peakmarks® Benchmark Software

Technical Whitepaper

May 2022

contact@peakmarks.com

Introduction

The peakmarks® Benchmark Software enables the fast and reliable determination of generally understandable performance indicators for Database Services, both on-premises and in the cloud.

Performance indicators of Database Services can be used for various tasks.

Evaluation. Even for the well-known cloud services from Amazon, Google and Microsoft, no comparable and comprehensible performance indicators can be found. Considerations of the price-performance comparison of the various offerings are not possible without performance indicators. Performance indicators enable fact-based decisions.

Capacity planning. When migrating databases to new platforms, whether on-premises or in the cloud, performance indicators help with solid capacity planning.

License cost optimization. License costs often far exceed infrastructure costs. License costs can be significantly reduced by optimizing the infrastructure. The most suitable infrastructure components are quickly and clearly identified with the help of performance metrics.

Quality Assurance. Cloud services can change their infrastructure at any time and without notice. Performance analysis based on key figures enables regular performance reviews - quickly and reliably. Performance promises made by providers can thus be easily verified at periodic intervals.

The peakmarks® Benchmark Software was introduced to a broader benchmark audience at the 11th TPC Technology Conference 2019 in Los Angeles[1]. Please refer to the literature by Bermbach, Wittern, and Tai for more in-depth considerations on the systematic benchmarking of cloud services[2].

[1] Drozd: *Benchmarking Database Cloud Services*. In: Nambiar, R., Poess, M. (eds.) Performance Evaluation and Benchmarking for the Era of Cloud(s), Lecture Notes in Computer Science (LNCS), vol. 12257, pp. 139-153. Springer, Switzerland (2020).

[2] Bermbach, Wittern, Tai.: *Cloud Service Benchmarking – Measuring Quality of Cloud Services from a Client Perspective*. Springer International Publishing 2017.

Requirements for benchmark tools

Huppler[3] describes a good benchmark's five most important characteristics: relevant, repeatable, fair, traceable, and economical. peakmarks® meets all these requirements.

But other features are also critical to customer acceptance of benchmark tools.

Simplicity. It must be easy to install the benchmark software, run the benchmark and interpret the results. peakmarks® is implemented with database tools and without operating system scripts. Therefore, peakmarks® runs unchanged wherever the database software is available. Any database administrator can smoothly run the benchmark software without additional know-how.

Speed. The installation, the loading of the data, the processing of the different workloads, and the performance metrics evaluation should be fast. peakmarks® is installed in a few hours, including all adjustments to the database. The database loading time depends on the database size and the performance of the infrastructure. The scalable load process automatically adapts to the performance of the database platform. On powerful systems, load times of 4 TByte per hour are measured.

Complete benchmark runs with all workloads take between 12 and 24 hours; the results are available immediately. A comprehensive benchmark project can be completed within one week.

Understandable performance metrics. Many benchmarks provide only a single performance metric. This simplifies the comparison of different systems. However, a single complex metric is difficult to interpret[4]. peakmarks® delivers a set of representative and easy-to-understand metrics for various aspects. Concrete questions about performance can thus be answered more easily. Performance bottlenecks and malfunctions are detected more quickly.

Different load situations. Often, it is not the maximum value of a performance metric of interest but the optimal performance range where a sustained and predictable performance output occurs. peakmarks® analyzes the performance of a database service in different load situations. Workloads start with a low load. The load is continuously increased until system saturation. In this way, the optimal performance range can be determined.

Continuous further development. In contrast to many open-source benchmark tools, the peakmarks® Benchmark Software is continuously developed. The requirements of new hardware technologies such as *flash storage* and *persistent memory* are incorporated into further development, as are new database technologies (*smart scan*, *in-memory column store*, *memory-optimized tables,* etc.). The list of workloads is supplemented as this is relevant for measuring the infrastructure and the solution architecture choice.

[3] Huppler: *The Art of Building a Good Benchmark*. In: Nambiar, R., Poess, M. (eds.) Performance Evaluation and Benchmarking, LNCS, vol. 5895, pp. 18-30. Springer, Heidelberg (2009).

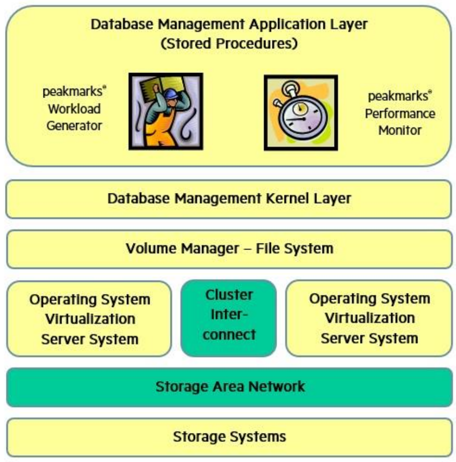
[4] Crolotte: *Issues in Benchmark Metric Selection*. In: Nambiar, R., Poess, M. (eds.) Performance Evaluation and Benchmarking 2009, LNCS, vol. 5895, pp. 146-152. Springer, Heidelberg (2009).

The architecture of the peakmarks® Benchmark Software

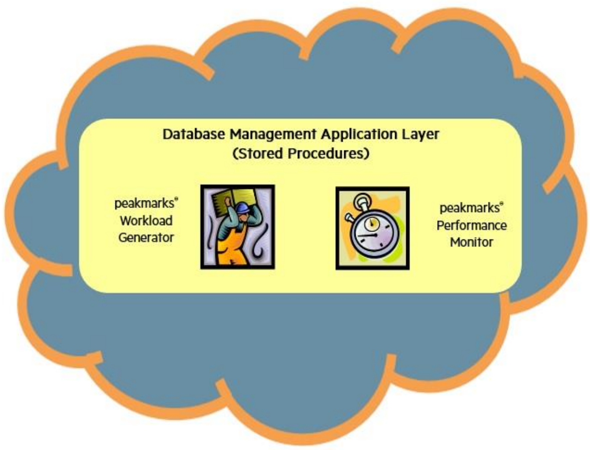
The peakmarks® Benchmark Software is stored in the database in the form of stored procedures. Tools such as Putty are required to access the database server. Tools based on SCP transfer the benchmark software and exchange the results.

The peakmarks® Benchmark Software consists of two components. A workload generator creates the database load, so-called workloads. All workloads are generated within the database and executed by database jobs.

A performance monitor collects all relevant statistics before and after each performance test and immediately displays the result in tabular form. The performance is measured at the interface to the application, i.e., the measured performance is directly available to the application.



peakmarks® Benchmark Software
on-premises



peakmarks® Benchmark Software
in the Cloud

Via various peakmarks® configuration parameters, the benchmark database, which is generated synthetically, can be adapted to customer requirements:

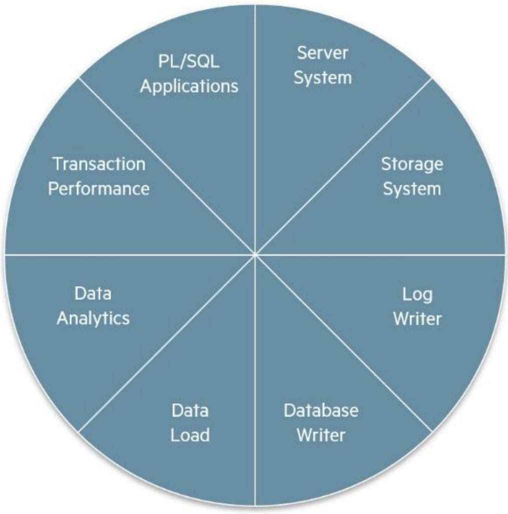
- Database size between 50 GByte and 64 TByte per database instance.
- Record length of the benchmark tables between 40 and 4000 bytes.
- Redundancy of data to test deduplication procedures of storage systems.
- Optional use of Database Flash Cache for conventional database servers and Cell Flash Cache for Oracle engineered systems.
- Optional use of encryption methods. The essential encryption methods are supported.

The runtime of each performance test can be set between 60 seconds and 72 hours. Optionally, replication technologies such as Oracle Data Guard can be configured. The Oracle Real Application Cluster technology with multiple database servers is also supported.

Full 360-degree performance overview

With over 30 workloads, so-called micro-benchmarks, in 8 workload groups, the peakmarks® Benchmark Software provides a representative and complete overview of a Database Service performance in all load situations.

The two most essential components of a platform - server system and storage system - have a decisive influence on the performance of all database operations and often determine license and maintenance costs. Therefore, one group of workloads analyzes the performance behavior of a server system in database operations. Another workload group determines the performance behavior of a storage system in database operation.



