory Lane

with a bit of NoCOUG nostalgia

by Joel Rosingana

I'm sitting at my laptop staring at a blank MSWord document. I think I understand the term "writer's block". When our Journal Editor asked me to do an article, I said, "Sure, why not?" Then later, writer's fear sets in! What should I talk about? I could chat about volunteerism. It's been done before – many times. I could discuss how NoCOUG performs a valuable service to the Oracle community. We've done that before. So, I think I'll just leave the subject with . . . I've been involved with NoCOUG for about a decade and it's been great fun and very satisfying. When you see me, ask how YOU can have this much fun. The fact that I've been around for a while, leads me to a topic for this article – *nostalgia.*

I was rummaging through my office archives (stacks of old stuff) the other day.

Up pops a NoCOUG newsletter dated October 1990. It is Volume 3, Number 4. If my arithmetic is correct, this could be the 12th newsletter published. I sat down and read it cover to cover. I believe the contents would be a trip down memory lane for the "old timers" and a bit of NoCOUG history for the new folks. Let me cover just a few items of interest from the newsletter.

The light beige 18-page newsletter starts with a message from the president, John DeVoy. John was a long-term president. He returned for another term in the late 90s. The message is very similar to today's content – the importance of membership and volunteerism. The technology changes. But, the administrative needs remain the same.

The list of the 1991 NoCOUG Board: President – John DeVoy, VP – Dan Lamb, Membership – Merrilee Nohr, Treasurer – Ann Seki, Secretary – Gary Falsken, Publicity Director – open.

Some of the article contributions follow.

"An Object-Oriented Approach to Data Storage" by Robert Burnham of Design Generation. This would be considered a "before its time" presentation.

Joel Rosingana

"Affinity Group Meets with Oracle Top Management in Anaheim" by Merrilee Nohr of Design Generation. This article describes a newly chartered group to address unresolved problems from Oracle users. The task is defined as "a pro-active endeavor between Oracle's executive staff, USA user group representatives and various IOUG officers."

"International Oracle Users Week September 1990" by Robert Burnham. The IOUW was held at the Anaheim Hilton Hotel and Convention Center. The article described the theme as "the renewed Oracle commitment to quality and service."

To end the nostalgia trip

Here is the Agenda of the November 28, 1990 NoCOUG Regional Meeting. The meeting was held at Xerox Engineering Systems in Santa Clara. The format was single track starting at 9:15 am and ending at 2:15 pm. There were five presentations running from 45 minutes to an hour each. The hour and a half lunch included SIG meetings – SQL*Forms, DBA and CASE tools. The presentations covered Oracle telephone support services, version 2 SQL*Forms applications to version 3 conversion, customized Menus for SQL*Forms applications, providing context-sensitive help in SQL*Forms, and Transparent distributed processing.

This historical trip and the User Group's continued involvement in the Oracle environment is best represented by the following statement of purpose found on the inside cover of that 1990 newsletter: "NoCOUG was formed to provide a forum for sharing information within the Oracle user community." This commitment is as strong today as it was in 1990.

I hope to see you all at the Chevron Summer Conference. ▲

Running an OS Command from Oracle

By Chris Lawson

Chris Lawson

It is occasionally desirable to issue an operating system command directly from the database. For instance, after a database restart, a webserver or other application might need to be re-initialized in order to reconnect to the database. Rather than have an administrator manually reset the application each time, it would be more convenient if the database could run a UNIX script that issues the application reset automatically, once the database is open. This automatic restart would be especially important if cold backups are regularly performed on the database.

Oracle 8i Method

Prior to Oracle 8i, it was necessary to create an "external procedure" in order to issue an OS command from the database. With Oracle 8.1.6, however, the OS command is somewhat easier to activate—using database features that are already provided as part of a normal 8i install. There are no special external procedures to build.

The new OS-command method makes use of the Oracle 8i Java Virtual Machine, or JVM, which is included in Oracle 8.1.6 and later. Oracle Corporation supplies a sample Java procedure that is used to actually issue the OS command. The DBA need only prepare the JVM, then build and test the Java procedure. Optionally, the DBA can use a new type of database trigger, so that upon database startup, the new Java stored procedure is called, which in turn issues the OS command.

The steps to create the OS-command process are:

- Initialize the JVM (Java Virtual Machine)
- Build a simple Java Stored Procedure
- Test execution of the stored procedure
- (Optional) Build trigger to execute the stored procedure upon database startup

Setup Java Virtual Machine

Although the Java Virtual Machine is included in the Oracle typical installation, the Oracle documentation is not very clear on precisely how to prepare it for use. In fact, several MetaLink queries and a few experiments were necessary to clarify that the user that will run the new java procedure must be granted JAVASYSPRIV. The DBA role is not needed.

There are just a few other steps in preparing the JVM. The JVM initialization script must be run to initialize the JVM. This is accomplished by running, as SYS, initjvm, found in $ORACLE_HOME/javavm/install.

In addition to the initialization script, it is also important to ensure that the Java Shared Pool is set > 50 MB. The final setup step is to ensure that CLASSPATH environmental variable is set. Typical value:

```
/u01/app/oracle/product/8.1.6/jlib:/u01/app/oracle/product/
8.1.6/product/jlib
```

Create Java Procedure

Once the JVM is prepared, the DBA should create the Java stored procedure. The source of this procedure is provided by Oracle, and is the key to the OS-command method. This procedure, ExecuteCmd.java, is shown in Figure 1.

The DBA should first compile and load the Java class:

- Compile source into class file: javac ExecuteCmd.java
- Load java class into database:
  loadjava -u user/password ExecuteCmd.class

Then, the DBA should store the procedure in the database:

```
Create or replace procedure EXECUTECMD (S1 VARCHAR2)
AS LANGUAGE JAVA name 'ExecuteCmd.main(java.lang.String[])';
```

Testing the Procedure

Once the procedure is in the database, it is fairly simple to test. First, build a simple script, 'test,' that writes directory contents to some file. There is only one line in this script:

```
ls > /home/oracle/scripts/test.log
```

Now, we are ready to try executing the stored procedure. In SQL*Plus, simply execute the stored procedure executecmd, while providing two arguments—the shell binary and the path to the test script. Note that complete paths must be provided for both arguments. For example:

```
Exec executecmd('/usr/bin/sh /home/oracle/scripts/test');
```

Successful execution of this simple test script will cause a new logfile, test.log to be created.

Should there be problems running the procedure, it is helpful to set debugging options. Typically, problems experienced will be related to not providing the complete path for each argument.

To use the debugging mode, start SQL*Plus then run:

```
set serveroutput on
call dbms_java.set_output(2000);
```

Now, re-run the procedure, as described above. The various debug lines in the ExecuteCmd Java procedure will now be active. When the procedure is called with this debugging option on, the response should be:

```
In main
Return code from process 0     [0 represents no errors]
Done executing
PL/SQL procedure successfully completed.
```

Figure 1.

Java Stored Procedure: ExecuteCmd

```
/*
* This is a sample application that uses the Runtime Object
* to execute a program.
*
*/

/* Import the classes needed for Runtime, Process, and Exceptions */
import java.lang.Runtime;
import java.lang.Process;
import java.io.IOException;
import java.lang.InterruptedException;

class ExecuteCmd {

     public static void main(String args[]) {

          System.out.println("In main");

          try {
               /*   Execute the command using the Runtime object
                    and get the Process which controls this command */

               Process p = Runtime.getRuntime().exec(args[0]);

               /*   Use the following code to wait for the process
                    to finish and check the return code from the
                    process */
               try {

                    p.waitFor();

               /*   Handle exceptions for waitFor() */

               }    catch (InterruptedException intexc) {

                    System.out.println("Interrupted Exception on waitFor: " +
intexc.getMessage());
               }

         System.out.println("Return code from process"+ p.exitValue());

         System.out.println("Done executing");

      /*  Handle the exceptions for exec() */

      }   catch (IOException e) {
               System.out.println("IO Exception from exec : " +
                                  e.getMessage());
               e.printStackTrace();
      }
     }
}
```

Create Trigger

Now that an OS command can be issued via a stored procedure, it is possible to run the OS command immediately after database startup. This is accomplished via a new form of database trigger that runs after startup of the database. The trigger executes our Java stored procedure, executecmd, that in turn runs any desired Unix script. In the example below, a script called vreset is executed after database startup.

```
Create Or Replace Trigger Restart After Startup On
Database
Begin
executecmd('/usr/bin/sh [path]/vreset');
end;
/
```

Note: It is critical that the 'sh' command uses the full path.

