Benchmarking Database Cloud Services

Manfred Drozd(✉)

Peakmarks Ltd., Zürcherstrasse 59, 8800 Thalwil, Switzerland
manfred.drozd@peakmarks.com

Abstract. Meanwhile, many database cloud services are available. The well-known providers are AWS (Amazon Web Service), Google Cloud, and Azure (Microsoft). Oracle and IBM offer cloud services for their in-house database products. In the past, the TPC organization has focused on performance measurement of database systems. Often, however, a database system is predefined, and the question arises as to the most efficient infrastructure and the best price/performance ratio - whether on-premise or as a cloud service. On the Internet, you can hardly find comparable and traceable information about the performance of database cloud services. Therefore, it is challenging to make corresponding price/performance comparisons [1]. The company Peakmarks was founded in 2011 to provide a robust and comprehensive benchmarking framework to identify representative performance indicators of database services. Peakmarks does not sell any hardware but runs benchmarks on behalf of users and manufacturers and thus guarantees absolute independence. Users can license Peakmarks benchmark software to perform their own performance tests. This presentation gives a rough overview of the Peakmarks benchmark software, its architecture, and workloads. Examples are used to show how understandable key performance metrics for database cloud services can be determined quickly and practically.

Keywords: Benchmark · Databases · Cloud services

1 Requirements to Benchmark Software

Huppler [3] described the five most important characteristics of a good benchmark: relevant, repeatable, fair, verifiable, and economical. Peakmarks[1] meets all these requirements.

However, other features are also crucial for customer acceptance of benchmark software:

Simplicity. It must be easy to install the benchmark software, perform the benchmark, and interpret the results. Peakmarks is implemented with the tools of the database without operating system scripts. Therefore, Peakmarks runs unchanged everywhere where the database software is available. Any DBA can easily manage the benchmark software without additional know-how.

[1] From now on, we use the word Peakmarks synonymously with Peakmarks benchmark software.

R. Nambiar and M. Poess (Eds.): TPCTC 2019, LNCS 12257, pp. 139–153, 2020.
https://doi.org/10.1007/978-3-030-55024-0_10

Speed. The installation, loading of the data, processing of the various workloads, and evaluation of the performance key figures should be fast. Peakmarks is installed in a few hours, including all adjustments of the database. The loading time of the database depends on the database size and the performance of the infrastructure. On powerful systems, loading times of 4 TByte per hour were measured. Complete benchmark runs with all workloads typically take between 12 and 24 h; the results are immediately available. A comprehensive benchmark project can be completed within a week. This is significantly faster than many proof-of-concepts, which may take several weeks and whose value is limited to the tested application.

Different Load Situations. Often it is not the maximum value of a performance metric that is of interest, but the optimal value. Peakmarks analyzes the performance of a database service in all load situations. A benchmark test starts with a low load and increases the load continuously until the system is saturated. In this way, the optimum performance range of a database service can be determined.

More Performance Metrics. Many benchmarks provide only a single performance metric. This dramatically simplifies the comparison of different systems. However, a single metric is difficult to understand [2]. Peakmarks provides a set of representative and easy-to-understand metrics for different aspects. Actual performance questions can be answered more easily. Performance bottlenecks and malfunctions can be detected more easily. Since several performance metrics are available, the user must decide with which priority the individual metrics are to be included in the decision-making process, when choosing the right cloud service.

Product Specific Workloads. When customers have to pay license fees for database software, they are interested in getting the highest performance out of their database service. That's why we've deliberately implemented workloads that can only be found on certain database products but are essential for the solution architecture. Currently, Peakmarks is available for Oracle 12.2 and upwards. There are considerations to port the software to other database systems as well. Peakmarks is not suitable to compare different database products; it only serves to compare the underlying infrastructure, on-premise or in the cloud.

2 Key Performance Indicators

Representative performance indicators of database services can be used for various tasks:

Quality Assurance. A database service is validated for its performance properties. Performance bottlenecks can be quickly identified; performance promises of the providers are easily checked.

Evaluation. Performance indicators are used for price/performance considerations of various database services, technologies, or configurations.

Capacity Planning. When systems migrate to new platforms or cloud services, performance indicators help with capacity planning.

License Cost Optimization. Our experience over many years has shown that many users can halve license costs for the same performance, only by optimizing the infrastructure. License costs often exceed infrastructure costs by far.

3 The Architecture of Peakmarks Benchmark Software

Peakmarks is written in the procedural SQL extension of the database system, in the case of Oracle in PL/SQL. The size of the database can be configured in a range from 250 GByte up to 64 TByte per database server. Clusters with multiple servers are supported. The record length of the benchmark tables can be configured between 80 bytes and 4'000 bytes. The redundancy of the benchmark data can be controlled via a parameter. The data can optionally be encrypted using a further configuration parameter. All encryption methods offered by the database system are supported. The scalable loading process of the benchmark data automatically adapts its parallelism to the performance capabilities of the database platform.