Two service processes are of great importance for the smooth operation of the database. Log writer processes are responsible for transaction management, and database writer processes for buffer management. The peakmarks® Benchmark Software examines the performance of these two crucial database service processes with specially developed workloads.

Performance indicators for representative database operations such as data load, data analytics, and transaction processing are necessary for capacity planning. Different technologies can be used for this.

The data load workloads determine the throughput for loading data using different methods such as *Buffered Load*, *Direct Load,* and *Stream Load* for IoT applications.

The data analytics workloads explore various technologies' performance to accelerate searching non-indexed data, such as *Smart Scan* technology on Oracle Engineered Systems or *in-memory* technology available on many platforms.

In transaction processing, workloads determine transaction throughput and response time behavior for transactions of varying complexity.

Some applications encapsulate essential business functions and transactions in stored procedures. The peakmarks® Benchmark Software offers workloads to analyze the PL/SQL code's performance behavior on different processors.

All essential key performance indicators are presented in peakmarks® own reports and summarized in a 2-page performance certificate. Oracle AWR reports are automatically generated for a detailed analysis of all performance tests.

Configuration of peakmarks® Performance Tests

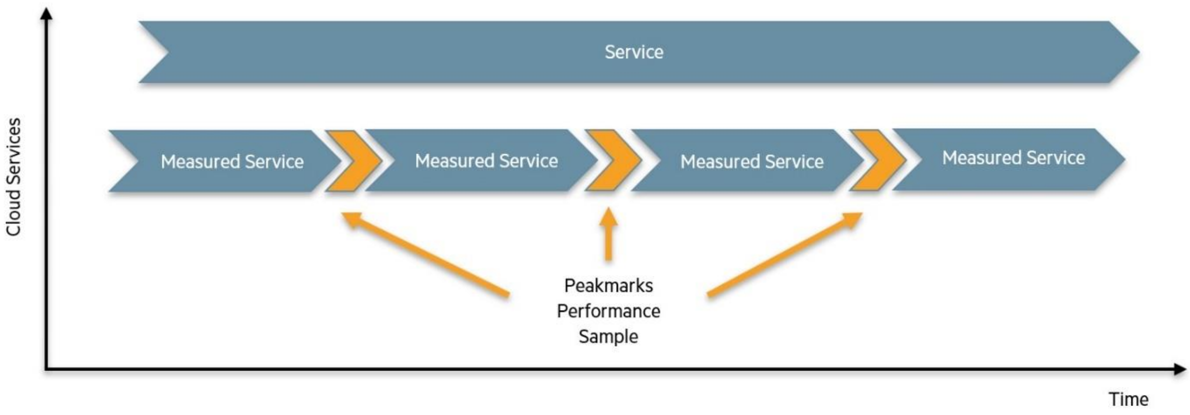
For the configuration of a single performance test, the peakmarks® Benchmark Software needs the following information:

- Workload to be performed.
- For some workloads, there are optional workload parameters.
- Internal parallelism can be selected for some workloads.
- Number of cluster nodes on which the workload is to be executed.
- Number of load processes that execute the workload. The processes are distributed round-robin evenly across all cluster nodes in a cluster.
- Target runtime for the performance test.

Workloads must be configured for different load situations to measure a database service ultimately. This requires a sequence of performance tests that can be configured in different ways.

Smart Benchmark Configuration. *Smart Benchmark Configuration* is fully automated. It is the most convenient and fastest way to get a Database Service performance overview. For each workload, a sequence of performance tests is generated automatically. Should the system reach saturation as the load increases, the test sequence will not continue.

Sample Benchmark Configuration. The *Sample Benchmark Configuration* also runs entirely automatically but only performs a selected set of preconfigured performance tests. This type of benchmark configuration is often used with cloud services to periodically check within a maintenance window of 60 minutes whether the database service still fully meets its performance characteristics.



Manual Benchmark Configuration. The *Manual Benchmark Configuration* offers the highest flexibility for the configuration of benchmark tests. Each parameter of a performance test can be selected individually. Engineers prefer this procedure to analyze special load situations in detail.

Structure of the peakmarks® Benchmark Reports

Units of measurement and abbreviations

The following units of measurement and abbreviations are used in the benchmark reports:

Abbreviation	Meaning	Abbreviation	Meaning
[MBps]	megabyte per second	[s]	seconds
[GBps]	gigabyte per second	[ms]	milliseconds
[TBph]	terabyte per hour	[µs]	microseconds
[rps]	rows per second	[dbps]	database blocks per second
[qps]	queries per second	[rbps]	redo blocks per second
[tps]	transactions per second	[kBpt]	kiloByte per transaction
[IOPS]	I/O operations per second	[Mops]	Million operations per seconds
BuCache	Database buffer cache	FlCache	Flash Cache (Database, Exadata)

Examples

The following examples show two tabular peakmarks® benchmark reports[5].

The first example shows the performance behavior of the system when running the SRV-QUERY25 workload. In the 2-node cluster, each server has 36 cores and 72 threads.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Queries total [qps]	Queries per cpu [qps]	Response time [ms]	Log reads total [dbps]	Log reads per cpu [dbps]	BuCache read [%]	Elapsed time [s]
1	1	SRV-QUERY25	2	2	3	2	1	97	39,821	19,911	0.050	1,258,276	629,138	99.95	301
	2	SRV-QUERY25	2	36	26	25	1	74	546,839	15,190	0.066	14,908,531	414,126	100.00	301
	3	SRV-QUERY25	2	72	51	50	1	49	975,648	13,551	0.074	26,373,323	366,296	100.00	301
	4	SRV-QUERY25	2	108	76	75	1	24	1,075,852	9,962	0.100	29,036,208	268,854	100.00	301
	5	SRV-QUERY25	2	144	97	95	1	3	1,120,775	7,783	0.128	30,210,434	209,795	100.00	301

The second example shows the performance behavior of the system when running the STO-RANDOM workload.

Run	Test	Workload	Wri [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Phys reads total [dbps]	Phys reads total [IOPS]	IO time read [ms]	Phys reads total [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
1	6	STO-RANDOM	0	2	2	3	2	1	97	0	57,730	56,309	0.260	451	88.39	100.00	182
	7	STO-RANDOM	0	2	18	15	10	4	85	0	438,618	436,789	0.304	3,427	50.45	100.00	182
	8	STO-RANDOM	0	2	36	23	15	6	77	0	591,892	590,781	0.398	4,624	43.95	100.00	181
	9	STO-RANDOM	0	2	54	28	18	7	72	0	819,169	818,538	0.812	6,400	36.19	100.00	182
	10	STO-RANDOM	0	2	72	32	20	8	68	0	866,014	865,802	1.161	6,766	35.34	100.00	181
	11	STO-RANDOM	0	2	90	35	22	9	65	0	900,986	900,717	1.591	7,039	34.34	100.00	181

[5] All the following benchmark reports were determined on the current (as of April 2022) peakmarks® reference system: an Exadata X5-2 Quarter Rack (release date 2015) with 2 database servers and 3 high capacity storage servers. The database servers are each equipped with 2 Intel Xeon E5-2699 v3 processors (2.3 - 3.6 GHz) and 768 GByte main memory. Each of the three intelligent high capacity storage servers has a HDD capacity of 48 TByte and a flash cache of 6.4 TByte. The database and storage servers are interconnected by an internal InfiniBand network (40 Gbps).

Description of the performance tests

The first column of each benchmark report identifies the benchmark run (column *Run*). Within a benchmark run, all performance tests are numbered (column *Test*). The third column describes the workload. Optionally, additional workload parameters depend on the workload. In the second example, this is the percentage of write operations (column *Wri [%]*).

The *Nodes* column indicates how many cluster nodes are involved in a test. The *Jobs* column documents the number of processes on all cluster nodes involved that generate the load. The processes are distributed round-robin to all cluster nodes for tests in clusters.

An additional column DOP (degree of parallelism), describes the internal database parallelization for some workloads.

Choice of measurement points for CPU-bound performance tests

For *CPU-bound* workloads like SRV-QUERY25, the peakmarks® Benchmark Software automatically selects five measurement points to determine the server's performance in different load situations: Single-thread, 25% CPU utilization, 50% CPU utilization, 75% CPU utilization, 100% CPU utilization.

Choice of measurement points for other workloads

For other workloads, the peakmarks® Benchmark Software selects up to thirty-two measurement points. Starting with one process per server, the load is automatically increased until system saturation occurs. The load increase is determined automatically depending on the available threads and can be overridden manually.

In the second example with the STO-RANDOM workload, the measurement series is aborted after the sixth test because there is no longer a significant performance increase. The system is saturated.

CPU utilization

The following four columns describe the CPU utilization of all involved servers. For workloads with a high I/O share, an additional column represents the percentage of CPU I/O wait (*CPU iow [%]* column).