Summary

Oracle 8i provides a simple method to issue an OS command from the database, without the use of any special external routines, as required in older Oracle versions. Using the new Java Virtual Machine, contained within Oracle 8i, it is possible to run any UNIX script via a stored Java procedure. New types of database triggers further allow the DBA to automate OS command execution immediately after database startup. Together, these features can greatly simplify the task of resetting certain applications that need to be reset after database cycling.

New types of database triggers further allow the DBA to automate OS command execution immediately after database startup.

References

The software versions assumed for purposes of this paper are: Oracle Enterprise Edition 8.1.6.2, Operating System Solaris 2.6.

Oracle Corporation, Note 109095.1, How to do a System Call from a JAVA Stored Procedure.

About the Author

Chris Lawson consults in the San Francisco Bay Area, where he specializes in performance tuning of financial applications. He is a regular presenter for the Northern California Oracle Users Group (NoCOUG). His previous papers, including the "Ten Database Mysteries" series and "Oracle DBA: Physician or Magician?" may be found at http://www.dbspecialists.com. Chris may be reached at Chris_Lawson@yahoo.com. ▲

SPONSORSHIP APPRECIATION

Many Thanks to Our Sponsors

NoCOUG would like to acknowledge and thank our generous sponsors for their contributions. Without this sponsorship, it would not be possible to present regular events while offering low cost membership dues. If your company is able to offer sponsorship at any level, please contact NoCOUG President Vilin Roufchaie at 925-487-9588 or write to board@nocoug.org.

Long-term full event sponsorship:

LOCKHEED MARTIN

CHEVRON

Long-term supplemental event sponsorship:

ORACLE

Thank you! Year 2001 Gold Level Support Vendors:

- BMC Software
- Cast Software, Inc.
- Database Specialists, Inc.
- Embarcadero Technologies
- Informatica
- Lecco Technology, Inc.
- Network Appliance
- Precise Software Solutions
- Quest Software, Inc.
- Veritas Software Corporation
- XIOtech Corporation

For information about our Gold Level Vendor Program, Contact Nora Rosingana at 925/820-1589 or via email at noraros@pacbell.net

TREASURER'S REPORT

Judy Lyman, *Treasurer*

Beginning Balance		
April 1, 2001		**$ 46,361.51**
Revenue		
Advertising	600.00	
Sponsorships	-	
Membership Dues	11,200.00	
Meeting Fees	480.00	
Vendor Receipts	6,000.00	
Interest	186.93	
Total Revenue		**$ 18,466.93**
Expenses		
Regional Meeting	2,265.68	
Journal	5,491.28	
Membership	2,709.36	
Administration	952.18	
Income Taxes	-	
Board Meeting	452.96	
Web Site	165.00	
Web Site Design	350.00	
Miscellaneous	500.00	
Total Expenses		**$ 12,886.46**
Ending Balance		
June 30, 2001		**$ 51,941.98**

Oracle Mythology: The Cache Buffer Hit Ratio

By Kyle Hailey

Introduction

There are a number of myths and folk tales about Oracle. Many of the myths surrounding the area of performance tuning originate from methods that were once valid in Oracle 6, but have since been replaced with more efficient methods and tools. One example of just such a myth is that of determining the cache buffer hit ratio (CBH ratio). The CBH is the ratio of how much I/O Oracle reads off disk (bad, slow) versus how much data it is able to read from its cache buffer (good, fast).

The CBH formula has a number of variations but is usually defined as:

```
1 - ((physical reads) / (db
block gets + consistent reads))
```

where the values for db block gets, consistent gets, and physical reads come from v$sysstat.

The CBH ratio is often the first statistical value a DBA looks at to find out if a database is tuned correctly.

Once considered an important tuning statistic, the CBH ratio allowed a database administrator to determine how much time the database spent waiting for data to be returned from disk (I/O), as opposed to how much was being found in the faster (and thus more efficient) buffer cache.

Beginning with Oracle 7, this same information can be found in the wait event table V$SYSTEM_EVENT.

The V$SYSTEM_EVENT table is more and more widely used for tuning but even after 10 years of having access to the wait events, DBAs still ignore these indispensable tuning statistics in situations where wait events would rapidly point out the problem.

Where the CBH ratio of Oracle 6 only allowed administrators to make educated guesses about how much latency users incurred when waiting for data to be returned, the wait events in the V$SYSTEM_WAIT table provide accurate information on how much time users are spending waiting for data to be returned.

Analyzing using CBH

The CBH is used to show when there is an I/O problem where too many blocks are read off disk (which is slow) versus how many are found in the Oracle buffer cache. But the main problem with this line of reasoning is, "What is too much I/O off disk?"

This question of what is too many disk I/Os depends on how much that disk I/O is actually slowing down the users. This can only be guessed from the CBH .

Thus CBH ratios that are acceptable tend to vary from one DBA to another, but generally a good CBH ratio is considered to be around the 94-96% range. Below that, efficiency is compromised.

Here are some examples where the CBH is misleading:

1) Alarmingly low CBH but, in fact, response time is good:

Good response time and a CBH ratio below 90% can happen on a machine where the Oracle buffer cache is too small and the machine uses UNIX file buffer cache (using data files on a cooked file system) or intelligent disk cabinets with read back cache. The data can be cached in the UNIX buffer cache or the read back cache, and not in the Oracle buffer cache because it is too small. Oracle will report high physical reads because it is asking the machine for data, which it assumes, is coming from disk. Due to these I/O requests by Oracle, the cache hit ratio is low; however, the service times will be extremely rapid because the data is cached in memory and not from disk (if, of course, it is already cached in the UBC or read back cache). This is only an example. I'm not suggesting, depending on the UNIX, file buffer cache or read back cache.

2) Acceptably high CBH but, in fact, response time is bad:

Take a fairly large SGA, cache a

large table in memory (data warehouse kind of thing). Do an update on a column of the large table joining with some smaller table that is say 20 times smaller but isn't cached. The cache hit ratio could be around 95%. For example, say by the end of the operation you need to read all of the big table and all of the little table (or index) so that 20 out of every 21 blocks are in memory. Now suppose accessing that 1 in 20 blocks is 1/10th a second because of a disk problem. If the big table has 360000 blocks and the little table (or index) has 1/20th, 18000 blocks. So then at 1/10th a second access for 18000 blocks adds up to half an hour of wait time with a buffer cache hit ratio of 95%. Your bad disk problem won't show up with the CBH ratio but shows up very distinctly in the wait events.

Analyzing using V$SYSTEM_WAIT

The V$SYSTEM_EVENT table lists the resources and events that users must wait for when they are working. Using the V$SYSTEM_EVENT table, you can easily determine if and where an I/O problem is causing users latency in retrieving data. A global I/O problem can be seen in V$SYSTEM_EVENT by determining how much time the system waits for I/O. V$SYSTEM_EVENT reports I/O which reads data into the buffer cache with the following two types of wait events:

```
Db file sequential read – single block reads
Db file scattered read – multi block reads
```

Example 1: Where the CBH would point out a problem the wait events show that response time is fast:

We would see that our total time waited for I/O was low – probably not a bottleneck.

Event Name	Count	Total Time	Avg Time
db file sequential read	332	3.32	0.01
db file scattered read	48300	483.30	0.01

The time values are given in 1/100th a second. In the above case our reads are taking .01 hundredths of a second, or 1/10,000 a second. This speed is more rapid than a disk could currently sustain, but happens when reading from memory file caches. In this situation, despite the fact that we waited for 48 thousand disk reads, our total time waited for the resources is only 4 seconds, which probably isn't much. This 4 seconds is the total for all the users over the period the utlbstat/estat snapshot was run.

Example 2: Where CBH shows good buffer cache usage but the wait events show a bottleneck:

We would see that the I/O waits were being serviced slowly and that there was a problem.

