

# **Silly Little Oracle Benchmark**

Another episode in the popular  
series *“Troy’s Tools”*

*Troy Ligon*



**How do I determine the max IOPS  
my system can push?**

# **Lots of Ways**

---

- ⇒ **Oracle-provided Tools**  
(AWR, ORION, SwingBench, calibrate\_io)
- ⇒ **Commercial Tools**  
(Quest Benchmark Factory,  
Mercury Interactive LoadRunner, etc.)
- ⇒ **Free Tools**  
(iometer, HammerDB, SLOB, etc.)

# **AWR**

---

## **PROS:**

- **Already in place for most environments**
- **Text or HTML output**

## **CONS:**

- **10g and up only**
- **Needs license**
- **General tool and Cannot adjust output**
- **Only measures, doesn't generate load**

# ORION

---

## **PROS:**

- **Author - Kevin Closson**
- **Uses Oracle I/O subroutines**
- **Easy to execute**
- **Included with 11g, download for earlier**

## **CONS:**

- **Operates on raw filesystems**
- **Destructive test**

(see chapter 8 of Database  
Performance Tuning Guide)

# SWING BENCH

---

## **PROS:**

- **Author - Oracle UK's Dominic Giles**
- **GUI interface w/Real-time output**
- **TPC or custom loads**

## **CONS:**

- **Difficult to configure and get up and running**

`http://www.dominicgiles.com/  
Swingbench.pdf`

# CALIBRATE\_IO

---

## **PROS:**

- **Supplied PL/SQL utility**
- **Easy to execute**

## **CONS:**

- **Only available for 11g**
- **Only does reads**

(see PL/SQL Packages and Types Reference)

# DBMS\_RESOURCE\_MANAGER CALIBRATE\_IO

---

SET SERVEROUTPUT ON TIMING ON

DECLARE

lat INTEGER;

iops INTEGER;

mbps INTEGER;

BEGIN

-- DBMS\_RESOURCE\_MANAGER.CALIBRATE\_IO

-- (num\_physical\_disks, max\_latency, iops, mbps, lat);

DBMS\_RESOURCE\_MANAGER.CALIBRATE\_IO (2, 10, iops, mbps, lat);

dbms\_output.put\_line('max\_iops = ' || iops);

dbms\_output.put\_line('latency = ' || lat);

dbms\_output.put\_line('max\_mbps = ' || mbps);

end;

/



# **DBMS\_RESOURCE\_MANAGER CALIBRATE\_IO**

---

**max\_iops = 6278**

**latency = 11**

**max\_mbps = 839**

**Elapsed: 00:15:04.16**

# DBMS\_RESOURCE\_MANAGER CALIBRATE\_IO

---

```
select * from V$IO_CALIBRATION_STATUS;
```

STATUS	CALIBRATION_TIME
--------	------------------

-----	-----
-------	-------

NOT AVAILABLE	
---------------	--

IN PROGRESS	
-------------	--

READY	25-JUN-13 11.31.17.022 AM
-------	---------------------------

# DBMS\_RESOURCE\_MANAGER CALIBRATE\_IO

---

```
select * from DBA_RSRC_IO_CALIBRATE;
```

START_TIME			END_TIME	
-----			-----	
MAX_IOPS	MAX_MBPS	MAX_PMBPS	LATENCY	NUM_PHYSICAL_DISKS
-----	-----	-----	-----	-----
25-JUN-13	11.16.12.974357	AM	25-JUN-13	11.31.17.022466 AM
6278	839	763	11	2

# Issues:

- ⇒ Not available below 11g
- ⇒ Only does reads
- ⇒ How many disks to specify
- ⇒ What max latency to specify

# Lots of Ways

---

⇒ **Oracle-provided Tools**  
(AWR, ORION, SwingBench, calibrate\_io)

⇒ **Commercial Tools**  
(Quest Benchmark Factory,  
Mercury Interactive LoadRunner, etc.)

⇒ **Free Tools**  
(iometer, HammerDB, SLOB, etc.)