Key performance indicators

Now columns with the key performance indicators follow. In the first example, the number of queries (column *Queries total [qps]*), the average response time of the queries (column *Response time [ms]*), and the number of logical database block access in the database buffer cache (column *Log reads total [dbps]*).

In the second example, the two key performance indicators are the number of I/O operations (column *Phys reads total [IOPS]*) and the I/O service time (column *IO Time read [ms]*).

The performance indicators often contain additional columns that show the performance per CPU used[6]. These key figures are essential for price-performance comparisons to determine the performance per core. License and maintenance costs are usually based on the number of sockets used and cores.

Other key figures

Columns with additional key figures follow. These key figures often provide additional information to understand the system behavior better or check whether the workload has also run optimally.

In our first example, these are the database buffer cache hit rate (*BuCache read [%]* column) and the workload processing time (*Elapsed time [s]* column). In the second example, we also find the Flash Cache Hitrate (column *FlCache read [%]*).

Workloads for server systems

peakmarks® offers various workloads for server systems. The selected server workloads occur across all industries in all database applications. They show the *real-world* performance of a server system in database operation.

All server workloads access tables via SQL with different access patterns. The affected tables are entirely in the database buffer cache. There are almost no I/O operations, so these workloads' performance is completely limited by CPU power.

Server Performance Tests answer questions about servers' scalability and the efficiency of technologies such as multithreading, virtualization, and data encryption. For on-premises platforms, they enable a simple price-performance comparison of processors and servers, taking licensing into account. They also provide comparable metrics on a cloud provider's price-performance ratio when evaluating cloud platforms, where the hardware components are not always known.

Performance indicators for server systems in database operation

To measure the performance of server systems in database operation, the following performance indicators have proven to be helpful:

- SQL query throughput in *queries per second* [qps].
- Average SQL *query response time* in *milliseconds* [ms].
- The number of logical database block accesses in the buffer cache (*logical reads throughput*) in *database blocks per second* [dbps].
- SQL Buffer Cache Scan Rate in *MegaByte per second* [MBps].

[6] For processors without multithreading, one CPU corresponds to one core; for systems with multithreading, one CPU corresponds to one thread.

Workloads to determine the server performance in database operation

The following table shows the different workloads used to assess server performance in database operations.

Workload	Action	Performance Indicator	Unit of measure
SRV-QUERY1	Query type 1: Select 1 record via index. Example: Select account, product, order, invoice, etc. This workload shows the maximum query throughput and the minimum query response time.	Query throughput Query response time	[qps] [ms]
SRV-MEMOPT	Same workload as SRV-QUERY1. Uses the *memory-optimized row store for reading*.	Query throughput Query response time	[qps] [ms]
SRV-QUERY25	Query type 2: Select an average of 25 records via index. Example: Select account postings last week; item list of order, etc.	Query throughput Query response time	[qps] [ms]
SRV-REPORT	Online Report: Select an average of 125 records via index. This workload corresponds to a simple online report and shows the maximum value for *a server's logical reads*. Example: Online report mobile phone call records from last month, online report e-banking with account mutations from the previous month, etc.	Block access in the buffer cache (logical reads)	[dbps]
SRV-SCAN	Data search without index. This corresponds to a *full table scan*. Only one record is determined, but the entire data set in the buffer cache must be searched.	Scan rate in buffer cache	[MBps]
SRV-MIXED	A complex workload with a mix of equally-weighted simple workloads running concurrently: ▪ SRV-QUERY1 ▪ SRV-QUERY25 ▪ SRV-REPORT ▪ SRV-SCAN.	Query throughput Query response time	[qps] [ms]

The respective key performance metrics columns are marked in red in the following examples.

Example: Workload SRV-QUERY1

The SRV-QUERY1 workload (1 hit per query) shows the maximum possible number of queries and the shortest query response time for a server system.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Queries total [qps]	Queries per cpu [qps]	Response time [ms]	Log reads total [dbps]	Log reads per cpu [dbps]	BuCache read [%]	Elapsed time [s]
2	1	SRV-QUERY1	2	2	3	2	1	97	112,149	56,075	0.018	519,738	259,869	100.00	301
	2	SRV-QUERY1	2	36	26	25	1	74	1,677,037	46,584	0.021	5,214,149	144,837	100.00	301
	3	SRV-QUERY1	2	72	51	50	1	49	2,965,689	41,190	0.024	9,075,311	126,046	100.00	301
	4	SRV-QUERY1	2	108	76	75	1	24	3,219,260	29,808	0.033	9,831,671	91,034	100.00	301
	5	SRV-QUERY1	2	144	97	96	1	3	3,224,884	22,395	0.045	9,842,566	68,351	100.00	301

Example: Workload SRV-QUERY25

The SRV-QUERY25 workload (25 records per query) shows the performance behavior of the server system for somewhat more complex queries, where a balanced relationship between latency and throughput is required.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Queries total [qps]	Queries per cpu [qps]	Response time [ms]	Log reads total [dbps]	Log reads per cpu [dbps]	BuCache read [%]	Elapsed time [s]
2	6	SRV-QUERY25	2	2	3	2	1	97	39,821	19,911	0.050	1,258,276	629,138	99.95	301
	7	SRV-QUERY25	2	36	26	25	1	74	546,839	15,190	0.066	14,908,531	414,126	100.00	301
	8	SRV-QUERY25	2	72	51	50	1	49	975,648	13,551	0.074	26,373,323	366,296	100.00	301
	9	SRV-QUERY25	2	108	76	75	1	24	1,075,852	9,962	0.100	29,036,208	268,854	100.00	301
	10	SRV-QUERY25	2	144	97	95	1	3	1,120,775	7,783	0.128	30,210,434	209,795	100.00	301

Example: Workload SRV-REPORT

The workload SRV-REPORT (online report with an average of 125 records per query) shows the maximum number of logical database block accesses for a database server.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Queries total [qps]	Queries per cpu [qps]	Response time [ms]	Log reads total [dbps]	Log reads per cpu [dbps]	BuCache read [%]	Elapsed time [s]
2	11	SRV-REPORT	2	2	3	2	1	97	11,500	5,750	0.174	1,637,776	818,888	99.97	300
	12	SRV-REPORT	2	36	26	26	1	74	156,327	4,342	0.230	19,851,695	551,436	100.00	301
	13	SRV-REPORT	2	72	51	50	1	49	263,023	3,653	0.273	33,129,757	460,136	100.00	301
	14	SRV-REPORT	2	108	77	75	1	23	308,636	2,858	0.349	38,750,436	358,800	100.00	301
	15	SRV-REPORT	2	144	98	97	1	2	331,726	2,304	0.433	41,556,035	288,584	100.00	301

Example: Workload SRV-SCAN

The workload SRV-SCAN (query with a *full table scan*, where one hit is determined) shows the buffer cache's scan rate.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Scan rate total [MBps]	Scan rate per cpu [MBps]	Log reads total [dbps]	Log reads per cpu [dbps]	BuCache read [%]	Elapsed time [s]
2	16	SRV-SCAN	2	2	2	2	1	98	4,118	2,059	527,088	263,544	99.99	181
	17	SRV-SCAN	2	36	26	25	1	74	62,428	1,734	7,990,833	221,968	100.00	181
	18	SRV-SCAN	2	72	50	49	1	50	107,269	1,490	13,730,415	190,700	100.00	181
	19	SRV-SCAN	2	108	76	74	1	24	130,768	1,211	16,738,291	154,984	100.00	181
	20	SRV-SCAN	2	144	95	94	1	5	140,471	975	17,980,230	124,863	100.00	181

Notes

Main memory accesses[7] enable ultimate database performance. Extremely short response times between 18 and 45 µs are determined for the SRV-QUERY1 workload.

Misleading CPU utilization. If Intel Xeon processors are operated with multithreading, the CPU utilization information is not always meaningful. In the SQL-QUERY1 workload, all cores are occupied by processes in test 3. The CPU utilization is 51%, and 2,965,689 queries are processed per second. In test 5, all threads are occupied by processes, CPU utilization increases to 97%, and 3,224,884 queries are processed per second. The server is now fully utilized. The CPU utilization almost doubled between the two tests, but the throughput only increased by just under 9%. I.e., already in test 3, the actual CPU utilization was 92% and not 51%!

Many operating systems rate threads as resources when calculating CPU utilization, just like cores. But they are not. Threads are merely a concept for improving the parallel processing of instructions.

Intel Xeon processors' multithreading can lead to a throughput improvement of up to 25% for Oracle workloads. Other values apply for processor architectures from other vendors such as AMD EPYC, IBM POWER, or IBM z.