Event Name	Count	Total Time	Avg Time
db file sequential read	332	3320	10.00
db file scattered read	48120	481200	10.00

In this case we have a disk or disk controller problem and the disk reads are taking 1/10th a second. Now our 48K disk reads result in almost an hour and half of wait time.

In Conclusion

The cache buffer hit ratio is a concept of little meaning in Oracle 7 and Oracle 8. Even if the CBH rate is at 100%, it can be falsely reassuring[1]. Why try to figure something out in the dark when you can turn the lights on? Why use cache buffer hit ratio which can be a bit ambiguous to determine inefficiency when wait events will give a clear detailed picture? What is important is how long we waited to get the data not how many times we had to wait. The average wait time for a block can be 1/10,000ths a second to 1/10th a second. The same buffer cache ratio in both situations had two very different performance impacts. The cache buffer hit ratio is a carry over from Oracle 6 where there were no wait events and the only way get an idea how much we waited for disk reads was the buffer cache hit ratio. But the information that the CBH attempts to estimate has been provided in detail in Oracle 7 in the table V$SYSTEM_EVENT.

About the Author

Kyle Hailey worked for Oracle Corporation for 10 years between 1990 and 2000. He started in UNIX support, ported Oracle onto Digital UNIX machines, worked in Oracle France on performance and support problems for some of the largest European customers and benchmarks, and finally worked in Oracle's kernel development group on performance issues. His personal tuning web site including documentation and tools can be found at http://oraperf.sourceforge.net.

(See Appendix starting on page 15)

[1] *From the ever-impressive Jonathan Lewis (http://www.jlcomp.demon.co.uk/myths.html) is a wonderful example of a small SQL statement that by reducing the fetch array size, the number of consistent gets increases. Thus the CBH goes down while the response time of the SQL statement goes up. He also gives an example of a system where the CBH is around 100% but it is caused by badly written SQL. The SQL statements need tuning but the system statistics say things are running well based on the 100% CBH.*

Appendix: Using the V$SYSTEM_EVENT Wait Events Table

To use V$SYSTEM_EVENT table to determine wait times for resources and events, the following init.ora parameter needs to be set:

```
Time_statistics=true
```

(another myth is that timed_statistics slows the database down. The only cases where timed_statistics=true has had a noticible impact has been in the cases of a couple of rare bugs in Oracle)

Once TIMED_STATISTICS is set to TRUE, then the wait data can be gathered by:

1) querying V$SYSTEM_EVENT directly

2) running ORACLE_HOME/ admin/utlbstat.sql and utlestat.sql

3) running snap pack from 8.1.6 onwards (see ORACLE_HOME/ rdbms/admin/sp*.sql)

4) using your tools, like oraperf. sourceforge.net/seminar/collect.html

Once the wait event data has been collected, the waits can be sorted by time, with the longest waits first. Among the list of waits from V$SYSTEM_EVENT are idle waits, background process waits and user process waits. The user process waits are the first waits to look at to find and eliminate bottlenecks.

A typical waits event section of a report.txt from utlestat.sql looks like:

```
SQLDBA> Rem System wide wait events.
SQLDBA> select n1.event "Event Name",
     2>         n1.event_count "Count",
     3>         n1.time_waited "Total Time",
     4>        (n1.time_waited/n1. event_count) "Average Time"
     5>         from stats$event n1
     6>         where n1.event_count > 0
     7>         order by n1.time_waited desc
```

Event Name	Count	Total Time	Average Time
client message	1770692	26676762	15.065726846
rdbms ipc message	7626	782385	102.59441385
pmon timer	535	158481	296.22616822
smon timer	7	146587	20941
db file sequential read	83662	79166	.94625995075
db file scattered read	5224	26120	5
db file parallel write	6982	6848	.98080779146
inactive session	60	6060	101
latch free	15653	4442	.28377946719
enqueue	28	2268	81
log file sync	3638	2245	.61709730621
log file parallel write	3892	2004	.51490236382
rdbms ipc reply	19	1813	95.421052632
control file parallel write	843	850	1.0083036773
log file space/switch	5	236	47.2
db file single write	726	210	.28925619835
control file sequential rea	1196	22	.01839464883
library cache pin	24	10	.41666666667
buffer busy waits	28	6	.21428571429
library cache load lock	2	5	2.5
write complete waits	5	5	1
log file sequential read	1	2	2
log file single write	2	0	0

23 rows selected.
SQLDBA>

We can immediately simplify this list by eliminating the wait events that are not related to users:

Event Name	Count	Total Time	Average Time
client message	*1770692*	*26676762*	*15.065726846*
rdbms ipc message	*7626*	*782385*	*102.59441385*
pmon timer	*535*	*158481*	*296.22616822*
smon timer	*7*	*146587*	*20941*
db file sequential read	83662	79166	.94625995075
db file scattered read	5224	26120	5
db file parallel write	**6982**	**6848**	**.98080779146**
inactive session	**60**	**6060**	**101**
latch free	15653	4442	.28377946719
enqueue	28	2268	81
log file sync	3638	2245	.61709730621
log file parallel write	**3892**	**2004**	**.51490236382**
rdbms ipc reply	19	1813	95.421052632
control file parallel write	**843**	**850**	**1.0083036773**
log file space/switch	5	236	47.2
db file single write	**726**	**210**	**.28925619835**
control file sequential rea	**1196**	**22**	**.01839464883**
library cache pin	24	10	.41666666667
buffer busy waits	28	6	.21428571429
library cache load lock	2	5	2.5
write complete waits	5	5	1
log file sequential read	**1**	**2**	**2**
log file single write	**2**	**0**	**0**

The above output gives a profile of the "bottlenecks" in the database. We can eliminate most of the lines because they are usually irrelevant. For example:

- *italics* lines are idle events. The processes are idle and waiting for work to do.
- **Bold** lines are background waits, and not user shadow process waits. Thus, at least in the first analysis, we can ignore them. We are interested in why users have to wait and not why the background processes (pmon,smon,lgwr,dbwr) have to wait.

(continued on next page)

This leaves us with:

```
Event Name               Count   Total Time  Average Time
---------------------- ------  ---------- -----------
db file sequential read   83662       79166  .94625995075
db file scattered read     5224       26120             5
latch free                15653        4442  .28377946719
enqueue                      28        2268            81
log file sync              3638        2245  .61709730621
rdbms ipc reply              19        1813  95.421052632
log file space/switch         5         236          47.2
library cache pin            24          10  .41666666667
buffer busy waits            28           6  .21428571429
library cache load lock       2           5           2.5
write complete waits          5           5             1
```

The TIME_WAITED is measured in 100ths of seconds. Some of the basic easily tunable events are:

buffer_busy_waits - rbs problems, or need free list, or maybe just I/O throughput

free_buffer_waits - dbwr not working hard enough or buffer cache too small

write_complete_waits - dbwr may be working too hard

db_file_scattered_read - read (full table scan) disk wait, buffer cache too small

db_file_sequential_read - rowid (index) block read, read disk wait, buffer cache too small

enqueue - lock waits

latch_free - waits for latch, see v$latch to see exactly which latch

log_buffer_space - log buffer needs to be larger

log_file_space_switch - log buffer space wait or log file switch wait (pre-Oracle 7.3)

log_file_switch_checkpoint_incomplete - redo logs too small or not enough redo logs

log_file_switch_completion - wait for log file while switching log file

log_file_sync - write to log file wait, raw devices are faster for log files

From the report.txt above the biggest waits are on

```
db file scattered read
db file sequential read
```

Thus, the most important waits are:

```
Event Name               Count   Total Time  Average Time
---------------------- ------  ---------- -----------
db file sequential read   83662       79166  .94625995075
db file scattered read     5224       26120             5
```

The sum of the total time waited for I/O reads into the buffer cache is:

```
= 79166 + 26120
= 105286 hundredths of second
= 1052 seconds
= 17.5 minutes
```

Now the question, so is 17.5 minutes significant? To determine if 17.5 minutes is important we can compare it to:

1) length of time over which the statistics were taken

2) the number of users connected during the time the statistics were taken

RESOURCE CORNER

At **DBAsupport.com** (http://www.dbasupport.com), you'll find extensive tips, including many already dedicated to Oracle 9i. They also provide an email newsletter. Visit their message boards to discuss the latest burning Oracle topics.