# Commercial Tools

---

## **PROS:**

- **Typically GUI**
- **Typically Easy to Configure**
- **Typically Easy to Execute**

## **CONS:**

- **Cost Money**
- **Can be difficult to configure**

# Lots of Ways

---

- ⇒ **Oracle-provided Tools**  
(AWR, ORION, SwingBench, calibrate\_io)
- ⇒ **Commercial Tools**  
(Quest Benchmark Factory,  
Mercury Interactive LoadRunner, etc.)
- ⇒ **Free Tools**  
(iometer, HammerDB, SLOB, etc.)

# IOMETER

---

## **PROS:**

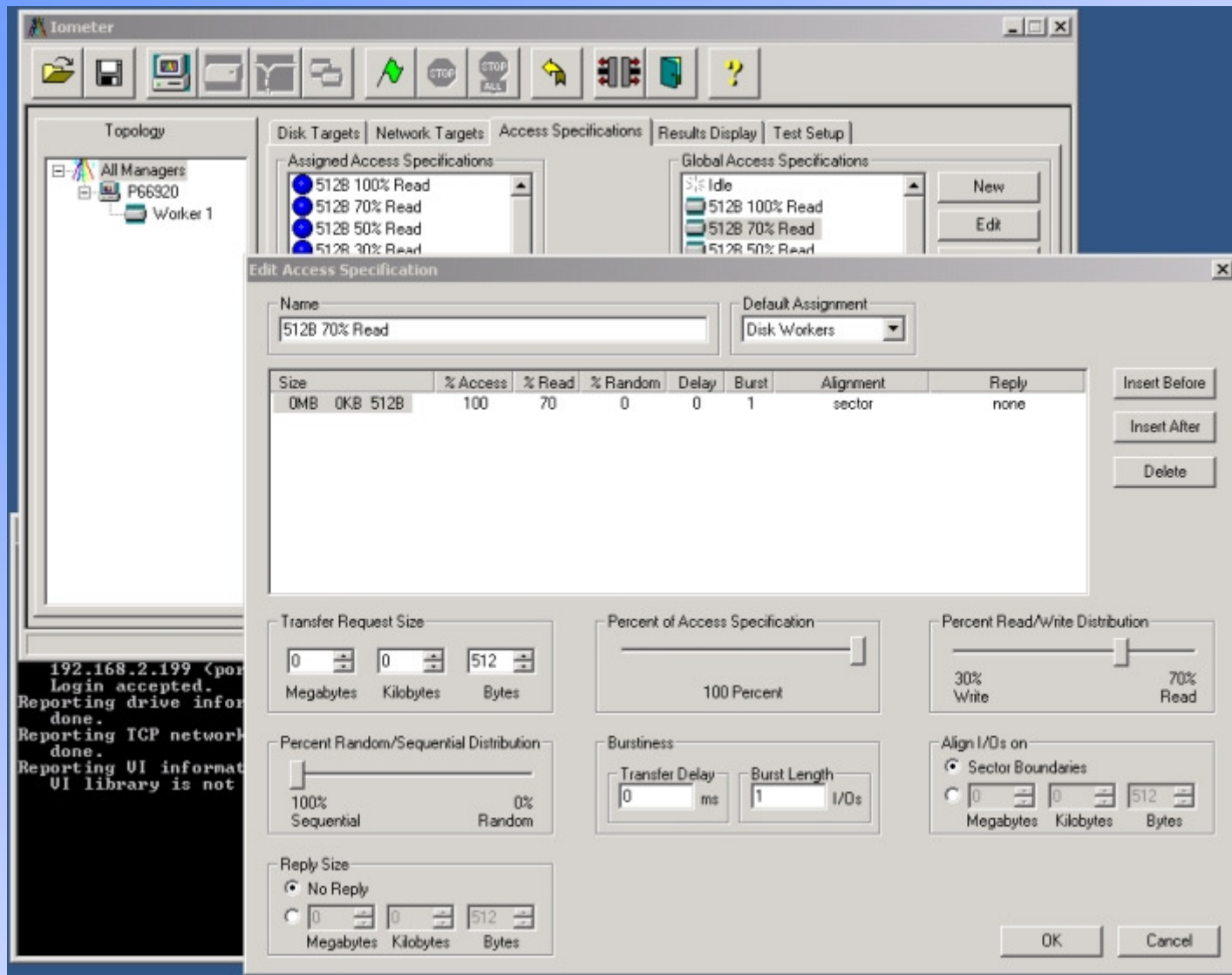
- Originally developed by Intel, now Open Source
- Linux, Mac, Windows, Solaris
- GUI interface – easy setup