Scalability with increasing load. We observe a decreasing performance per CPU with increasing load for each server workload. For example, the scan rate for the SRV-SCAN workload drops from 2,059 MBps (test 16, single-thread performance) to 975 MBps (test 20, system saturation). With increasing load, the performance per thread drops to about 50% of the peak value.

We observe this behavior, especially in Intel Xeon processors with a large spread of the clock rate. In our case (Intel Xeon E5-2699 v3), the maximum clock rate is 3.6 GHz. When the load increases, the clock rate is reduced to 2.3 GHz for thermal reasons. This means a reduction in power output of 36%.

As a rule, Intel Xeon processors with a high number of cores also have a high clock rate spread. And vice versa, Intel Xeon processors with a lower number of cores also have a lower clock rate spread and can thus guarantee a more stable performance output.

Server evaluation for database operation. Intel Xeon processors with fewer cores are recommended to achieve predictable and persistent CPU performance in all load ranges.

Such processors also deliver the highest per-core performance and improve the price-performance ratio, considering licensing and maintenance costs.

More sockets may then be necessary to achieve scalability requirements. This also leads to the ability to use significantly higher main memory capacities, further improving performance and promoting *in-memory* technologies.

[7] For orientation: the fastest processors (as of March 2021) require less than 0.1 nanoseconds for a DRAM access. The database needs about 1 microsecond for a block access in the database buffer cache and between 50 and 500 microseconds (depending on the storage fabric and protocol) for a block access on flash storage.

Workloads for storage systems

Conventional I/O benchmark tools such as fio, vdbench, iometer, and Orion often show performance values not achieved in real database operations. The reason for this is the complexity of database I/O operations, which is not considered by these tools.

When a data block is read, the buffer cache management of the database must perform many tasks:

- find a free place for the block.
- If there is no free space, replace older blocks.
- Synchronize all database processes, simultaneously trying to occupy free spaces in the buffer cache.
- When using a shared disk cluster architecture (Oracle Real Application Cluster), synchronization must be performed cluster-wide, which requires exchanging information (messages) between cluster nodes; in the worst case, the exchange of database blocks must be performed additionally.
- Finally, blocks are checked for integrity and consistency during I/O transfer. Configuration parameters can often define the scope of integrity and consistency tests. These tests can lead to an additional load on processors and the I/O system.

The peakmarks® Benchmark Software generates I/O load with so-called SQL-generated I/O operations to determine representative performance metrics for the storage system in database operations. For sequential operations, the intra-SQL parallelism can additionally be tested to find the optimal SQL parallelism.

Storage performance tests provide answers to questions about the efficiency of protocols such as NVMe *over fabrics* (IP, FC), RoCE (*RDMA over Converged Ethernet*), and technologies such as flash storage, persistent memory, storage tiering, offload functions of intelligent storage systems and data reduction through deduplication and compression. They provide well-founded key figures for price-performance comparisons for storage systems for databases and the evaluation of suitable storage replication methods.

Performance indicators for storage systems in database operation

The following performance indicators are used to measure the performance of storage systems in database operation:

- *Sequential read* throughput of storage systems in *megabytes per second* [MBps]. Large I/O units of 1 MByte are typically used.
- *Random* access of storage systems (*random I/O*) in *I/O operations per second* [IOPS]. Typically, smaller I/O units like database blocks are used. The size of database blocks is configurable. The default value is often 8 KByte.
- I/O service time during random access in *milliseconds* [ms].

Workloads for determining storage performance in database operation

The following table shows the available workloads of the peakmarks® Benchmark Software for measuring the performance of storage systems in database operation.

Workload	Action	Performance Indicator	Unit of measure
STO-READ	Sequential read operation generated by SQL statements.	Throughput *sequential* I/O	[MBps]
STO-OFFLOAD	Sequential read operation generated by SQL statements using intelligent storage servers with offload functionality (*smart scan*).	Throughput *sequential* I/O	[MBps]
STO-RANDOM	Random read and write operations generated by SQL statements. In this workload, read operations are performed by foreground processes, and write operations are performed by background processes. The ratio of read and write operations can be configured.	Throughput *random* I/O I/O service time	[IOPS] [ms]
STO-SCATTER	Random write operations generated by SQL statements. In this workload, write operations are performed by foreground processes.	Throughput *random* I/O	[dbps]

Example: Workload STO-READ

The workload STO-READ shows the I/O throughput between the database server and storage system during sequential read operations. In this example, an DOP of 4 is used. The entire data can be loaded from the flash cache (*FlCache read* column).

Run	Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Phys reads total [dbps]	Phys reads total [IOPS]	Phys reads total [MBps]	FlCache read [%]	Elapsed time [s]
3	1	STO-READ	2	2	4	2	2	1	98	0	848,265	6,698	6,627	100.00	174
	2	STO-READ	2	8	4	3	2	1	97	0	1,344,795	10,580	10,506	100.00	184
	3	STO-READ	2	16	4	4	3	1	96	0	1,397,957	10,997	10,922	100.00	189
	4	STO-READ	2	24	4	4	3	1	96	0	1,410,993	11,122	11,023	100.00	204
	5	STO-READ	2	32	4	4	3	1	96	0	1,424,512	11,203	11,129	100.00	211

Example: Workload STO-OFFLOAD

The STO-OFFLOAD workload is the same as STO-READ but leverages the storage systems' intelligence and uses *Smart Scan* technology. When using *Smart Scan* technology, it is not necessary to specify a DOP because the storage servers optimize access.

Run	Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Phys reads total [dbps]	Phys reads total [IOPS]	Phys reads total [MBps]	FlCache read [%]	Elapsed time [s]
3	6	STO-OFFLOAD	2	2	1	2	1	1	98	0	3,674,072	28,930	28,704	100.00	286
	7	STO-OFFLOAD	2	8	1	2	1	1	98	0	3,802,718	29,926	29,709	100.00	287
	8	STO-OFFLOAD	2	16	1	2	1	1	98	0	3,902,466	30,703	30,488	100.00	289
	9	STO-OFFLOAD	2	24	1	2	1	1	98	0	3,871,813	30,474	30,249	100.00	291
	10	STO-OFFLOAD	2	32	1	2	1	1	98	0	3,881,741	30,539	30,326	100.00	292

Example: Workload STO-RANDOM

The workload STO-RANDOM shows the I/O throughput between the database server and storage system with *random* access to individual database blocks. In this example, the size of the database blocks is 8 KByte.

Only read operations are measured in the first benchmark report (tests 11 to 16). In the second benchmark report (tests 17 - 25), the ratio of read to write operations is 80:20 (column *Wri [%]*).

Run	Test	Workload	Wri [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Phys reads total [dbps]	Phys reads total [IOPS]	IO time read [ms]	Phys reads total [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
3	11	STO-RANDOM	0	2	2	3	1	1	97	0	63,489	62,190	0.270	496	0.00	100.00	302
	12	STO-RANDOM	0	2	12	10	6	3	90	0	348,349	347,755	0.335	2,721	1.44	100.00	302
	13	STO-RANDOM	0	2	24	16	10	4	84	0	529,676	529,643	0.447	4,138	1.75	100.00	302
	14	STO-RANDOM	0	2	36	22	14	6	78	0	626,316	626,257	0.579	4,893	2.09	100.00	302
	15	STO-RANDOM	0	2	48	23	14	6	77	0	764,040	763,963	0.836	5,969	2.33	100.00	302
	16	STO-RANDOM	0	2	60	27	17	7	73	0	811,227	811,199	1.044	6,338	2.71	100.00	301
	13	STO-RANDOM	0	2	72	31	19	8	69	0	844,894	844,876	1.285	6,601	2.98	100.00	301
	14	STO-RANDOM	0	2	84	33	20	9	67	0	875,005	874,974	1.558	6,836	3.05	100.00	301
	15	STO-RANDOM	0	2	96	34	21	9	66	0	894,074	894,047	1.850	6,985	3.09	100.00	301

Run	Test	Workload	Wri [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Phys reads total [dbps]	Phys reads total [IOPS]	IO time read [ms]	Phys reads total [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
3	17	STO-RANDOM	20	2	2	2	1	1	98	0	57,779	57,744	0.271	451	16.61	99.95	301
	18	STO-RANDOM	20	2	12	8	5	2	92	0	253,920	253,885	0.374	1,984	17.25	99.94	302
	19	STO-RANDOM	20	2	24	14	9	3	86	0	395,169	395,134	0.525	3,087	18.07	99.94	302
	20	STO-RANDOM	20	2	36	19	13	4	81	0	490,533	490,494	0.673	3,832	18.63	99.94	302
	21	STO-RANDOM	20	2	48	23	16	5	77	0	558,467	558,427	0.850	4,363	19.04	99.93	301
	22	STO-RANDOM	20	2	60	27	18	6	73	0	603,210	603,179	1.033	4,713	19.34	99.93	302
	23	STO-RANDOM	20	2	72	30	20	7	70	0	637,117	637,076	1.241	4,977	19.62	99.93	302
	24	STO-RANDOM	20	2	84	32	22	7	68	0	664,766	664,728	1.445	5,193	19.91	99.93	302
	25	STO-RANDOM	20	2	96	34	23	8	66	0	682,108	682,067	1.658	5,329	20.17	99.93	301

Notes

I/O bandwidth. With workload STO-READ, the amount of data transferred is usually limited by the I/O bandwidth between the server and storage systems.

