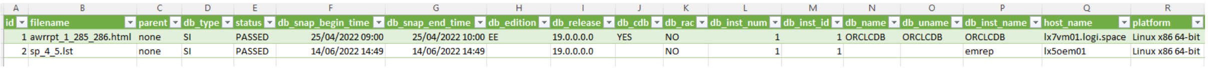
AWR documentation

The following abstract will give you an overview of the excel / csv files generated by the ATROPOSS AWR / Statspack module. The column name are based on the generated csv which is used for the dashboard in PowerBI. In the generated excel they might looks slightly different.

In general there is a good documentation available under - [AWR Reports (pafumi.net)]() for additional research.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
id	filename	parent	db_type	status	db_snap_begin_time	db_snap_end_time	db_edition	db_release	db_cdb	db_rac	db_inst_num	db_inst_id	db_name	db_uname	db_inst_name	host_name	platform
1	awrrpt_1_285_286.html	none	SI	PASSED	25/04/2022 09:00	25/04/2022 10:00	EE	19.0.0.0.0	YES	NO	1	1	ORCLCDB	ORCLCDB	ORCLCDB	lx7vm01.logi.space	Linux x86 64-bit
2	sp_4_5.lst	none	SI	PASSED	14/06/2022 14:49	14/06/2022 14:49		19.0.0.0.0		NO	1	1			emrep	lx5oem01	Linux x86 64-bit

Id
running sequence number

Filename
filename of the uploaded AWR / statspack file

Parent
reference name for RAC cluster and their appropriate instances

db_type
shows the type of deployed database.

- RAC (real application cluster / RAC One)
- RACI (real application cluster instance)
- SI (single instance)

status
shows if the AWR/Statspack lv.7 can be parsed or an error happened. Currently are not supported all level of statspack except level 7 and on the AWR side Diff – and pdb reports. The following values are supported (additional ones might follow in the future):

- Passed
- Failed (not in report)
- Unsupported (PDB level report)
- Failed

db_snap_begin_time
The start of the snapshot. AWR's display an average of the consumed resources of the underlying infrastructure. So, general spoken if the database is only busy few hours per week a snapshot over the week (7 days) might be useless but the d Depending on the workload different duration of the snapshot are useful.

- OLAP over a longer period of time
- OLTP over a short period of time

Tim Gorman has written a script "busiesest AWR" available here which is showing - [here]()

db_snap_end_time
The end of the snapshot (see point db_snap_begin_time)

db_edition

- What kind of database is parsed, like

- PO = Personal Ed.
- SE = Standard Ed.
- EE = Enterprise Ed.
- XE = eXpress Ed.
- CS = Standard Core Ed. (nur auf Exadata CC oder OCI cloud)
- CE = Enterprise Core Ed. (nur auf Exadata CC oder OCI cloud)
- HP = Enterprise Ed. High Perf. (nur auf Exadata CC oder OCI cloud)
- XP = Enteprise Ed. Extreme Perf (nur auf Exadata CC oder OCI cloud)

see - Licensing Information (oracle.com)

db_release

shows the database release like 8.1.x till 23.x. For databases where the db_release shows 19.0.0.0.0 an additional query on

- Software patch level like `$ORACLE_HOME/OPatch/opatch lspatches` or
- SQL patch level like select `description from dba_registry_sqlpatch order by action_time desc;`

Can be executed. It is always a good indicator if the IT department is able to follow the patch interval of Oracle for example.

db_cdb

Flag to show if the database is an oracle container managed database by YES or NO

db_rac

Flag to show if the database is a real application cluster database by YES or NO

db_inst_num

Flag to show the size of the real application cluster by number of rac instances.

db_name

name of the oracle database. For RAC database it shows the cluster name

db_uname

unique name of the oracle database. For RAC database it shows the unique cluster name

db_inst_name

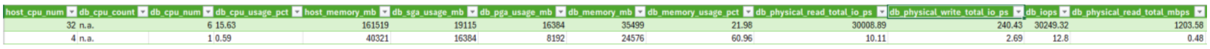
Instance name of the oracle database. For RAC it show the rac instance name.

host_name

show the host name of the deployed database. On virtualized environments the physical hostname and domain can be reached out via the available tool set of vmware for example.

Platform

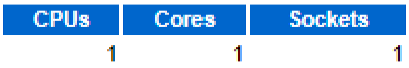
show the operating system of underlying infrastructure



host_cpu_num	db_cpu_count	db_cpu_num	db_cpu_usage_pct	host_memory_mb	db_sga_usage_mb	db_pga_usage_mb	db_memory_mb	db_memory_usage_pct	db_physical_read_total_io_ps	db_physical_write_total_io_ps	db_iops	db_physical_read_total_mbps
32	n.a.	6	15.63	161519	19115	16384	35499	21.98	30008.89	240.43	30249.32	1203.58
4	n.a.	1	0.59	40321	16384	8192	24576	60.96	10.11	2.69	12.8	0.48

host_cpu_num

shows the number of hosts cpus where the database is deployed on.