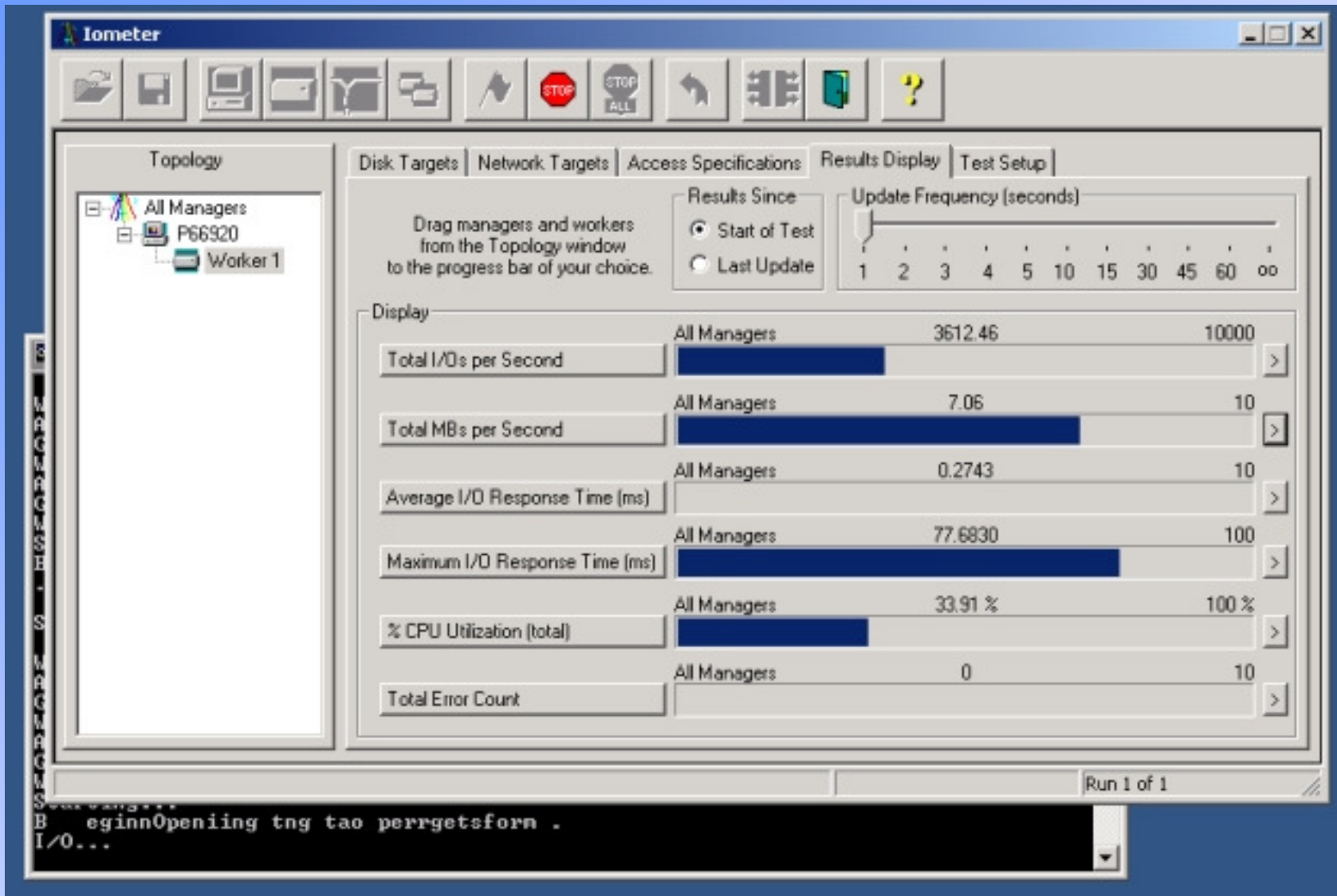
## **CONS:**

- Not Oracle specific
- Destructive test

<http://www.iometer.org/>









# HammerDB

---

## **PROS:**

- Originally HammerOra, now multi-DB
- GUI interface – easy setup

## **CONS:**

- Not Oracle specific
- TPC-C test
- Not as accurate as other tools

`http://hammerora.sourceforge.net/  
index.html`

**7% HAMMERORA**

File Edit Options Help

**Benchmark**

- Oracle
  - TPC-C
    - Schema Build
    - Options
    - Build**
    - Driver Script
    - Virtual User
    - Autopilot
    - Transactions
    - Mode
  - SQL Server
  - MySQL

**Script Editor** **Virtual User Output** **Transaction Counter** **Autopilot**

**RUNNING - TPC-C creation**

Virtual User	Iterations	Complete	Status
1	1	0	
2	1	0	
3	1	0	

loading history file ... 43 events added  
 Main console display active (Tcl8.5.9 / Tk8.5.9)  
 The xml in config.xml is well-formed, applying variables  
 (hammerora-2.8) 44 %