I/O service time. With the workload STO-RANDOM, the I/O service time is influenced by the efficiency of the I/O stack. Service times of less than 500 microseconds are expected when using all-flash arrays. Thanks to new protocols such as NVMe-oF (NVMe *over Fabrics*), I/O service times of less than 100 microseconds are now possible.

Log Writer Workloads

The Log Writer processes are primarily responsible for transaction logging and database recovery from system failures. These processes are critical to the database system's overall performance when processing transactions. The latency of transaction logging can significantly impact the response time of user transactions.

Optionally, Log Writer processes are also used for database replication to synchronize standby databases. This technology is very popular for disaster recovery solutions. Replication can be performed in synchronous or asynchronous mode. Data transfer between the primary and standby databases can optionally be both encrypted and compressed. With synchronous replication, local transactions must wait until the standby databases have received the transaction log. This can significantly delay local transaction processing.

Performance metrics for Log Writer processes

The performance indicators for Log Writer processes are:

- Number of SQL Commit operations in *transactions per second* [tps].
- Average latency for SQL commit operations in *milliseconds* [ms].
- REDO throughput in *MegaByte per second* [MBps].

Workloads for Log Writer processes

To analyze the Log Writer processes performance for different transaction sizes, the peakmarks® Benchmark Software offers different workloads.

Workload	Action	Performance Indicator	Unit of measure
LGWR-LAT1 LGWT-LAT25 LGWR-LAT125	Transactions with 1, 25, or 125 records per transaction and *COMMIT WAIT*. The difference between these three workloads is the transaction size or how much REDO data is generated per transaction.	SQL Commit operations SQL Commit latency	[tps] [ms]
LGWR-THR	A transaction that generates an extensive REDO data set and terminates with *COMMIT WAIT.*	Log Writer throughput	[MBps]

Example: Workload LGWR-LAT1

The workload LGWR-LAT1 shows the maximum number of commit operations that the system can process. The average transaction size, measured by the REDO data volume, is under 2 KByte (column *REDO data [kBpt]*).

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Commit throughput [tps]	Commit latency [ms]	REDO blocks [rbps]	REDO writes [IOPS]	REDO syn writes [IOPS]	REDO data [MBps]	REDO data [kBpt]	Log file syn [ms]	FlCache write [%]
4	1	LGWR-LAT1	2	2	3	2	1	97	0	3,808	0.521	16,590	3,832	3,809	7	1.88	0.346	93.87
	2	LGWR-LAT1	2	18	8	6	2	92	0	31,705	0.564	140,050	11,384	31,706	60	1.94	0.389	93.13
	3	LGWR-LAT1	2	36	11	9	2	89	0	54,778	0.652	245,651	10,045	54,780	104	1.94	0.472	93.60
	4	LGWR-LAT1	2	54	15	12	2	85	0	73,404	0.732	331,070	9,216	73,408	140	1.95	0.564	91.95
	5	LGWR-LAT1	2	72	18	15	2	82	0	89,709	0.794	399,881	8,535	89,714	170	1.94	0.635	89.90
	6	LGWR-LAT1	2	90	21	18	3	79	0	103,864	0.857	457,147	7,870	103,870	197	1.94	0.708	87.42
	7	LGWR-LAT1	2	108	23	20	3	77	0	114,516	0.935	497,967	7,239	114,523	217	1.94	0.797	85.02
	8	LGWR-LAT1	2	126	26	23	3	74	0	123,835	1.009	531,025	6,648	123,844	234	1.93	0.872	81.49
	9	LGWR-LAT1	2	144	26	23	3	74	0	127,302	1.121	536,050	5,866	127,311	239	1.92	0.960	78.40

Example: Workload LGWR-THR

The LGWR-THR workload indicates the maximum REDO data the Log Writer processes can process.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Commit throughput [tps]	Commit latency [ms]	REDO blocks [rbps]	REDO writes [IOPS]	REDO syn writes [IOPS]	REDO data [MBps]	REDO data [kBpt]	Log file syn [ms]	FlCache write [%]
4	10	LGWR-THR	2	2	3	2	1	97	0	183	10.879	182,987	268	183	86	481.22	1.292	56.45
	11	LGWR-THR	2	18	17	14	2	83	0	1,201	14.639	1,216,741	2,078	1,203	571	486.85	2.335	64.68
	12	LGWR-THR	2	36	28	23	3	72	0	1,562	21.572	1,578,859	1,154	1,566	743	487.09	5.972	64.22
	13	LGWR-THR	2	54	28	24	3	72	0	2,084	24.709	2,092,195	986	2,090	986	484.48	8.866	56.93
	14	LGWR-THR	2	72	29	25	3	71	0	2,027	32.990	2,037,224	756	2,036	961	485.48	15.782	59.21

Notes

Oracle Data Guard. Synchronous replication is used for mission-critical systems using Oracle Data Guard as a BCP[8] solution. This means an additional latency of at least 1 ms for the log writers.

[8] A BCP (Business Continuity Plan) serves to protect data in the event of a data center failure. This is usually achieved by synchronous or asynchronous replication of the data to a second data center. Replication can take place at database level (Oracle Data Guard), at server level (host-based mirroring) or at storage level (storage-based mirroring).

Database Writer Workload

The database writers are responsible for managing the buffer cache. These processes are critical to the database system's overall performance when many blocks are modified, such as during conventional buffer cache data loading or intensive transaction processing.

The number of Database Writer processes is determined automatically by Oracle. If this value is not sufficient, it can be adjusted manually.

Performance metrics for Database Writer processes

The performance metric for Database Writer processes is:

- Throughput of database blocks written back in *database blocks per second* [dbps].

Workloads for determining the Database Writer performance

The DBWR-THR workload is available to assess Database Writer performance.

Workload	Action	Performance Indicator	Unit of measure
DBWR-THR	Massive block changes in the buffer cache.	Database Writer throughput	[dbps]

Example: Workload DBWR-THR

The DBWR-THR workload indicates the maximum number of blocks written back by the Database Writer processes.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Instance database writers	Log writes total [dbps]	Phys writes total [dbps]	Phys writes total [IOPS]	Phys writes total [MBps]	REDO data [MBps]	FlCache write [%]	Elapsed time [s]
5	1	DBWR-THR	2	2	3	2	1	97	0	12	163,897	37,975	22,905	297	43	77.66	302
	2	DBWR-THR	2	8	10	8	2	90	0	12	576,647	144,337	56,426	1,128	154	80.24	302
	3	DBWR-THR	2	16	17	14	3	83	0	12	1,037,316	272,138	60,318	2,126	277	81.18	302
	4	DBWR-THR	2	24	23	19	4	77	0	12	1,389,894	339,189	100,439	2,650	371	81.13	303
	5	DBWR-THR	2	32	28	22	5	72	0	12	1,439,232	338,007	187,978	2,641	383	78.99	304
	6	DBWR-THR	2	40	30	23	5	70	0	12	1,452,709	337,631	201,828	2,638	387	78.18	302
	7	DBWR-THR	2	48	30	22	5	70	0	12	1,371,240	316,917	211,589	2,476	365	76.19	308
	8	DBWR-THR	2	56	30	23	5	70	0	12	1,396,357	320,646	194,481	2,505	371	77.17	308

Notes

ASM redundancy. Different redundancy levels can be defined (*external*, *normal redundancy*, *high redundancy*). This has a significant impact on the Database Writer's performance. Switching from *high redundancy* to *normal redundancy* results in a considerable improvement in the performance of the Database Writer processes.

Data Load Workloads

System architects and capacity planners need performance metrics from database services regarding their ability to load data. This is especially important for data warehouse and data analytics systems, where data volumes are increasing and time windows for loading are decreasing.

Database systems provide several technologies for loading data:

- Conventional loading via the buffer cache; this technology is preferred in transaction processing systems.
- Direct loading bypasses the buffer cache; this method is mainly used in the data warehouse and analytics environment to quickly load large amounts of data.
- As of Oracle 19c, a new loading procedure is available for IoT (Internet of Things) applications. Mass data can be loaded quickly, but transaction consistency is not always guaranteed. This loading procedure is acceptable if aggregates such as average values, etc., do not depend on each data row.