Send your favorite resources to journal@nocoug.org for publication. ▲

3) the amount of CPU used by users during the same period

The amount of CPU used can be found by looking at the table V$SYSSTAT and finding the value for the statistics:

```
CPU used by this session
```

in the statistic section of the bstat/estat report. Compare CPU time to the WAIT time, for example:

```
CPU > 2 * WAIT : probably ok
CPU ~= WAIT : probably some tuning to do
CPU < WAIT : problem area
```

Another way to look at the relative importance of the wait events is to look at how many users were connected during the bstat/estat to figure out the average time waited by users. First one needs to know over what period of time the bstat/estat was run. This is given at the bottom of the report.txt.

Second, one needs to know how many users were working (thus the maximum number of possible users waiting for these events). The scripts utlbstat/estat only report the number of users correctly from 8.1.6 onwards. On versions before 8.1.6 the following for the number of users connected:

```
select value from v$sysstat s, v$statname n
where s.statistic#=n.statistic# and
n.name='logons current';
```

With the information on the period of time over which the statistics were taken and the number of users connected one can decide whether the wait time is important. For example:

```
stat time 18 minutes with one user ==>
probably needs tuning
stat time 18 hours with 1000 users ==>
probably not a big deal
```

More information is available about individual users in V$SESSION_EVENT and V$SESSION_WAIT. V$SESSION_WAIT gives not only the times waited but also detailed information on the events themselves. For example, this table contains the actual file and block address for each read into the buffer cache. ▲

Comparing Join Techniques

(The following is an excerpt from a white paper by Tim Quinlan, TLQ Consulting Inc. entitled, "Advanced SQL Tips for Developers and DBAs.")

Nested Loop Joins

The Nested Loop join is one of the two most common join techniques - the other being the Sort Merge join. This is also a relatively simple join to understand. Take the example of two tables joined as follows:

```
Select *
From Table1 T1, Table2 T2
Where T1.Table1_Id = T2.Table1_id;
```

In the case of the Nested Loop Join, the rows will be accessed with an outer table being chosen (say Table1 in this case) and for each row in the outer table, the inner table (Table2) will be accessed with an index to retrieve the matching rows. Once all matching rows from Table2 are found, then the next row on Table1 is retrieved and the matching to Table2 is performed again.

It's important that efficient index access is used on the inner table (Table2 in this example) or that the inner table be a very small table. This is critical to prevent table scans from being performed on the inner table for each row of the outer table that was retrieved. Nested Loop joins are usually used in cases where relatively few rows are being accessed in the outer table and joined to the inner table using efficient index access. It is the most common join performed by transactional (OLTP) systems.

Cluster Joins are a special case of Nested Loop join where the outer table can access the inner table with the cluster index. Clusters can be very efficient, but have many maintenance and performance drawbacks. These are rarely used.

Sort-Merge Joins

The Sort-Merge join – also called a Merge Scan – is a very popular join performed when accessing a large number of rows. It is also seen in background tasks, batch processing and decision support systems. Take an example of two tables being joined and returning a large number of rows (say, thousands) as follows:

```
Select * From Table1 T1, Table2 T2
Where T1.Table1_Id = T2.Table1_id;
```

The Merge Scan join will be chosen because the database has detected that a large number of rows need to be processed and it may also notice that index access to the rows are not efficient since the data is not clustered (ordered) efficiently for this join.

The steps followed to perform this type of join are as follows:

1) Pick an inner and outer table

2) Access the inner table, choose the rows that match the predicates in the Where clause of the SQL statement.

3) Sort the rows retrieved from the inner table by the joining columns and store these as a Temporary table. This step may not be performed if data is ordered by the keys and efficient index access can be performed.

(continued on page 25)

Data Warehouse Cost Components

By Sekhar Palli

Data warehouse:
"A copy of transaction data specifically structured for query and analysis"
—Ralph Kimball

Cost components in estimating the ROI (Return on Investment) of a data warehouse:

Many companies incorrectly value their data warehouse (DW). The truth is, the value of the warehouse stems from the new and changed business processes it enables. Thus, in developing a warehouse, it is extremely important to envision who is going to use the warehouse, how they will use it and why using it will improve the current process. It is not unusual for well-considered and designed warehouses to generate first-year ROIs of 100 to 400 percent. The value of the new business processes can be determined by comparing the costs with the expected benefits.

Costs

Costs include items like:

- Hardware, software, development personnel and labor costs
- Operational costs like ongoing systems maintenance

Benefits

Benefits typically fall into two categories, new revenue and reduced costs.

Added revenue

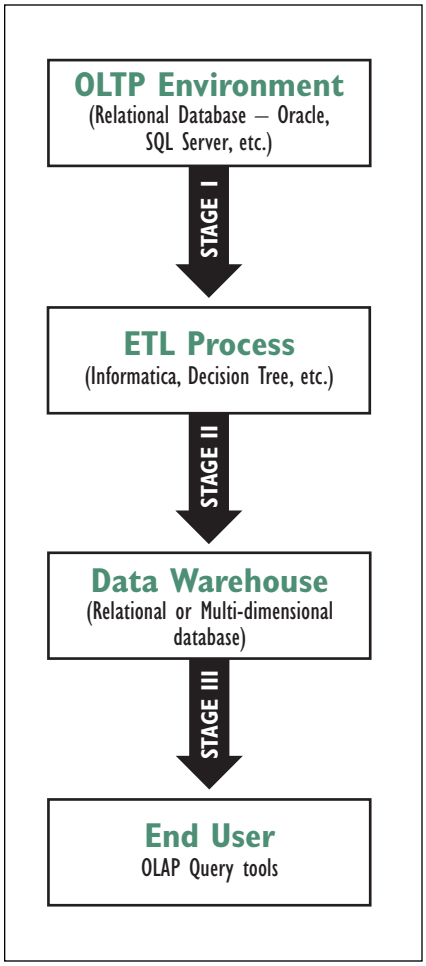
- Will the new (business objective) process generate new customers? What is the estimated value?
- Will the new (business objective) process increase the buying propensity of existing customers? By how much?
- Is the new process necessary to ensure that the competition doesn't offer a demanded service that you can't match?

Reduced costs

- What costs of current systems will be eliminated?
- Is the new process intended to make an operation more efficient? If so, how and what is the dollar value?

Data warehouse components

Let's take a look at the various components involved in data warehousing, and try to estimate their costs. It is very hard to estimate cost without an actual project specification. Costs can be classified into one of three categories: software, hardware and labor.



Software elements of a data warehouse.

Software

Costs include the software applications required by the project such as ETL tools, desktop query tools and the relational database management system (RDBMS) license fees. Software costs can vary depending upon the architecture you implement and the tools you choose in the process. Let's look at the typical data warehousing process.

First let's cost the databases.

Relational databases can cost anywhere from $500 to $500,000 depending on the hardware you run it on and the number of users using it. If you choose an Oracle database and the application is going to serve more people, typically it is easier to go with server based licensing, which is calculated on the processor speed used on the box. If you are using multi-dimensional databases such as Hyperion, the minimum cost might be $100,000 and increase on a per-user basis.

ETL tools:

In stage I, data is extracted from an OLTP (online transaction processing) environment, transformed and loaded into a database. Depending upon the data's complexity, it might be staged and processed at several levels. Informatica's (Power Mart/ Power Center) and Cognus' (Decision Tree) ETL tools are the more prominent tools in the industry. These tools come with lots of bells and whistles, which are at your disposal. The typical cost of the Informatica tool can vary from $75,000 to $200,000 depending upon the processor speed and number of CPUs used in the server. Cognus' ETL tool, Decision Tree, which does not have as many extras as Power Mart/Power Center, will satisfy 90% of typical DW projects. It costs around $50,000 to $100,000.