A workload generator generates the database load with database jobs, and a performance monitor collects all relevant performance statistics before and after each performance test. All workloads are generated within the database, and all performance statistics also originate from the database (Fig. 1).

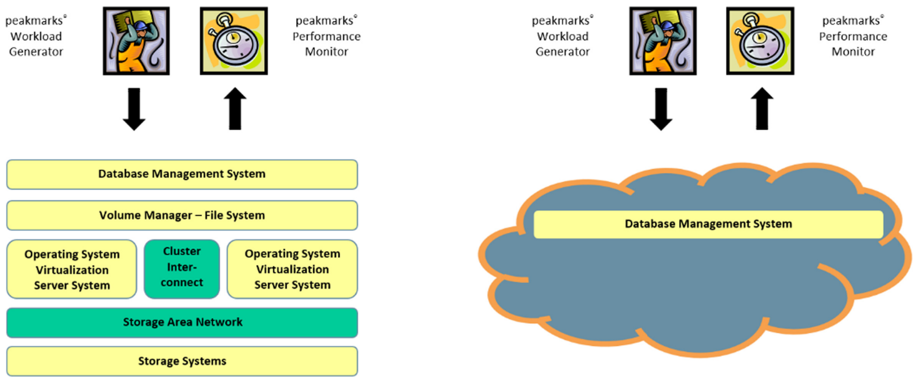

Fig. 1. Peakmarks benchmark software – on-premise and in the cloud

Peakmarks provides a library of workloads to determine the most important performance indicators of database services for:

- Server and storage systems in database operation.
- Critical database background processes, responsible for transaction management (log writer) and buffer management (database writer).
- Typical database operations such as data load, data analytics and transaction processing.
- PL/SQL application code.

4 Simple and Complex Workloads

Peakmarks distinguishes between simple and complex workloads. Simple workloads execute precisely one type of load (SQL statement). Complex workloads are hierarchically composed of different simple workloads to simulate complex load situations.

The runtime of the workloads is configurable. Runtimes between 5 and 10 min per test have proven to be representative. Let us take a closer look at some of the workloads.

4.1 Server Workloads

Server workloads determine the power of a server with its processors, main memory, and internal memory channels in database operation. These workloads are especially crucial if license cost must be optimized. License costs are often linked to the number of processors[2] used. In this case, a server with the highest performance per processor is searched for.

Server workloads also show the efficiency of multithreading and virtualization technologies and provide hints about scalability when high numbers of sockets and cores are used (NUMA effects). If database encryption is selected, its impact on the overall database performance can also be determined.

All server workloads access tables via SQL with different access patterns. The affected tables are fixed in the buffer cache. There are almost no I/O operations, so these workloads are entirely CPU-bound.

The essential primary performance metrics are queries per second (qps), the response time, and the scan rate of queries (memory bandwidth). A secondary performance metric is the number of logical reads per second (Table 1).

Table 1. Peakmarks workloads to determine server performance.

Workload	Action	Key performance metric	Unit
SRV-LIGHT	Select single row via index. Example: select account, product, order, invoice, etc.	• Query throughput • Response time	[qps] [μs]
SRV-MEDIUM	Select avg 25 rows via index. Example: select account postings last week; item list of order, etc.	• Query throughput • Response time	[qps] [μs]
SRV-HEAVY	Select avg 125 rows via index. Example: report of last month's call records, etc.	• Query throughput • Response time	[qps] [μs]
SRV-SCAN	Data search without index support	Buffer scan rate	[MBps]
SRV-MIXED	Complex workload with a mix of simple server workloads and concurrent table scan.	• Query throughput • Response time	[qps] [μs]

2 A processor may be a core or a thread, dependent on processor architecture and the used multithreading technology.

The benchmark report in Fig. 2 shows the performance of a database service for workload SRV-LIGHT. This benchmark comprises 5 tests. The second column shows the workload name. The column *Nodes* indicate how many cluster nodes are used in the test. The column *Jobs* describes the number of processes that generate the load for the workloads. The next 4 columns describe the percentage CPU load in the different CPU modes. The column *Queries total* describes the total number of queries processed per second. The column *Queries per cpu* shows the performance per involved processor. This information is important for license cost considerations. The columns *Logical reads total* and *Logical reads per cpu* are the corresponding performance metrics for database accesses in the buffer cache. The column *BuCache read* displays the hit rate of all read accesses in the buffer cache and is only used to check that this workload has been optimally processed.

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 [%]
1	SRV-LIGHT	1	1	13	12	1	87	62,241	62,241	0.016	186,890	186,890	100.0
2	SRV-LIGHT	1	2	25	24	1	75	115,499	57,750	0.017	346,601	173,301	100.0
3	SRV-LIGHT	1	4	50	48	2	50	198,302	49,575	0.020	595,053	148,763	100.0
4	SRV-LIGHT	1	8	99	95	4	1	284,839	35,605	0.028	854,567	106,821	100.0
5	SRV-LIGHT	1	12	99	95	4	1	283,995	35,499	0.041	848,451	106,056	100.0

Fig. 2. Benchmark report for a server system workload.