Key figures for data load performance

There is only one performance metric for data load workloads:

- Throughput of loaded data (*data load rate*) in *MegaByte per second* [MBps] or *TeraByte per hour* ([TBph].

Workloads for determining the data load performance

The peakmarks® Benchmark Software offers workloads for all three loading techniques.

Workload	Action	Performance Indicator	Unit of measure
DL-BUFFER	The data is program-generated and conventionally loaded via buffer cache. Three additional indexes are updated during loading. The loading process uses *COMMIT WRITE WAIT IMMEDIATE* at transaction completion.	Data Load throughput	[MBps]
DL-DIRECT	Data is loaded from a data source, directly bypassing the buffer cache. Only one additional index is updated during loading. The loading process uses the *NOLOGGING* option and *COMMIT* WRITE *WAIT IMMEDIATE on* transaction completion.	Data Load throughput	[MBps]
DL-STREAM	The data is program-generated and uses the *memory-optimized row store for write* for IoT applications. Only one additional index is updated during loading. The load process uses *COMMIT WRITE NOWAIT BATCH* at transaction completion.	Data Load throughput	[MBps]

Example: Workload DL-BUFFER

The workload DL-BUFFER shows the maximum load rate achieved with the *buffered data load.* This method of loading is preferred in transaction-oriented systems.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Loaded user data [rps]	Loaded user data [MBps]	REDO data [MBps]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
6	1	DL-BUFFER	2	2	3	2	1	97	0	101,728	29	96	99.98	1.25	30.20	301
	2	DL-BUFFER	2	8	7	6	1	93	0	358,421	103	337	99.99	2.42	36.73	302
	3	DL-BUFFER	2	16	13	11	2	87	0	615,284	176	579	99.99	3.84	38.36	302
	4	DL-BUFFER	2	24	17	15	2	83	0	848,914	243	799	99.99	5.74	37.88	302
	5	DL-BUFFER	2	32	21	19	2	79	0	1,022,363	293	963	99.99	8.50	37.70	302
	6	DL-BUFFER	2	40	24	21	2	76	0	1,094,665	313	1,031	99.98	14.57	43.89	303
	7	DL-BUFFER	2	48	24	22	2	76	0	1,082,419	310	1,020	99.98	16.07	47.01	302
	8	DL-BUFFER	2	56	25	22	2	75	0	1,081,935	310	1,019	99.98	18.41	49.99	302

Example: Workload DL-DIRECT

The workload DL-DIRECT shows the maximum load rate achieved with the *direct data load* bypassing the buffer cache. This loading method is preferably used in data warehouse systems.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Loaded user data [rps]	Loaded user data [MBps]	REDO data [MBps]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
6	9	DL-DIRECT	2	2	3	2	1	97	0	664,166	190	65	99.70	7.36	74.54	301
	10	DL-DIRECT	2	8	8	7	1	92	0	2,158,221	617	229	99.81	85.23	78.66	302
	11	DL-DIRECT	2	16	13	11	2	87	0	3,514,681	1,006	351	99.80	88.23	79.46	302
	12	DL-DIRECT	2	24	16	13	2	84	0	4,557,863	1,304	385	99.78	86.23	80.29	303
	13	DL-DIRECT	2	32	18	15	3	82	0	5,060,505	1,448	449	99.75	83.98	78.71	303
	14	DL-DIRECT	2	40	18	15	2	82	0	5,057,368	1,447	476	99.71	79.26	76.75	303
	15	DL-DIRECT	2	48	18	15	2	82	0	5,280,355	1,511	489	99.67	76.93	76.50	303
	16	DL-DIRECT	2	56	19	16	3	81	0	5,547,978	1,587	480	99.63	77.03	76.99	304

Example: Workload DL-STREAM

The workload DL-STREAM shows the maximum load rate achieved in *stream mode.* This method of loading is preferred in Internet-of-Things applications.

Run	Test	Workload	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Loaded user data [rps]	Loaded user data [MBps]	REDO data [MBps]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
6	17	DL-STREAM	2	2	3	2	1	97	0	248,085	71	103	99.90	1.19	36.66	302
	18	DL-STREAM	2	8	8	7	1	92	0	880,591	252	365	99.92	2.78	42.71	302
	19	DL-STREAM	2	16	14	12	2	86	0	1,605,357	459	665	99.92	4.22	45.06	303
	20	DL-STREAM	2	24	21	19	2	79	0	2,313,229	662	959	99.93	9.28	47.96	304
	21	DL-STREAM	2	32	29	24	3	71	0	2,303,407	659	957	99.92	14.29	71.13	303
	22	DL-STREAM	2	40	31	26	3	69	0	2,306,234	660	956	99.91	14.93	70.64	302
	23	DL-STREAM	2	48	32	27	4	68	0	2,309,945	661	960	99.91	14.45	70.36	302
	24	DL-STREAM	2	56	32	27	4	68	0	2,296,175	657	954	99.90	16.48	68.21	303

Data Analytics Workloads

System architects and capacity planners require performance metrics from Database Services regarding their ability to search large amounts of data. These applications are typically based on *full table scan* operations and are not supported by index structures.

Full table scans' performance depends on the data's position in the storage hierarchy (storage, main memory) and the technology used to increase scan performance (*Smart Scan*, *In-Memory Column Store*).

The *Smart Scan* technology is only available on Oracle Engineered Systems and is additionally subject to licensing. The *In-Memory* technology is available on many platforms and is license-free from Oracle 20c to a specific capacity.

Data analytics tests answer questions about both technologies' efficiency and enable price-performance comparisons.

Key figures for data analytics performance

For data analytics workloads, there is only one performance metric:

- Throughput of scanned data (*data scan rate*) in *megabytes per second* [MBps].

Workloads for determining the data analytics performance

peakmarks® offers workloads to test different data locations (storage, main memory) and test boost technologies (*Smart Scan*, *In-Memory Column Store*). Here, the intra-SQL parallelism can be modified to find the optimal database parallelism (DOP) during Data Scan.

Workload	Action	Performance Indicator	Unit of measure
DA-STORAGE	*Full table scan* with an aggregate of low complexity. The data is read from the storage system.	Data Scan throughput	[MBps]
DA-OFFLOAD	*Full table scan* with an aggregate of low complexity. The intelligent storage system processes the data with offload functionality (*smart scan*).	Data Scan throughput	[MBps]
DA-ROWSTORE	*Full table scan* with an aggregate of low complexity. The data is read from the *row store* in the buffer cache.	Data Scan throughput	[MBps]
DA-COLSTORE	*Full table scan* with an aggregate of low complexity. The data is processed by the *column store* in the buffer cache.	Data Scan throughput	[MBps]

Example: Workload DA-STORAGE

The DA-STORAGE workload shows the maximum scan rate when all data must be read from the storage. The processing of the data scan (filter operations) takes place on the database server.

The *Scanned user data [MBps]* metric corresponds to the actual amount of data scanned. The *Scanned user data [rps]* metric depends on the data model and applies to a record length of 300 bytes (configuration parameters of the peakmarks® Benchmark Software).

Run	Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
7	1	DA-STORAGE	2	2	4	3	2	1	97	0	18,602,359	6,328	0.00	100.00	171
	2	DA-STORAGE	2	8	4	5	4	1	95	0	30,549,385	10,391	0.00	100.00	172
	3	DA-STORAGE	2	16	4	5	4	1	95	0	31,959,319	10,871	0.00	100.00	172
	4	DA-STORAGE	2	24	4	6	5	1	94	0	32,656,231	11,108	0.00	100.00	172
	5	DA-STORAGE	2	32	4	6	5	1	94	0	33,066,694	11,247	0.00	100.00	173

Example: Workload DA-OFFLOAD

The DA-OFFLOAD workload shows the maximum scan rate when all data is processed by the intelligent storage system (filter operations). Only the result set is transferred to the database server.

This offload technology is only available on Oracle Engineered Systems and requires additional licenses for their storage servers.

Run	Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
7	6	DA-OFFLOAD	2	2	4	3	2	1	97	0	33,502,974	11,395	0.00	100.00	171
	7	DA-OFFLOAD	2	8	4	4	3	1	96	0	71,284,198	24,245	0.00	100.00	171
	8	DA-OFFLOAD	2	16	4	5	3	2	95	0	80,138,103	27,257	0.00	100.00	172
	9	DA-OFFLOAD	2	24	4	5	3	2	95	0	82,588,125	28,090	0.00	100.00	172
	10	DA-OFFLOAD	2	32	4	5	3	2	95	0	83,859,267	28,522	0.00	100.00	172

Example: Workload DA-ROWSTORE

The workload DA-ROWSTORE shows the maximum scan rate when all data is processed in the buffer cache of the Database Server (e.g., in the buffer cache's keep pool). A large main memory capacity has an advantageous effect in storing as much data as possible in the buffer cache.