OLAP query tools:

These tools are used by the end user to do adhoc querying. These tools make IS professionals' and analysts' jobs easier. Business Objects, Cognus and Brio have similar strengths in the market place and are priced competitively. They are priced on a per user and per server basis, again based on the processor speed and number of CPUs used. Typical desktop tools cost $1000 per user and $25,000 to $50,000 per server. The key here is to distinguish your user population between power users and casual users and license them accordingly.

Hardware

Costs include the devices required for the project such as server system(s), client systems, and network communication devices.

Typical hardware cost:

One Sun Enterprise 4500 server with 4 processors to host the database and OLAP application costs approximately $400,000. This can be more or less expensive, depending on how much data is processed and how many people are accessing the system.

Labor

Costs pertain to the use of internal and external resources dedicated to the project. Individuals working on the data warehousing project fill the following roles: project manager, business analyst(s), DW application specialist(s), database administrator, systems administrator, and trainer(s). Labor can be the biggest cost, and varies based on the experience of the people as well as whether you fill the position internally (employees) or externally (consultants). The labor cost is hard to gauge until the scope of the project is specified. It is always better to start with a narrow definition (divisional or subject scope), keeping the corporate vision in the process.

Conclusion

Estimating the cost and ROI of a data warehousing project is a very complex process. In order to assess the actual cost of the project, first determine its scope and the size of the data to be processed and stored, followed by defining the architecture. Finally, you should allocate the cost to the various components. Several companies, such as Wal-Mart, have taken advantage of warehouses to significantly increase their revenue, but only after doing a detailed analysis of all aspects of the process.

Note: *Data provided is for informational purposes only. Detailed analysis is required when estimating data warehousing projects. The prices shown in this article are list prices based upon conversation with the sales representatives from the respective companies. Prices can vary based on the relationship you have your vendors in addition to the architecture you choose. Data warehouse projects are one of the most expensive in nature in the IT projects.*

About The Author

Sekhar Palli is currently working as a Senior Applications Manager at C-Cube Microsystems. He can be reached at spalli@c-cube.com for any comments.

Special thanks to my VP/CIO Judy Armstrong for her continued support, and to my friend Steven Horwitz for helping me out in writing this article. ▲

Elevate Your Game

Oracle8i Tuning Using Statspack

By Shankaran Iyer, Renaissance Worldwide, Inc.

Introduction

Tuning is an iterative process and is largely based on how effectively the DBA is able to collect statistics as well as analyze them all together. One of the critical items for DBAs is knowing their environment. From this perspective it is critical for a DBA to not only understand the type of application and access methods but also the ability to effectively analyze the database statistics at different times of the day and take proactive corrective action. The knowledge about the environment in terms of utilization of memory, I/O, waits and the like can help the DBA to proactively tune their environment. Statspack is the new set of Oracle provided package and procedures which provides a diagnostic tool for the administrator to collect statistics about database usage. Unlike utlbstat/utlestat, Statspack provides a more robust and complete set of tuning statistics. There are a plethora of tools available to collect statistics for different performance layers like operating system, database, network, application, etc. Every tool has its pros and cons and it really becomes important to use a few of them together to get more information about the environment. The primary objective of these diagnostic tools is to help us understand our environment. Statspack is Oracle's newest set of diagnostic tools for proactive and reactive tuning. This paper contains more details on how to implement and analyze your database using Statspack.

What is Statspack?

Statspack is a set of scripts provided by Oracle which can be used for collecting, storing and viewing database related statistics. Statspack scripts are mainly written using SQL and PL/SQL. Statspack provides the ability to collect database statistics over a period of time and can help in conducting impact analysis and establish whether database or application upgrades have improved the performance or not.

Statspack is intended to be used starting with Oracle version 8.1.6 onwards. However, an unsupported 8.0.x and 8.1.5 compatible version can be downloaded from Oracle's website. The older version works just as well as the new one and can be installed in a production environment with minimal risk.

Statspack versus BSTAT/ESTAT

Statspack can be viewed as a successor to utlbstat/utlestat. The utlestat/utlbstat tool has, so far (till Oracle version 8.0), been the only diagnostic tool which gives out database statistics in the form of a report for a single window of time. The utlbstat/utlestat report provides a reasonably good overall knowledge about the database. However, it has become archaic in the sense that it does not provide good information for newer features which are now available with Oracle – for example large buffers, locally managed tablespaces, etc. Also, utlbstat/utlestat provides information only for a single window of time. This could be insufficient particularly if we want to track the performance hourly or do a trend analysis. Utlbstat/utlestat reports are hard to interpret and hard to read. Comparing reports for two different periods is often difficult. Furthermore, these reports have a lot of statistics and numbers that are undocumented and may not always help with tuning. Statspack provides a better alternative for Oracle instances as compared to utlbstat and utlestat. By removing some of the drawbacks, Statspack provides for a solid method of collecting statistics relating to the Oracle database server.

Benefits of Statspack

Following are some of the benefits:

1. Statspack provides the ability take statistics at regular intervals and provides reporting to compare two different snapshots.

2. It is easier to work with since the statistics collection process is different from the reporting process. As we will see in the subsequent sections, a snapshot of database statistics is collected using the procedure called snap. Both standard and custom reporting can be done at any time from snapshots collected separately.

3. Helps to establish baseline timings and comparison of current statistics with the baseline to identify performance issues.

4. Statspack can be run in an OPS environment where we can connect to specific instances to collect statistics on them and keep a central repository of statistics.

5. The database statistics are stored permanently in tables and can be used for custom reporting. This increases the value of Statspack since we can develop many different reports for trends and proactive tuning.

6. Statspack provides the ability to collect snapshots at more frequent intervals when there is a performance issue and still maintains the regular collection process. For example, under the normal circumstances, we can set up a process to collect statistics every hour but when working on a specific production performance issue, snapshots can be taken every half hour or even for a single session.

How does Statspack Work?

SNAPSHOT

A snapshot refers to a single collection of performance data. A snapshot is collected by executing the snap procedure of the statspack package once. This can be done using the command:

```
execute statspack.snap;
```

Oracle generates a SNAP_ID for every snapshot and stores more information about the snapshot itself in the table called STATS$SNAPSHOT.

Below is a list of tables in Version 8.1.7 with a brief description of their purpose. See Figure 1.

SNAPSHOT Levels

Statspack can be executed in three levels.

Level 0

Statspack collects overall statistics about the database including wait statistics, library cache, rollback segment, I/O, etc.

Level 5

Level 5 is the default level for Statspack. In this level, in addition to level 0 statistics, Statspack also gathers resource-intensive SQL statements. The thresholds for identifying a SQL statement that is resource intensive can be tuned. This is helpful to restrict your list and avoid collecting huge amounts of statistics. The idea here is to tune only the top offenders.

Following are the thresholds that can be changed:

- the number of executions — default value is 100
- the number of disk reads — default values is 1000
- the number of parse calls — default value is 1000
- the number of buffer gets — default values is 10000

Level 10

Level 10 takes longer to execute and should only be used on advice from Oracle Support. There might be some performance impacts of running Statspack with level 10.

Installation Steps for Statspack

Statspack Scripts and their Functions

Figure 2 (next page) shows various scripts Oracle provides for installation of Statspack along with their functions. Starting with Oracle 8.1.6, these scripts are available under $ORACLE_HOME/rdbms/admin directory. Look for sp*.sql scripts. If you are planning to use Statspack with either version 8.0.x or 8.1.5, make sure you download the scripts and copy them under $ORACLE_HOME/rdbms/admin or in the dba scripts directory.

It is a good idea to create a separate tablespace for statspack since you can have a better control on the tables.

Step-by-Step Installation

Step 1

Create a new tablespace to store the schema created by Statspack:

```
CREATE TABLESPACE STATSPACK DATAFILE
'/u01/oradata/finp11i/statspack1.dbf' SIZE
1024M
DEFAULT STORAGE- (INITIAL 1M NEXT 1M
PCTINCREASE 0);
```

It is a good idea to create a separate tablespace for statspack since you can have a better control on the tables. Also keep your initial extent and next extent small since some of the create index statements do not have storage parameters assigned to them. You can also create this tablespace as locally managed to make the extent management local.