CPUs	Cores	Sockets
1	1	1

See - CPU CORES SOCKETS in AWR — oracle-mosc

- The "Cores" in the AWR report represents the amount of **physical** CPU cores you have in your host.
- The "CPUs" represents the thread count, i.e. taking into account the simultaneous multithreading (such as Intel's hyper-threading) architecture

Based on the available virtualization solutions where the database might be deployed on, check the number of physical cpus on the OS level like cat /proc/cpuinfo

db_cpu_count

shows if the cpu_count parameter is set during initialization. With 19c the cpu_min_count (currently missing in the report!) parameter is interesting as well – see here

• CPU_COUNT (on PDB level) – since Oracle 12.2
• CPU_MIN_COUNT (19c)

select name, value from v$parameter where name like 'cpu%count';

db_cpu_num

shows the consumed cpu's of the database during the AWR. Please consider again that the AWR's are displayed the average of the overall time period. The cpu is a calculation by including the foreground -, background time and cpu idle time, cpu busy time.

db_cpu_usage_pct

shows the percentage of consume database cpus to available host cpus

host_memory_mb

shows the available memory of the underlying host where the oracle database is deployed on. Please keep in mind that virtualized environments are able to move dynamically resources like memory and cpu between the databases.

db_sga_usage_mb

shows how much memory of the host is assigned to the SGA (share global area) of the database

db_pga_usage_mb

shows how much memory of the host is assigned to the PGA (program global area) of the database

db_memory_mb

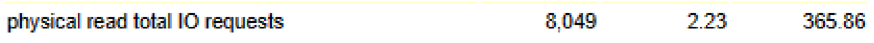
shows the total number of memory assigned to the oracle database

db_memory_usage_pct

shows the percentage of the overall host memory to the assigned memory of the oracle database

db_physical_read_total_io_ps

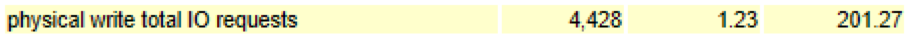
show how the database is used from an workload perspective. Here specific the physical read total IO requests per second (the number in the middle (2.23) of the subsequent snap) in the section "Instance Activity Stats".



physical read total IO requests	8,049	2.23	365.86

dp_physical_write_total_io_ps

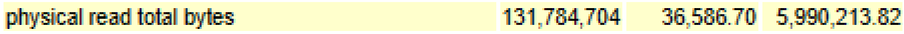
shows the equivalent of the previous total read IO request



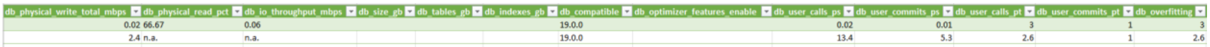
physical write total IO requests	4,428	1.23	201.27

db_iops shows the sum of total IO request from the read total IO requests and write total IO requests.

db_physical_read_total_mbps shows the physical read total bytes in mbps of the section “Instance Activity Stats” (the number in the middle of the subsequent snap).



physical read total bytes	131,784,704	36,586.70	5,990,213.82



db_physical_write_total_mbps	db_physical_read_pct	db_io_throughput_mbps	db_size_gb	db_tables_gb	db_indexes_gb	db_compatible	db_optimizer_features_enable	db_user_calls_ps	db_user_commits_ps	db_user_calls_pt	db_user_commits_pt	db_overfitting
0.02	66.67	0.06				19.0.0		0.02	0.01	3	1	3
2.4	n.a.	n.a.				19.0.0		13.4	5.3	2.6	1	2.6

db_physical_write_total_mbps shows the physical write total bytes in mbps of the section “Instance Activity Stats” (the number in the middle of the subsequent snap).

physical write total bytes	71,639,040	19,888.77	3,256,320.00

db_physical_read_pct shows the percentage of the read throughput

db_io_throughput_mbps shows the sum of the read total bytes and write total bytes = total throughput

db_size_gb Because data volume of the database is not available inside the database, a separate script can be downloaded from ATROPOSS and executed against the database. The output of a csv file can be uploaded together with the other information to enrich the overall information. The size is the raw data volume (net data + index data)

db_tables_gb Listed the biggest data volumes of the existing tables in the oracle database. The parameter is configurable and per default on 10%

db_indexes_gb shows the index data volume of the oracle database. Interesting for modernization purposes for example because the volume doesn’t need to be migrated.

db_compatible *Setting COMPATIBLE ensures that new features do not write data formats or structures to disk that are not compatible with the earlier release, preventing a future downgrade – more information here.*

db_optimizer_feature_enable Information is available here

db_user_calls_ps This metric represents the number of logins, parses, or execute calls per second during the sample period.

db_user_commits_ps This metric represents the number of user commits performed per second during the sample period. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.

db_user_calls_pt This metric represents the number of logins, parses, or execute calls per transaction during the sample period.