Run	Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
7	11	DA-ROWSTORE	2	1	1	2	1	1	98	0	4,092,183	3,449	98.34	0.00	181
	12	DA-ROWSTORE	2	36	1	26	25	1	74	0	166,341,215	52,351	99.89	0.00	182
	13	DA-ROWSTORE	2	72	1	51	51	1	49	0	295,742,535	91,330	100.00	0.00	181
	14	DA-ROWSTORE	2	108	1	76	75	1	24	0	344,596,712	106,019	100.00	0.00	181
	15	DA-ROWSTORE	2	144	1	98	97	1	2	0	373,359,152	114,112	100.00	0.00	181

Example: Workload DA-COLSTORE

The workload DA-COLSTORE shows the maximum scan rate when all data is processed in the *Database Server column store*. The use of this *in-memory* technology requires appropriate licenses on the database server.

It is unnecessary to store entire tables in the Column Store, but only individual selected columns are frequently used in the search criterion. Compression methods can also increase the amount of data stored in the column store. A large main memory capacity is also advantageous for keeping as much data as possible in the *column store.*

Run	Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]	Elapsed time [s]
7	16	DA-COLSTORE	2	2	1	3	2	1	97	0	104,128,104	39,666	0.00	0.00	181
	17	DA-COLSTORE	2	36	1	26	25	1	74	0	1,654,589,283	630,295	0.00	0.00	181
	18	DA-COLSTORE	2	72	1	51	50	1	49	0	2,867,362,100	1,092,286	0.00	0.00	181
	19	DA-COLSTORE	2	108	1	76	75	1	24	0	3,032,650,884	1,155,250	0.00	0.00	181
	20	DA-COLSTORE	2	144	1	97	95	1	3	0	3,099,355,999	1,180,661	0.00	0.00	181

Notes

The efficiency of the various processes. It is easy to see that *in-memory* technology with the *column store has the* highest performance and leaves *smart scan* technology far behind. Therefore, in recent years, *in-memory* technology has become the preferred technology for data analytics applications.

Price performance comparison of the different processes. Additional licenses are required for both *Smart Scan* technology and *in-memory* technology. It is noticeable that the *in-memory* technology has a significantly better price-performance ratio than the *Smart Scan* technology if the scan rates are considered concerning the license costs.

However, *Smart Scan* technology works mostly transparently for applications; code changes are rarely necessary. *Smart Scan* technology can be combined with *in-memory* technology on the storage servers and manage immense data capacities. However, *Smart Scan* technology can only be used on Oracle Engineered Systems.

On the other hand, *in-memory* technology is available on many platforms, requires no special hardware, and delivers ultimate performance. However, the data amount is limited by the database server's main memory capacity (DRAM, soon also PMEM). Furthermore, the application's interventions must specify which table columns should be included in the *column store.* Compression methods help to use the capacity of the *column store* optimally.

Transaction Processing Workloads

System architects and capacity planners need performance metrics from database services regarding their ability to execute typical transactions.

Transaction processing is a very complex form of data processing. All platform components and database service processes are significantly involved and must be precisely balanced.

Transaction processing performance depends on very many factors. Essential factors are:

- Ratio of database size and buffer cache size, expressed in the buffer cache hit rate.
- I/O performance when reading data.
- II/O performance when writing the transaction log by REDO Log Writer processes.
- I/O performance when writing data through Database Writer processes.

Storage tiering and many other factors can influence the I/O performance for reading data. The I/O performance of the log writer processes can be affected by technologies mirroring data between different data centers.

Key figures for transaction processing performance

For transaction processing workloads, the following performance metrics are of interest:

- SQL *transaction throughput* in *transactions per second* [tps].
- Average SQL *transaction response time* in *milliseconds* [ms].

Workloads for determining transaction processing performance

The peakmarks® Benchmark Software provides transaction processing workloads of different complexity (light, medium, and heavy). The ratio between reading and changing operations can be configured.

Workload	Action	Performance Indicator	Unit of measure
TP-LIGHT	Select/Update single record via index. Example: select/update account, product, order, invoice, etc. Transactions are always completed with a *COMMIT WRITE WAIT IMMEDIATE.*	Transaction throughput Transaction response time	[tps] [ms]
TP-MEDIUM	Select/Update Ø 25 records via index. Example: select/update account postings last week; item list of order, etc. Transactions are always completed with a *COMMIT WRITE WAIT IMMEDIATE.*	Transaction throughput Transaction response time	[tps] [ms]
TP-HEAVY	Select/Update Ø 125 records via index.	Transaction throughput	[tps]

	Transactions are always completed with a *COMMIT WRITE WAIT IMMEDIATE.* Example: select/update mobile phone call records from last month, etc.	Transaction response time	[ms]

Example: Workload TP-LIGHT

The TP-LIGHT workload shows the throughput and average response time for transactions with low complexity. The ratio of reading and changing transactions is 80:20 (column *Upd [%]*).

Run	Test	Workload	Upd [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Transactions per cpu [tps]	Response time [ms]	IO time read [ms]	LogFile sync [ms]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
8	1	TP-LIGHT	20	2	2	2	1	1	98	0	3,070	1,535	0.651	0.290	0.990	83.48	98.82	90.74	300
	2	TP-LIGHT	20	2	16	8	5	2	92	0	36,510	2,282	0.436	0.296	0.062	82.45	97.40	96.25	303
	3	TP-LIGHT	20	2	32	16	11	4	84	0	99,359	3,105	0.321	0.326	0.034	88.45	96.11	97.73	302
	4	TP-LIGHT	20	2	48	23	17	5	77	0	138,796	2,892	0.345	0.374	0.053	89.08	96.26	97.66	302
	5	TP-LIGHT	20	2	64	29	21	6	71	0	171,305	2,677	0.372	0.419	0.053	89.01	96.37	97.26	302
	6	TP-LIGHT	20	2	80	33	24	7	67	0	192,770	2,410	0.412	0.462	0.056	88.98	96.48	96.77	303
	7	TP-LIGHT	20	2	96	38	28	8	62	0	218,774	2,279	0.437	0.515	0.217	89.02	96.49	96.81	302
	8	TP-LIGHT	20	2	112	41	30	8	59	0	231,635	2,068	0.482	0.563	0.913	89.04	96.50	96.42	302

In addition to the primary performance indicators such as throughput and response time, the benchmark report for transaction-oriented workloads also lists important influencing factors: Buffer Cache Hitrate (Column *BuCache read [%]*), Flash Cache Hitrate - if available (column *FlCache read [%]*), the service time at random read (column *IO time read [ms]*) and the latency from the log writer at transaction completion (column *Log File syn [ms]*).

Example: Workload TP-MEDIUM

The TP-MEDIUM workload shows the throughput and average response time for transactions of medium complexity.

Run	Test	Workload	Upd [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Transactions per cpu [tps]	Response time [ms]	IO time read [ms]	LogFile sync [ms]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
8	9	TP-MEDIUM	20	2	2	4	2	2	96	0	790	395	2.532	0.302	0.562	76.30	96.98	94.11	307
	10	TP-MEDIUM	20	2	16	11	7	3	89	0	5,090	318	3.130	0.378	0.114	61.97	96.24	97.23	302
	11	TP-MEDIUM	20	2	32	18	12	5	82	0	7,986	250	3.995	0.479	0.078	60.64	96.76	95.77	302
	12	TP-MEDIUM	20	2	48	23	15	6	77	0	10,830	226	4.420	0.582	0.073	60.83	97.02	94.97	302
	13	TP-MEDIUM	20	2	64	26	17	7	74	0	12,774	200	4.998	0.690	0.061	62.02	97.42	93.77	302
	14	TP-MEDIUM	20	2	80	28	18	7	72	0	15,285	191	5.218	0.768	0.086	62.66	98.81	92.03	302
	15	TP-MEDIUM	20	2	96	32	22	8	68	0	17,006	177	5.633	0.830	0.113	63.35	98.93	91.01	302
	16	TP-MEDIUM	20	2	112	35	23	8	65	0	18,682	167	5.980	0.926	0.237	62.41	99.01	90.13	302

Example: Workload TP-HEAVY

The TP-HEAVY workload shows the throughput and average response time for more complex transactions.