Step 2

Execute the script to install statspack:

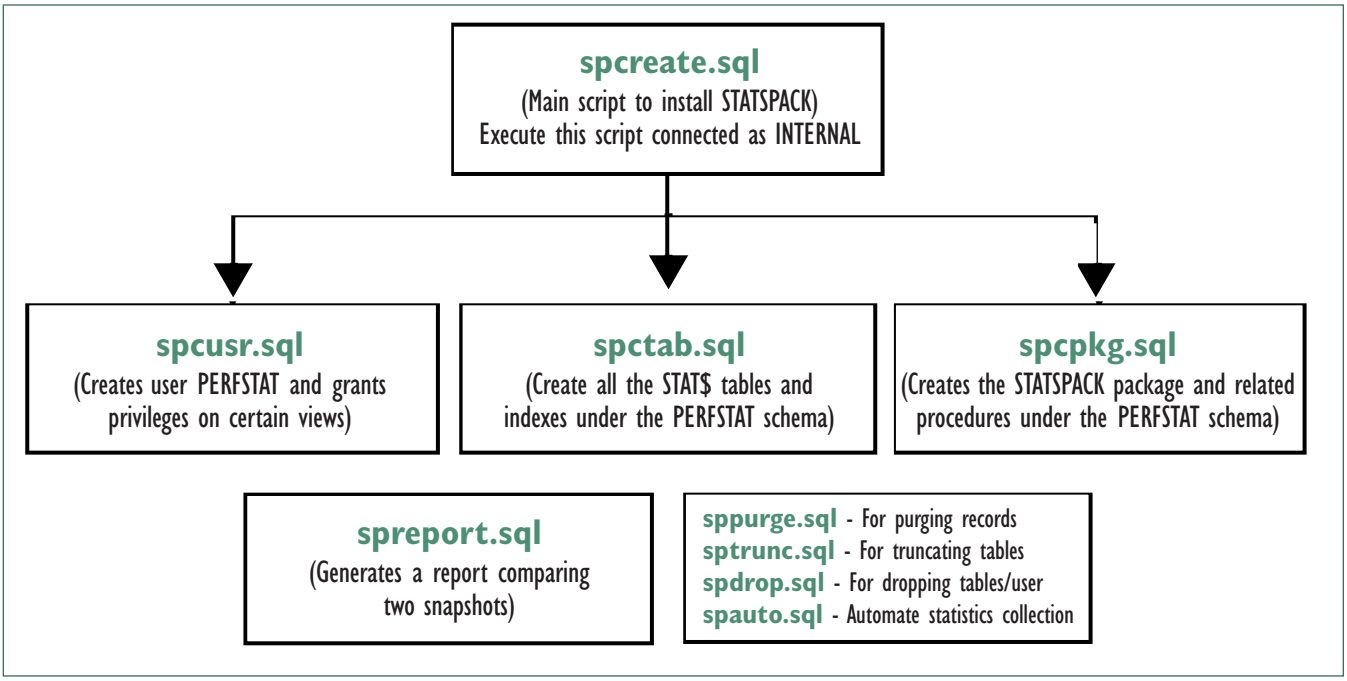
```
sqlplus internal
sql>$ORACLE_HOME/rdbms/admin/spcreate
```

This script creates a user called PERFSTAT and create all the STATS$ tables and indexes under PERFSTAT. The script also creates the statspack

Figure 1.

Table Name	Contents
STATS$BG_EVENT_SUMMARY	Summary of waits and timeouts
STATS$BUFFER_POOL	Not Available in 8.1.7
STATS$BUFFER_POOL_STATISTICS	Buffer Pool Statistics Information
STATS$DATABASE_INSTANCE	Initialization Parameters
STATS$ENQUEUESTAT	Enqueue Statistics for various snapshots
STATS$FILESTATXS	File I/O Statistics
STATS$IDLE_EVENT	-
STATS$LATCH	Latch Information
STATS$LATCH_CHILDREN	
STATS$LATCH_MISSES_SUMMARY	
STATS$LEVEL_DESCRIPTION	
STATS$LIBRARYCACHE	Library Cache statistics for each snapshot
STATS$PARAMETER	Statspack parameters information
STATS$ROLLSTAT	Rollback segment statistics
STATS$ROWCACHE_SUMMARY	
STATS$SESSION_EVENT	Session Event information
STATS$SESSTAT	Session Statistics
STATS$SGA	
STATS$SGASTAT	
STATS$SNAPSHOT	Information about the snapshot e.g. start time
STATS$SQLTEXT	New Table in 8.1.7
STATS$SQL_STATISTICS	New Table in 8.1.7
STATS$SQL_SUMMARY	
STATS$STATSPACK_PARAMETER	Current level and threshold values for various parameters
STATS$SYSSTAT	System Statistics
STATS$SYSTEM_EVENT	System Events for various snapshots
STATS$TEMPSTATXS	New Table in 8.1.7
STATS$WAITSTAT	Wait Statistics for various snapshots

Figure 2: Statspack version 8.1.7 scripts and their functions.



package under PERFSTAT. Make sure you specify the default tablespace for PERFSTAT as the PERFSTAT tablespace created in Step 1 above and the appropriate temporary tablespace.

Step 3

At this stage, you are ready to take snapshots of database statistics and save them to various tables under the PERFSTAT schema. You can either execute a snapshot from the SQL prompt or setup a cron job or a batch process to periodically execute the snapshot. Refer to the subsequent section for a discussion on setting up a cron job. For taking a snapshot execute the following commands:

```
execute statspack.snap;
```

Step 4

For executing the standard report available with Statspack execute the following script.

For version 8.1.5 onwards:

```
sqlplus perfstat/perfstat
sql>@$ORACLE_HOME/rdbms/admin/statsrep
```

For version 8.1.7:

```
sqlplus perfstat/perfstat
sql>@$ORACLE_HOME/rdbms/admin/spreport
```

The script first displays all the available snapshots (SNAP_ID and SNAP_TIME). It accepts two snapshots and then proceeds to print a comparative report for that window of time. The script also displays a report file name which can be modified. The default for the report name is st_<starting-snap-id>-<ending snap_id>.lst

The statsrep report prints a detailed report similar to utlbstat/utlestat. The first page of the report gives an overall summary of the database and has sections on cache sizes, load profiles, instance efficiency percentages and top five wait statistics. This first page can be used to analyze specific causes of contention eg. buffer waits, extended wait on db file scattered read etc. and the subsequent section of the report provides more details on aspects like library cache, rollback segment, I/O statistics, wait statistics as well as resource intensive SQL statements individually listed out ordered by executions, parse calls, buffer gets and disk reads.

Figure 3.

```
#####################################################################
# snap.ksh
# Shankaran Iyer Sept, 2000
# This script executes the snapshot procedure for collecting
# database statistics using STATSPACK
# This script can be executed using cron to run at regular times
#####################################################################
echo 'Snapshot Started at `date`
ORACLE_HOME=/u01/app/oracle/8.1.7
ORACLE_SID=finp11i
export ORACLE_HOME
export ORACLE_SID
$ORACLE_HOME/bin/sqlplus -s <<EOF
perfstat/perfstat
execute statspack.snap;
exit;
EOF
echo 'Snapshot Completed at `date`
```

CRONTAB ENTRY

The following entry will execute the snap.ksh every hour to collect snapshot:

```
00 0-23 * * * $ORACLE_BASE/admin/scripts/snap.ksh > /dev/null 2>&1
```

Automating the Process of Gathering Snapshots

Snapshots can be collected at periodic intervals (say every hour) by executing the snap procedure. The process can be automated either by using a cron job (on UNIX) or at (on Windows) or using a DBMS_JOB script that is part of the statspack scripts. Statspack includes a script called statsauto.sql (8.1.5) or spauto.sql (8.1.7) to help with the setup. Here we will illustrate automating this process using a cron job. We will develop a wrapper program called snap.ksh which will execute the statspack.snap procedure. The snap.ksh script will then be set up as a cron job to be run periodically. See Figure 3.

Sending Statspack Report by E-Mail

After having set up a process for collecting statspack periodically, we can also setup a process to email the statspack report to the administration team. This may or may not be required based on the size of the application and number of administrators actually involved in the tuning process. In any case, sending an email of the one-page overall statistics to critical project folks like the project lead, architect and the functional lead could be a useful method of communicating the health of the environment.