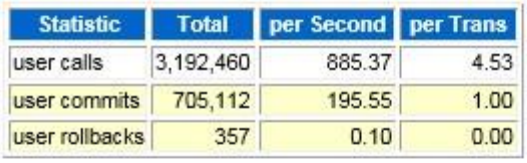
db_user_commits_pt

This metric represents the number of user commits performed per transaction during the sample period. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.

db_overfitting

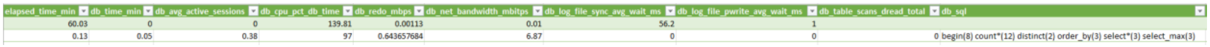
If the application commits to often, it can cause high waits on log file sync since each commit flushes redo data from the redo buffer to the redo logs. Oracle has a recommendation that user calls per (commits+rollbacks) should not be lower than 30, if it is the application is committing to frequent – see here.

In our case we have:



Statistic	Total	per Second	per Trans
user calls	3,192,460	885.37	4.53
user commits	705,112	195.55	1.00
user rollbacks	357	0.10	0.00

The formula are: user calls/(user commits+user rollbacks) which gives us
3.192.460/(705.112+357)=3.192.460/705.469=4.52 avg user calls per commit.
This is way below Oracles recommendations, ...



elapsed_time_min	db_time_min	db_avg_active_sessions	db_cpu_pct_db_time	db_redo_mbps	db_net_bandwidth_mbitps	db_log_file_sync_avg_wait_ms	db_log_file_pwrite_avg_wait_ms	db_table_scans_dread_total	db_sql
60.03	0	0	139.81	0.00113	0.01	56.2	1		
0.13	0.05	0.38	97	0.643657684	6.87	0	0	0	begin(8) count*(12) distinct(2) order_by(3) select*(3) select_max(3)

elapsed_time_min

Elapsed_time is the total time the query takes*. This includes the CPU time + waits (I/O, network, etc.). The elapsed time for a given execution should match the duration provided by SQL monitor. This is what you should pay most attention to. For example, if we generate AWR report for 1 hour then Elapsed Time in AWR report will be 60 mins.

db_time_min

The DB Time is a time model statistic that is the sum of all Oracle process' CPU consumption plus the sum of non-idle wait time. When optimizing Oracle databases, we focus on reducing the processing "time", usually by tuning SQL statements.

DB Time= CPU Time + Non IDLE wait time.

db_avg_active_sessions

Average Active Sessions (AAS) is the ratio of the change in Database Time (time in the database in seconds spent using CPU, IO, waiting) divided by the clock time. It is not a measure of how many sessions exist in the system at a given time, but rather how busy the database is

db_cpu_pct_db_time

db_redo_mbps For example, the load profile below shows that an average transaction generates about 18K of redo data, and the database produces about 1.8K redo per second. The above statistics give an idea about the workload the database experienced during the time observed- see here.

db_net_bandwidth_mbitps It's important to transfer and apply redo as fast as possible in a Data Guard environment. To achieve this we have to calculate the required network bandwidth based on the redo generation rate on the primary database.

Required bandwidth = ((Redo rate bytes per sec. / 0.75) * 8) / 1,000,000 = bandwidth in Mbps

Network bandwidth of redo in DG environments – here.

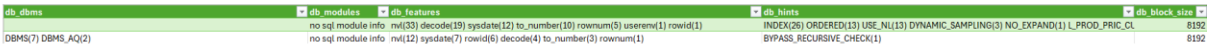
db_log_file_sync_avg_wait_ms see here

db_log_file_pwrite_avg_wait_ms The 'log file parallel write' event is caused by the Log writer (LGWR) process. The LGWR writes the redo buffer to the online redo log files . It issues a series of write calls to the system IO.

db_table_scans_dread_total It's critical to understand that a full-table scan is a symptom of a possible sub-optimal SQL plan. While not all full scans are evil to performance, full table scans are a symptom of other common tuning problems like missing indexes and sub-optimal schema statistics (dbms_stats). Remember, for small tables, a full-table scan is better than a full-scan, but a large-table full-table scan should always be examined as a "possible" problem.

db_sql shows in a aggregated way the number of dedicated sql statements.