File: TPC-C creation Mode: Local Row.Col: 1.0

76 Oracle TPC-C Build Options

Build Options

Oracle Service Name : xe

System User Password : oracle

TPC-C User : tpcc

TPC-C User Password : tpcc

TPC-C Default Tablespace : users

TPC-C Temporary Tablespace : temp

Number of Warehouses : 5

Virtual Users to Build Schema : 2

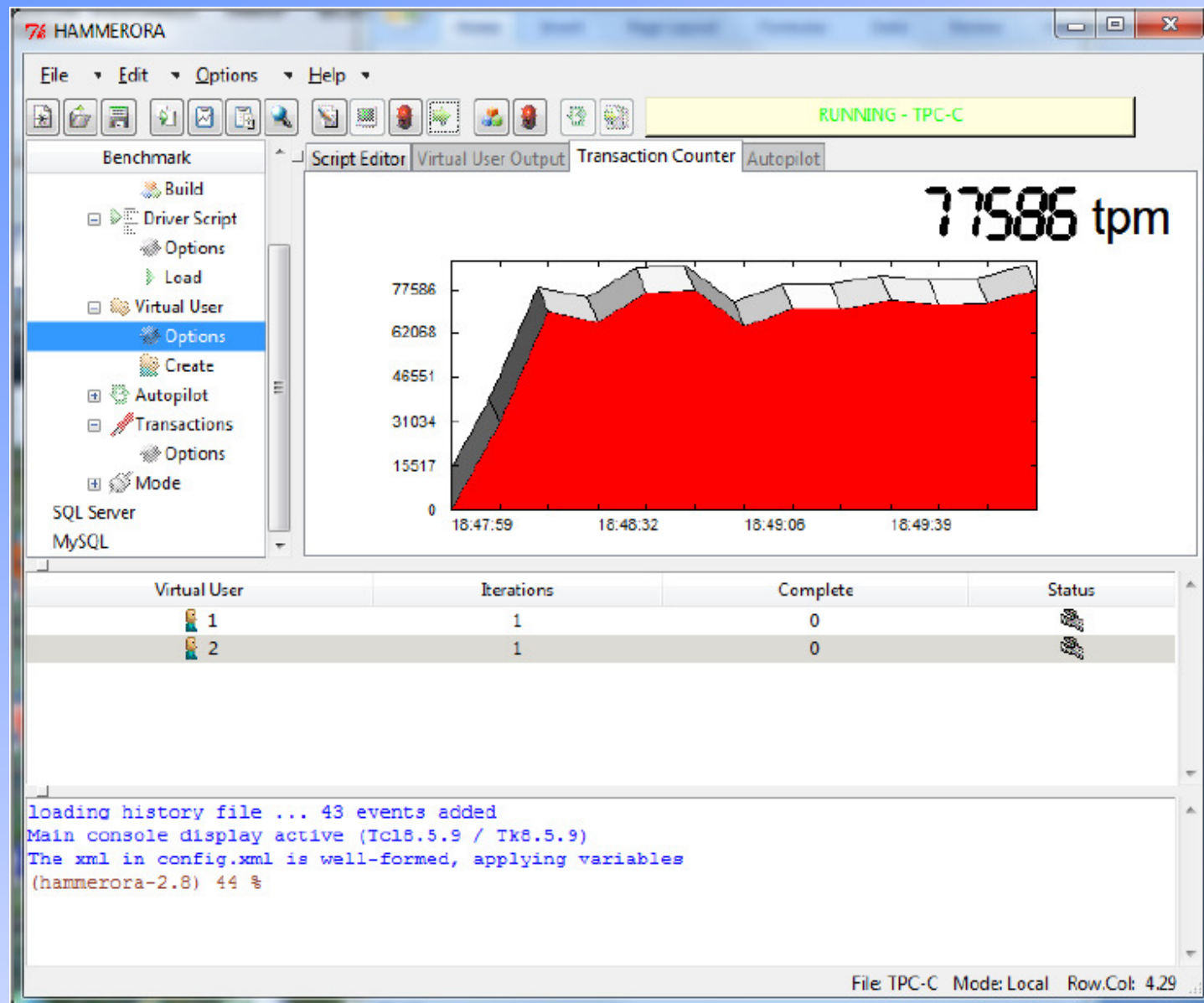
Use PL/SQL Server Side Load : ☐

Server Side Log Directory : C:/TEMP

Partition Order Line Table : ☐

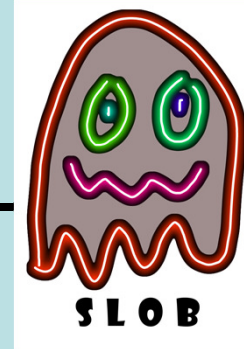
OK Cancel





# SLOB

---



## **PROS:**

- **Author – Kevin Closson (now with EMC)**
- **Small footprint**
- **Simple**

## **CONS:**

- **No application workloads**

`http://kevinclosson.wordpress.com/2012/02/06/introducing-slob-the-silly-little-oracle-benchmark/`



# Prepare the Database

---

```
==> sqlplus "/ as sysdba"
```

```
alter system set db_cache_size = 10M scope=spfile;  
alter system set filesystemio_options = setall scope=spfile;  
alter system set parallel_max_servers = 0 scope=spfile;  
alter system set cpu_count = 1 scope=spfile;  
alter system set recyclebin = off scope=spfile;  
alter system set "_db_block_prefetch_limit" = 0 scope=spfile;  
alter system set "_db_block_prefetch_quota" = 0 scope=spfile;  
alter system set "_db_file_noncontig_mblock_read_count" = 0 scope=spfile;
```

```
create BIGFILE tablespace IOPS datafile '/orav/iops.dbf' size 200m autoextend  
on next 200m maxsize 30000m  
NOLOGGING ONLINE PERMANENT EXTENT MANAGEMENT LOCAL  
AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO ;
```

```
startup force
```

# Unpack the Tar

---

```
==> tar -xvf 2013.05.05.slob2.tar
SLOB/
SLOB/runit.sh
SLOB/setup.sh
SLOB/slob.conf
SLOB/slob.sql
SLOB/README
SLOB/simple.ora
SLOB/wait_kit/
SLOB/wait_kit/trigger.c
SLOB/wait_kit/create_sem.c
SLOB/wait_kit/Makefile
SLOB/wait_kit/mywait.c
....
```