For implementing this, we can set up statsrep to run periodically as a cron job (or similar scheduling mechanism) and use a command similar to the following for emailing the report to the team.

```
mailx -s "Statspack Report" <E-Mail-ID>
</u04/logs/DBA/statsrep.log
```

Customizing Statspack Reports

The stats$ table can be used for custom reporting in a number of ways. We can develop a plethora of scripts to understand different statistics at various points. Using them we can see if there is a need to tune the environment differently for batch runs or during the daytime. By comparing specific statistics over a period of time we can not only establish our requirements in terms of capacity but also pinpoint which parameters need to be changed for optimal performance. Three examples of using statspack data and performing trend analysis are:

1. Overall database performance ratio
2. Analyzing hot datafiles for balancing I/O
3. Analyzing recursive calls

Conclusion

Actual execution time for running the snap package on a large production machine running 8.1.x is about 10-20 seconds. The response from the user's perspective is minimal. Adjusting the SQL statement thresholds to capture only the most critical statistics can further reduce the execution time. The effect of a snapshot for a reasonably large hardware environment running Oracle Applications should be minimal. Once the basic data capture has been completed and baselines have been established, frequency of Statspack can be further reduced. Statspack can help identify if the resources are being optimally used. Also, when comparing two snapshots, make sure that they reflect the same type of activity. This is just to ensure that you are comparing like variables. Statspack provides a more robust way of collecting and reporting database statistics.

About the Author

Shankaran Iyer
Technology Manager
Renaissance Worldwide, Inc.
2850 Metro Drive, Suite 800
Bloomington, MN 55425
Phone : (952)851-3160
Fax : (952)851-3161
E-mail : siyer@rens.com

Shankaran has been in the IS industry for over fifteen years. He has been working with Oracle Server, Tools and Applications for over twelve years. His primary focus areas are performance tuning, Internet-based application deployment and data warehousing. Shankaran is a regular presenter at the Oracle Conference in the US and Europe. ▲

This article is excerpted and reprinted with permission from the IOUG-A Live! Conference, April 2001.

TECH TIPS

Using XML With Oracle

Don't miss out on a worthwhile white paper on XML by Database Specialists' president and Oracle expert, Roger Schrag. His paper is a quick read that will get you started on understanding XML and how to go about incorporating it into an Oracle environment.

XML, short for "extensible markup language," is emerging as a new standard for information exchange, and there are numerous benefits to using it with Oracle. Although it has similarities to HTML, XML has powerful extensibility built into the standard that enables easy exchange of complex structured data between disparate systems.

Many content providers (such as Reuters) use XML to publish their content so that subscribers can easily present the content to their end-users as HTML, WML, or using just about any file format.

You can download this white paper by visiting the Database Specialists website at http://www.dbspecialists.com/4dbas/present.html.

EDUCATION OPPORTUNITY

IOUG-A University Master Classes Now in Their Third Year!

Check out this great learning opportunity that will make you immediately more efficient and effective. The IOUG-A University Master Class (UMC) faculty are recognized authorities in their areas of Oracle expertise. Not only are they leaders in their own companies, but they are also masters of the technology they are teaching. Their years of experience enable them to teach in-depth, real life tips, tricks and best practices on topics that you need to do your job. The two-day seminars consist of intensive learning, delivered in a team format by the UMC faculty.

The next class series will be held in Las Vegas.

October 14-16, Las Vegas, NV
Beginner DBA Toolkit
NEW! Advanced DBA Bootcamp

October 16-18, Las Vegas, NV
Developer Web Toolkit
NEW! iDBA Primer

Course Descriptions

Beginner DBA Toolkit

Are you a new DBA with less than two years of practical experience? Are you looking for the tips and techniques that take years to learn by trial and error, but you need the knowledge NOW? Then this is the UMC for you! Join our experienced faculty as they cover the following key topics:

- *Beginner DBA Best Practices and Backup & Recovery - Ari Kaplan, PocketDBA Systems*
- *Beginner Tuning - Gaja Krishna Vaidyanatha, Quest Software, Inc.*
- *Beginner PL/SQL - John Beresniewicz, Precise Software Solutions*

Advanced DBA Bootcamp

Have you been a working DBA for more than two years and know all the basics, but are now expected to do even MORE in your job than just backup and recovery? Are topics like data warehouse architecture and system speed improvement your next steps? Then this UMC fits your needs! Our expert faculty has designed this class to cover the following key topics:

- *Advanced DBA Best Practices - Michael Abbey, The Pythian Group*
- *Advanced Backup & Recovery - Babette Turner-Underwood, The Pythian Group*
- *Advanced Tuning Techniques - Rich Niemiec, TUSC*
- *Data Warehousing - Ian Abramson, Data Visions Inc.*

iDBA Primer

Just what is an iDBA? Well, if you have to deal more and more with gathering or deploying data over the web, then YOU are an iDBA! And, along with all of the other "routine" responsibilities of managing your company's data, you'll also be responsible for new issues like security, high availability, new languages and Internet servers. The iDBA UMC faculty knows your challenges and will be addressing them in following topic areas:

- *iDBA: The Issues - Roger Schrag, Database Specialists, Inc.*
- *Java and the iDBA - George Trujillo, Trubix*
- *Application Server - Charles Kim, DMS, Inc.*
- *iDBA Management - Bill Burke, i2 Technologies, Inc.*

Developer Web Toolkit

You've heard it over and over: "The web changes everything." And you've been working to develop applications that are deployed in the Internet already! And as the web changes everything – the web itself changes too! You must be current on all the tools needed to stay on top of this ever-evolving industry. From Java and JDeveloper to WebDB, Forms and PL/SQL – this UMC has it all:

- *PL/SQL Tips and Techniques - Joe Trezzo, TUSC*
- *Oracle Portal Development - Steve Vandivier, AVANCO International, Inc.*
- *Developing Java Database Applications in Oracle8iR3 - George Trujillo, Trubix*
- *Web Forms & JDeveloper - Peter Koletzke, Quovera*

Find course outlines and pricing at: **www.ioug.org** ▲

Comparing Join Techniques

(continued from page 17)

4) The outer table may also need to be sorted by the joining columns so that both tables to be joined are sorted in the same manner. This step is also optional and dependent on whether the outer table is already well ordered by the keys and whether efficient index access can be used.

5) Read both outer and inner tables (these may be the sorted temporary tables created in previous steps), choosing rows that match the join criteria. This operation is very quick since both tables are sorted in the same manner and Database Prefetch can be used.

6) Optionally sort the data one more time if a sort was performed (e.g. an 'order by' clause) using columns that are not the same as were used to perform the join.

The Merge join can be deceivingly fast due to database multi-block fetch (helped by initialization parameter db_file_multiblock_read_count) capabilities and the fact that each table is accessed only one time each. These are only used for equi-joins. The other init.ora parameter that can be tuned to help performance is sort_area_size.

Hash Join

The Hash join is a very efficient join when used in the right situation. With the Hash join, one table is chosen as the outer table. This is the larger of the two tables in the join – and the other is chosen as the inner table. Both tables are broken into sections and the inner tables join columns are stored in memory (if hash_area_size is large enough) and 'hashed'. This hashing provides an algorithmic pointer that makes data access very efficient. Oracle attempts to keep the inner table in memory since it will be 'scanned' many times. The outer rows that match the query predicates are then selected and for each outer table row chosen, hashing is performed on the key and the hash value is used to quickly find the matching row in the inner table. This join can often outperform a Sort Merge join, particularly when one table is much larger than another. No sorting is performed and index access can be avoided since the hash algorithm is used to locate the block where the inner row is stored. Hash joins are also only used for equi-joins. Other important init.ora parameters are: hash_join_enabled, sort_area _size and hash_multiblock_io_count.

Star Joins

A particular type of join common to data marts and data warehouses is known as the "star join" or "star-query". This is a join of a large "Fact" table with two or more smaller tables commonly called "Dimensions". Fact tables may be thought of as having transactional properties. An example of a Fact table is Sales which contains keys and some measures. This is commonly a large table with millions – and sometimes billions – of rows.