begin(3) count*(3) distinct(5) order_by(17) select*(1) select_max(12)



db_dbms	db_modules	db_features	db_hints	db_block_size
	no sql module info	nvl(33) decode(19) sysdate(12) to_number(10) rownum(5) userenv(1) rowid(1)	INDEX(26) ORDERED(13) USE_NL(13) DYNAMIC_SAMPLING(3) NO_EXPAND(1) L_PROD_PRIC_CL	8192
DBMS(7) DBMS_AQ(2)	no sql module info	nvl(12) sysdate(7) rowid(6) decode(4) to_number(3) rownum(1)	BYPASS_RECURSIVE_CHECK(1)	8192

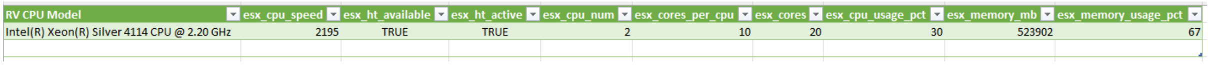
db_dbms shows what kind of DMBS modules are used and if the L&S or modernization makes sense. For example Oracle databases with DBMS_AQ has bigger modernization efforts because an alternative in the community Postgresql version is not available

db_modules shows if the application context is filled and if the migration is a technical or more a data migration focused one. So, an GIS application from ESRI can be identified and potential effort better calculated.

db_features shows with used sql functions of Oracle and give an indication about complexity and proprietary,

db_hints shows kinds of hints used in the database

db_block_size Typical values for DB_BLOCK_SIZE are 4096 and 8192. The value of this parameter must be a multiple of the physical block size at the device level.



RV CPU Model	esx_cpu_speed	esx_ht_available	esx_ht_active	esx_cpu_num	esx_cores_per_cpu	esx_cores	esx_cpu_usage_pct	esx_memory_mb	esx_memory_usage_pct
Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz	2195	TRUE	TRUE	2	10	20	30	523902	67

RV CPU Model shows kind of physical CPU model deployed in ESX environments for the database. A comparison with the available Azure CPU models caused into a better estimation of required licenses / sku's

esx_cpu_speed shows the clock speed of the physical deployed cpu model

esx_ht_available Hyperthreading available

esx_ht_active Hyperthreading active

esx_cpu_num number of assigned physical cpu's to the database

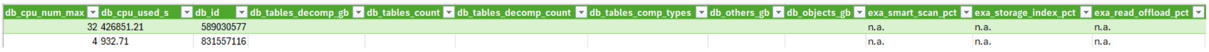
esx_cores_per_cpus number of cores per physical cpu

esx_cores number of physical cores

esx_cpu_usage_pct percentage of consumed cpu during the time interval of RVTool

esx_memory_mb number of assigned memory to the database

esx_memory_usage_pct number of used memory in percentag of the database



db_cpu_num_max	db_cpu_used_s	db_id	db_tables_decomp_gb	db_tables_count	db_tables_decomp_count	db_tables_comp_types	db_others_gb	db_objects_gb	exa_smart_scan_pct	exa_storage_index_pct	exa_read_offload_pct
32	426851.21	589030577							n.a.	n.a.	n.a.
4	932.71	831557116							n.a.	n.a.	n.a.

db_cpu_num_max max available cpu in gernal equal to host_cpu

db_cpu_used_s cpu seconds used by database

db_id The DBID is a very important part for Oracle databases. It is an internal, uniquely generated number that differentiates databases. Oracle creates this number automatically as soon as you create the database.

db_tables_decomp_gb size of compressed data in GB

db_tables_count Number of tables

db_tables_decomp_count Number of compressed tables

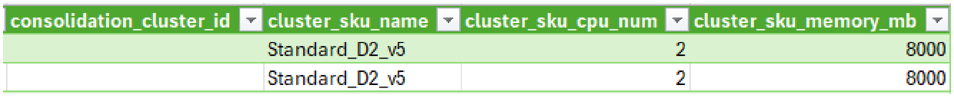
db_others_gb database net volume

db_objects_gb database index volume

exa_smart_scan_pct How many read operations can be benefit from smart scans

exa_storage_index_pct How many read operations are benefit from storage index

exa_read_offload_pct How many queries can be benefit from the Smart scan



consolidation_cluster_id	cluster_sku_name	cluster_sku_cpu_num	cluster_sku_memory_mb
	Standard_D2_v5	2	8000
	Standard_D2_v5	2	8000

consolidation_cluster_id based on a timeseries analysis the databases are consolidated based on the snapshot time period, the consumed CPU and consumed memory compared with the used sku at the beginning of the analysis.

cluster_sku_cpu_num Number of cpu’s of the chosen sku type

cluster_sku_memory_mb Number of memory of the chosen sku type