# Adjust SLOB.CONF

---

```
==> cat slob.conf
```

```
UPDATE_PCT=20  
RUN_TIME=300  
WORK_LOOP=0  
SCALE=10000  
WORK_UNIT=256  
REDO_STRESS=HEAVY  
LOAD_PARALLEL_DEGREE=8  
SHARED_DATA_MODULUS=0
```

```
# Settings for SQL*Net connectivity:  
#ADMIN_SQLNET_SERVICE=slob  
#SQLNET_SERVICE_BASE=slob  
#SQLNET_SERVICE_MAX=2  
#SYSDBA_PASSWD="change_on_install"
```

```
export UPDATE_PCT RUN_TIME WORK_LOOP SCALE WORK_UNIT  
LOAD_PARALLEL_DEGREE REDO_STRESS SHARED_DATA_MODULUS
```



# Make the WAIT\_KIT

---

```
==> cd SLOB/wait_kit
==> make
rm -fr *.o mywait trigger create_sem
cc      -c -o mywait.o mywait.c
cc -o mywait mywait.o
cc      -c -o trigger.o trigger.c
cc -o trigger trigger.o
cc      -c -o create_sem.o create_sem.c
cc -o create_sem create_sem.o
cp mywait trigger create_sem ../
rm -fr *.o
```

# Build the Test Users

---

**==> ./setup.sh <tablespace> <users>**

==> ./setup.sh iops 10

NOTIFY: Load Parameters (slob.conf):

LOAD\_PARALLEL\_DEGREE == 8

SCALE == 10000

ADMIN\_SQLNET\_SERVICE == ""

CONNECT\_STRING == "/ as sysdba"

NON\_ADMIN\_CONNECT\_STRING ==

NOTIFY: Testing connectivity to the instance to validate  
slob.conf settings.

NOTIFY: ./setup.sh: Successful test connection: "sqlplus -  
L / as sysdba"

NOTIFY: Creating and loading seed table.

Table created.



# Build the Test Users

---

PL/SQL procedure successfully completed.

NOTIFY: Seed table loading procedure has exited.

NOTIFY: Setting up user 1 2 3 4 5 6 7 8

NOTIFY: Waiting for background processes - Tue Jun 25  
10:44:33 EDT 2013

NOTIFY: Setting up user 9 10

Table dropped.

NOTIFY: ./setup.sh: Loading procedure complete (47  
seconds). Please check ./cr\_tab\_and\_load.out for any  
errors

# Run the Test

---

**==> ./runit.sh <sessions>**

==> ./runit.sh 10

NOTIFY:

UPDATE\_PCT == 20

RUN\_TIME == 300

WORK\_LOOP == 0

SCALE == 10000

WORK\_UNIT == 256

ADMIN\_SQLNET\_SERVICE == ""

ADMIN\_CONNECT\_STRING == "/ as sysdba"

NON\_ADMIN\_CONNECT\_STRING == ""

SQLNET\_SERVICE\_MAX == "0"

NOTIFY: Testing SYSDBA connectivity to the instance to validate slob.conf settings.

NOTIFY: Testing non-SYSDBA connectivity to instance.

NOTIFY: Connectivity verified.

NOTIFY: Setting up trigger mechanism.

NOTIFY: Connecting users 1 2 3 4 5 6 7 8 9 10

NOTIFY: Pausing for 1 seconds before triggering the test.



# Run the Test

---

```
NOTIFY: Triggering the test.  
NOTIFY: Test triggered. Executing  
Tm 300  
./runit.sh: line 263: 26667 Killed          (  
iostat -xm 3 > iostat.out )  
./runit.sh: line 263: 26668 Killed          (  
vmstat 3 > vmstat.out )  
./runit.sh: line 263: 26669 Killed          (  
mpstat -P ALL 3 > mpstat.out )  
NOTIFY: SLOB test is complete.
```



# Examine the Results

```
==> cat awr.txt
```

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst Num	Release	RAC	Host
PARADISO	1730247835	Paradiso	1	10.2.0.4.0	NO	dayrheocrp03