> ***The Merge join can be deceivingly fast.***

The Dimensional tables are used to describe the Fact table. Examples of Dimension tables are: Customer, Product and Time. Star queries get their name because there is a single Fact table in the middle and the smaller dimensional tables are directly related to the Fact table. To explain how Star joins work, let's consider an example of a central Fact table and three small Dimensional tables related to the Fact table. The first and most simple approach for resolving a star-query is to query the three-dimensional tables using the predicates in the Where clause to filter out unnecessary rows. The resulting rows from the three-dimension tables can be joined together forming a Cartesian product. This is not as bad as it sounds since these tables are relatively small and the rows matching the 'Where' criteria account for a subset of these. This Cartesian product can be sorted and stored as a temporary table. A merge-scan join can then be performed between the resulting temporary table and the Fact table.

There is a more complex approach that the Database optimizer can take towards Star joins that is used in Oracle8i. Consider the case of the central Fact table that is being joined to three smaller Dimensional tables as shown below:

```
Select *
From Fact, Dim1, Dim2, Dim3
Where Fact.dim1_id = Dim1.id
and Fact.dim2_id = Dim2.id
and Fact.dim3_id = Dim3.id
and Dim1.name like :input_variable_name
and Dim2.Descriprion between :input_var1
and :input_var2
and Dim3.Text < :input_variable_text;
```

The optimizer can transform this by adding three subqueries so that the new transformed query looks like this:

```
Select *
From Fact, Dim1, Dim2, Dim3
Where
Fact.dim1_id in (Select dim1.id from dim1
where dim1.name like :input_variable_name)
and Fact.dim2_id in (Select dim2.id from
dim2
where Dim2.descriprion between :input_var1
and :input_var2)
and Fact.dim3_id in (Select dim3.id from
dim3
where Dim3.Text < :input_variable_text);
```

Given the above, the subselects are performed first. If bitmap indexes exist on the Fact join columns, then the bitmap index entries that result from each subquery can be merged (in this case ANDed) together and the Fact table data can be retrieved using the resulting index values. This is a very quick operation. The Fact entries retrieved can then be joined to the Dimension tables to complete the query. Using this approach, a Cartesian product is not required. To implement star_query transformation:

- Set init.ora parameter star_transformation_enabled
- Create bitmap indexes for all of the foreign-key columns on the fact table
- Implement R.I. between the fact table and the dimension tables. ▲

NoCOUG Summer Conference

Thursday, August 23, 2001
Session Descriptions

KEYNOTE

9:45-10:30
OTN as a Developer Service Provider
Robert Dell'Immagine,
Director, Oracle Technology Network, Oracle Corporation

In this presentation, Robert will go into some detail about how Oracle Technology Network, OTN, is evolving from a technical information source into a provider of services (such as Oracle Portal Studio and Oracle Mobile Studio) that increases developer and DBA productivity and reduces costs. Come get a good inside view into what's happening at OTN.

DBA TRACK

11:00-12:00 & Continues 1:15-2:15
Oracle Performance Management - A Radical Approach
Gaja Krishna Vaidyanatha,
Director of Storage Management Products, Quest Software

Oracle performance tuning has developed a reputation as part science, part art, and part wizardry. This seminar will impart the core principles of performance management by sharing a step-by-step process of iteratively investigating, determining, and implementing tuning solutions using a proven methodology. This time-tested and field-proven methodology looks at performance management as having a system-wide scope. Further it steers you away from the old bad habit of simply throwing more memory at Oracle's shared memory areas. This seminar propagates the idea that the bulk of possible performance gains can be achieved quickly and with minimal effort, if you truly know how to identify and understand the nature of your system and more importantly its bottlenecks. And we will do this without checking one infamous cache hit ratio within Oracle.

3:00-4:00
Oracle Advanced Queuing: An Overview
Hamid Minoui, Oracle DBA, Fritz Companies, Inc.

In this presentation Hamid will discuss Oracle's advanced queuing facility, initially at a high level and then drilling down into detailed examples. Topics will include messaging concepts, the differences between synchronous and asynchronous messaging architectures, Oracle's features to support queuing, APIs and commands for creating queues and enqueuing and dequeuing messages, and an example that walks through the process of queue setup and use.

DATA WAREHOUSE TRACK

11:00-12:00 & Continues 1:15-2:15
Semantic Integration of Data from Multiple Sources into a Common Database
Michael Scofield, *Director of Data Quality, Experian Information Solutions*

More and more, data stewards are being asked to integrate data from multiple, dissimilar sources into a common database. This can be because of mergers, acquisitions, CRM efforts, or building other types of data warehouses. To accurately map source field to target (as well as to design the target database), the analyst must completely understand the full semantic meaning of each source data element, and how it behaves over the entire scope of source data. It is not enough to note similarity of format and field-name. This presentation will demonstrate practical techniques for the evaluation of source data (sometimes known as data profiling or domain studies) and show examples of the kinds of surprises which can lurk in production databases which are not thoroughly examined. We show the folly of relying solely on old documentation or data directories. Finally, we will survey techniques of establishing an on-going data surveillance program to ensure that later production-ized loads of data will not be caught by surprise when a source changes definitions or scope of the data it supplies.

3:00-4:00
SQL Optimization for the Data Warehousing Environment
Frank Irizawa, *Lecco Technology*

The key to top-performing database applications is found in the heart of the application: the SQL statement. Studies show that more than half of database performance problems are a direct result of under-performing SQL statements. This highlights the importance of optimizing your SQL to prevent database performance problems. This presentation shows a SQL tuning methodology for data warehousing and gives a detailed overview of techniques for tuning SQL statements to optimize the performance of data warehouse applications. Principles of query optimization and Oracle optimization features for data warehousing are presented. Attendees will see query tuning tips with multiple examples.

DEVELOPER TRACK

3:00-4:00
A Close Look at SQL Statement Tuning
Dan Hotka, *Director of Database Field Operations, Quest Software*

This presentation goes into considerable detail on the tuning of actual SQL statements. The attendee will understand the Oracle optimizer choices, the Explain Plan, and how to interpret the output of an Explain Plan. This presentation will also include a discussion on the differences of the various Explain Plan steps such as Merge-Join and Nested-Loops, when is it best to use each, etc. The presenter also discusses how indexes are accessed with the cost-based optimizer as well as some important index-oriented init.ora parameters. ▲

Thank you Chevron!

We appreciate your hosting our Summer conference.

NoCOUG Summer Conference

Thursday, August 23, 2001

Location: Chevron Park, Building A, San Ramon
(Building A is located at 6001 Bollinger Canyon Road in San Ramon.)

Our Summer Conference takes place at Chevron Park's Building A in San Ramon

8:00-9:00	**Registration and welcome. Refreshments served.**
9:00-9:45	**Opening Remarks and Announcements / Vendor Introduction**
9:45 -10:30	***Keynote:* OTN as a Developer Service Provider**
10:30-11:00	**Morning Break**
11:00-12:00	**Parallel Sessions** *(tracks continued in afternoon)*
	DBA Track: *Oracle Performance Management - A Radical Approach*
	Data Warehouse Track: *Semantic Integration of Data from Multiple Sources into a Common Database*
12:00-1:15	**Lunch Break**
1:15-2:15	**Parallel Sessions** *(tracks continued from morning sessions)*
	DBA Track: *Oracle Performance Management - A Radical Approach*
	Data Warehouse Track: *Semantic Integration of Data from Multiple Sources into a Common Database*
2:15-3:00	**Afternoon Break**
3:00-4:00	**Parallel Sessions**
	DBA Track: *Oracle Advanced Queuing: An Overview*
	Developer Track: *A Close Look at SQL Statement Tuning*
	Data Warehouse Track: *SQL Optimization for the Data Warehousing Environment*
4:15-???	**Reception/No Host Bar—O'Kane's in San Ramon**
Cost:	$40 admission fee for non-members. Includes lunch voucher. Members free.

For session descriptions, check out page 27.

Don't miss our keynote speaker from Oracle Technology Network!

RSVP online at http://www.nocoug.org/rsvp.htm

NoCOUG
P.O. Box 3282
Danville, CA 94526