Run	Test	Workload	Upd [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Transactions per cpu [tps]	Response time [ms]	IO time read [ms]	LogFile sync [ms]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
8	17	TP-HEAVY	20	2	2	5	3	2	95	0	513	257	3.895	0.292	0.000	75.52	93.46	87.38	301
	18	TP-HEAVY	20	2	16	12	8	3	88	0	2,222	139	7.176	0.479	0.227	59.80	97.59	90.60	302
	19	TP-HEAVY	20	2	32	19	13	5	81	0	3,500	109	9.113	0.780	0.132	60.89	98.73	90.13	302
	20	TP-HEAVY	20	2	48	25	17	6	75	0	4,269	89	11.211	1.148	0.221	63.46	99.10	89.03	302
	21	TP-HEAVY	20	2	64	28	18	7	72	0	4,617	72	13.826	1.572	0.090	62.81	99.11	88.87	302
	22	TP-HEAVY	20	2	80	31	21	7	69	0	4,697	59	16.890	2.038	1.093	61.79	99.16	88.97	308
	23	TP-HEAVY	20	2	96	32	21	8	68	0	4,973	52	19.270	2.633	0.105	61.36	99.24	88.78	301
	24	TP-HEAVY	20	2	112	35	23	8	65	0	4,970	44	22.495	3.157	0.053	62.00	99.02	89.06	301

Notes

Impact of storage tiering. If storage systems have storage tiering, care must be taken to ensure that all operational data is continuously accessed in the best tier. As the following example shows, there can be dramatic performance losses if this is not the case.

Run	Test	Workload	Upd [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Transactions per cpu [tps]	Response time [ms]	IO time read [ms]	LogFile sync [ms]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
8	1	TP-LIGHT	20	2	2	2	1	1	98	0	3,070	1,535	0.651	0.290	0.990	83.48	98.82	90.74	300
	2	TP-LIGHT	20	2	16	8	5	2	92	0	36,510	2,282	0.436	0.296	0.062	82.45	97.40	96.25	303
	3	TP-LIGHT	20	2	32	16	11	4	84	0	99,359	3,105	0.321	0.326	0.034	88.45	96.11	97.73	302
	4	TP-LIGHT	20	2	48	23	17	5	77	0	138,796	2,892	0.345	0.374	0.053	89.08	96.26	97.66	302
	5	TP-LIGHT	20	2	64	29	21	6	71	0	171,305	2,677	0.372	0.419	0.053	89.01	96.37	97.26	302
	6	TP-LIGHT	20	2	80	33	24	7	67	0	192,770	2,410	0.412	0.462	0.056	88.98	96.48	96.77	303
	7	TP-LIGHT	20	2	96	38	28	8	62	0	218,774	2,279	0.437	0.515	0.217	89.02	96.49	96.81	302
	8	TP-LIGHT	20	2	112	41	30	8	59	0	231,635	2,068	0.482	0.563	0.913	89.04	96.50	96.42	302

Run 8, tests 1 to 8 were performed with a 4 TByte database that fits entirely into the flash cache. The values for the flash cache hit rate (column *FlCache read [%]*) are correspondingly good, and the I/O service time (column *IO time read [ms]*) is in the expected range.

Run	Test	Workload	Upd [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Transactions per cpu [tps]	Response time [ms]	IO time read [ms]	LogFile syn [ms]	BuCache read [%]	FlCache read [%]	FlCache write [%]	Elapsed time [s]
9	1	TP-LIGHT	20	2	2	1	1	1	99	0	143	72	13.999	4.579	0.943	99.72	18.78	95.77	183
	2	TP-LIGHT	20	2	16	2	1	1	98	0	1,117	70	14.231	6.521	0.141	98.16	15.39	97.92	182
	3	TP-LIGHT	20	2	32	2	1	1	98	0	1,798	56	17.696	8.223	0.085	96.26	11.57	98.39	182
	4	TP-LIGHT	20	2	48	2	1	1	98	0	2,178	45	21.885	10.576	0.110	94.58	7.34	98.35	183
	5	TP-LIGHT	20	2	64	3	2	1	97	0	3,035	47	20.910	9.513	0.271	92.21	12.93	98.39	183
	6	TP-LIGHT	20	2	80	3	2	1	97	0	3,799	47	20.871	9.952	0.072	90.15	12.20	98.55	183
	7	TP-LIGHT	20	2	96	3	2	1	97	0	4,101	43	23.006	12.149	0.111	90.96	12.16	98.51	186

Run 9, tests 1 to 7, shows the same workload with a 32 TByte database that does not fit into the flash cache with a capacity of 19.2 TByte. The values for the flash cache hit rate (column *FlCache read [%]*) are very low, the I/O service time (column *IO time read [ms]*) experiences extreme fluctuations, and the transaction throughput collapses.

PL/SQL Performance

PL/SQL is Oracle's preferred programming language for complex transaction logic and algorithms. PL/SQL programs are stored and compiled in the database server. Some large applications, such as *Core Banking Systems*, are implemented entirely in PL/SQL.

Key figures for PL/SQL performance

For PL/SQL workloads, the following performance metrics are of interest:

- The number of executed PL/SQL operations per time unit in a *million operations per second* [Mops].
- The execution time of PL/SQL algorithms in *seconds* [s].

Workloads to determine PL/SQL performance

peakmarks® offers workloads to test the efficiency of PL/SQL programs on a specific processor. These workloads are completely limited by CPU performance.

Workload	Action	Performance In-dicator	Unit o measure
PLS-ADD	Addition of numbers	Throughput PL/SQL operations	[Mops]
PLS-BUILTIN	Data type-specific operations, including SQL functions, based on Core Banking Systems and Telco Billing Systems	Throughput PL/SQL operations	[Mops]
PLS-PRIME	Calculation of the first n prime numbers	Processing time of an algorithm	[s]
PLS-FIBO	Calculation of the first n Fibonacci numbers (recursive algorithm)	Processing time of an algorithm	[s]
PLS-MIXED	A complex workload with a mix of equally-weighted simple workloads running concur-rently: ▪ PLS_ADD with data type NUMBER ▪ PLS_ADD with data type PLS_INTEGER ▪ PLS_BUILTIN with data type NUMBER ▪ PLS_BUILTIN with data type VARCHAR2	Throughput PL/SQL operations	[Mops]

Example: Workload PLS-MIXED

The PLS-MIXED workload shows the throughput of different PL/SQL operations with different data types.

Run	Test	Workload	Data type	N	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Operations total [Mops]	Operations per cpu [Mops]	Elapsed time [s]
10	1	PLS-MIXED	0	0	2	8	6	6	1	94	633.04	79.13	121
	2	PLS-MIXED	0	0	2	72	50	49	1	50	4,641.97	64.47	121
	3	PLS-MIXED	0	0	2	144	95	94	1	5	4,755.79	33.03	121

Example: Workload PLS-FIBO

The PLS-FIBO workload displays the processing time for an algorithm that calculates Fibonacci numbers (column *N*). The data type used can be configured (column *Data type*).

Run	Test	Workload	Data type	N	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Operations total [Mops]	Operations per cpu [Mops]	Elapsed time [s]
10	4	PLS-FIBO	SI	39	1	1	6	3	2	94	0.000	0.000	16
	5	PLS-FIBO	SI	40	1	1	6	4	3	94	0.000	0.000	26
	6	PLS-FIBO	SI	41	1	1	6	4	2	94	0.000	0.000	41
	7	PLS-FIBO	SI	42	1	1	6	4	2	94	0.000	0.000	68

Conclusion

peakmarks® Benchmark Software provides system engineers and system architects with a robust and comprehensive benchmark framework for determining meaningful and understandable performance metrics of Database Services - **on-premises** and in the **cloud**.

The easy-to-understand performance metrics of peakmarks® make technologies, configurations, components, and complete systems comparable across vendors and service providers.

The peakmarks® Benchmark Software covers all conceivable workloads for a performance test of the infrastructure. The software is continuously being further developed and adapted to new database versions.

Benefits for IT operations. Many IT organizations can reduce the cost of database services by choosing *best-in-class* infrastructure components or *best-in-class* cloud services and minimizing licensing and maintenance costs.

Bottlenecks, misconfigurations, and malfunctions are detected before going live. Regular performance analyses of cloud services ensure performance quality. peakmarks® performance metrics are a factual basis for capacity planning when migrating to other platforms or cloud services.

The result is a Database Service with *predictable* and *persistent* performance in all load situations.

Benefits for IT hardware vendors and service providers. peakmarks® benchmarks are fast and efficient and support the process of developing and marketing license-optimized database solutions with the best price-performance ratio.

Customers and prospects benefit from understandable, user-friendly, and comparable performance metrics that can avoid time-consuming and expensive proofs-of-concept.

peakmarks® performance indicators help strengthen the positioning of a provider compared to the competition.