	Snap Id	Snap Time	Sessions	Curs/Sess
Begin Snap:	54932	25-Jun-13 13:23:53	33	2.3
End Snap:	54933	25-Jun-13 13:28:54	22	2.8
Elapsed:		5.02 (mins)		
DB Time:		48.89 (mins)		

Cache Sizes

~~~~~

|                   | Begin  | End    |                 |         |
|-------------------|--------|--------|-----------------|---------|
| Buffer Cache:     | 1,120M | 1,120M | Std Block Size: | 16K     |
| Shared Pool Size: | 336M   | 336M   | Log Buffer:     | 47,112K |

# Examine the Results

---

## Load Profile

~~~~~

	Per Second	Per Transaction
	-----	-----
Redo size:	16,832,291.74	281,811,839.11
Logical reads:	17,138.81	286,943.67
Block changes:	7,244.08	121,282.78
Physical reads:	14,399.92	241,088.22
Physical writes:	3,636.11	60,876.89
User calls:	0.76	12.72
Parses:	2.83	47.44
Hard parses:	0.19	3.11
Sorts:	1.13	19.00
Logons:	0.00	0.06
Executes:	64.03	1,072.00
Transactions:	0.06	

# Examine the Results

---

```
==> cat awr.txt | egrep 'physical read total IO requests|  
physical write total IO requests|  
physical read total bytes|  
physical write total bytes|  
Elapsed:'
```

Statistic	Total	per Second	per Trans
physical read total IO requests	4,347,138	14,425.0	241,507.7
physical read total bytes	76,262,058,496	253,057,978.4	#####
physical write total IO requests	942,844	3,128.6	52,380.2
physical write total bytes	33,560,088,064	111,361,379.6	#####
Elapsed:	5.02 (mins)		

# Examine the Results

---

Physical IOPS =

$$\begin{aligned} & \text{(physical read total IO requests + physical write total IO requests )} \\ & \quad / \text{(Elapsed * 60 )} \\ = & (4,347,138 + 942,844) / (5.02 * 60) \\ = & 17,563.02 \end{aligned}$$

Physical MBPS =

$$\begin{aligned} & \text{(physical read total bytes + physical write total bytes )} \\ & \quad / \text{(Elapsed * 60 * 1024 * 1024)} \\ = & (76,262,058,496 + 33,560,088,064 ) / (5.02 * 60 * 1024 * 1024) \\ = & 347.72 \end{aligned}$$

# **./misc/awr\_info.sh**

---

**Parses the awr files into Pipe-delimited txt suitable for viewing in Excel.**

```
cp awr.txt history/awr.001
./misc/awr_info.sh history/*
```

```
FILE|SESSIONS|ELAPSED|EXECUTES|PREADS|READ_MBS|PWRITES|
WRITE_MBS|REDO_MBS|DFSR_LAT|DPR_LAT|DFPR_LAT|DFPW_LAT|
LFPW_LAT|TOP WAIT|
history/awr.001|001|301|64|14399|    241|3636|    106|16.0|
622|    0||    504|    1007|db file sequential read 4337754
2699 1 92.0 User I/O db file sequential read 4337754 .0
2699 1 ##### Log archive I/O 4913 .0 42 9 272.9|
```

# **./misc/awr\_info.sh**

---

**EXECUTES** – SQL executions per second.  
**PREADS** – Physical reads per second.  
**READ\_MBS** – Physical read throughput in megabytes per second.  
**PWRITES** – Physical writes per second.  
**WRITE\_MBS** – Physical write throughput in megabytes per second.  
**REDO\_MBS** – Redo write throughput in megabytes per second.  
**DFSR\_LAT** – Latencies in microseconds for db file sequential read wait events.  
**DPR\_LAT** – Latencies in microseconds for direct path read wait events.  
**DFPR\_LAT** – Latencies in microseconds for db file parallel read wait events.  
**DFPW\_LAT** – Latencies in microseconds for db file parallel write background wait events.  
**LFPW\_LAT** – Latencies in microseconds for log file parallel write background wait events.

# **Silly Little Oracle Benchmark**

## **Q & A**

**Troy Ligon**  
**tligon@soug.org**