It is straightforward to see how, as the load increases, the response time also increases, but the number of queries per CPU decreases. The short response times of less than 30 μs show the exceptional efficiency of database queries when all the data is in the buffer cache. It is also noticeable that throughput and CPU utilization do not correlate above 50%; a typical characteristic of some processor architectures when multi-threading is enabled.

4.2 Storage Workloads

Conventional I/O benchmark tools such as vdbench, iometer, Orion often display performance values that are not achieved in real database operations. The reason for this is the complexity of database I/O operations.

If a data block is read, the buffer cache management of the database has to perform many tasks: a) find a free slot for the block; b) if there is no free slot, replace older blocks; c) synchronize all database processes that simultaneously try for free slots in the buffer cache; d) if a shared disk cluster architecture is used, the synchronization has to be cluster-wide; e) finally, blocks are checked for their integrity and consistency during I/O transfer.

Peakmarks, therefore, generates I/O load with so-called SQL-generated I/O operations to obtain representative performance metrics for the storage system (Table 2).

Table 2. Peakmarks workloads to determine storage performance.

Workload	Action	Key performance metric	Unit
STO-READ	SQL generated sequential block read	Sequential I/O throughput	[MBps]
STO-RANDOM	SQL generated random block read/write	• Random I/O throughput in database blocks per second • I/O service time	[dbps] [μs]

Storage workloads show the efficiency of the I/O stack (I/O scheduler, queues, multipathing, virtualization), the technologies used (HDD, SSD, Flash, SAS or PCI, NVMe, etc.) and storage specific functionalities (deduplication, compression, encryption, snapshots, mirroring, SQL offloading).

The benchmark report in Fig. 3 shows the performance of a database service for workload STO-RANDOM with 100% read operations.

Test	Workload	Wri [%]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Phys reads total [dbps]	Phys reads total [IOPS]	Service time [us]	Phys reads total [MBps]	BuCache read [%]	FlCache read [%]
6	STO-RANDOM	0	1	1	2	1	1	98	0	33,390	33,410	167	261	0.0	0.0
7	STO-RANDOM	0	1	2	3	1	1	99	0	63,570	63,590	175	497	0.0	0.0
8	STO-RANDOM	0	1	4	5	3	3	95	0	117,600	117,600	182	919	0.1	0.0
9	STO-RANDOM	0	1	8	10	5	5	90	0	209,800	209,800	194	1,640	0.1	0.0
10	STO-RANDOM	0	1	16	20	10	9	80	0	355,000	355,000	211	2,774	0.0	0.0
11	STO-RANDOM	0	1	32	37	21	17	63	0	549,700	549,700	243	4,295	0.1	0.0
12	STO-RANDOM	0	1	64	68	39	30	32	0	756,900	772,800	451	6,038	0.1	0.0
13	STO-RANDOM	0	1	96	88	48	39	12	0	801,900	825,200	945	6,447	0.2	0.0
14	STO-RANDOM	0	1	128	89	49	40	11	0	792,700	823,200	1,563	6,432	0.3	0.0

Fig. 3. Benchmark report for random read storage system workload.

This report shows the difference between the maximum and optimal range of performance. The storage system can read over 800,000 random single database blocks per second (dbps), but at a service time of just under one millisecond (Test 13). An all-flash storage system is used in this case study. We expect a service time of less than 500 microseconds per single database block read for this storage technology. The optimal performance is more like 750,000 dbps (Test 12). Higher values are possible, but only at the price of sharply increasing service times. It is a good advice to keep the storage utilization below this value.

4.3 Data Load Workloads

System architects and capacity planners need performance metrics from database services regarding their ability to load data. This is particularly important for Data Warehouse and Data Analytics systems, where data volumes are constantly growing as the time available for loading becomes smaller.

Oracle provides different technologies for loading data: conventional loading via buffer cache and direct loading bypassing the buffer cache. Peakmarks provides workloads for both data loading techniques (Table 3).

Table 3. Peakmarks workloads to determine data load performance.

Workload	Action	Key performance metric	Unit
DL-BUFFER	Buffered data load	Data load throughput	[MBps]
DL-DIRECT	Direct data load	Data load throughput	[MBps]

The benchmark report in Fig. 4 shows the performance of a database service for both data load workloads. The key performance metrics for data load is the amount of data that can be loaded within a certain timeframe in column *Loaded user data.*

Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Loaded user data [rps]	Loaded user data [MBps]	REDO data [MByte]	BuCache read [%]	FlCache read [%]	FlCache write [%]
15	DL-BUFFER	1	1	1	2	1	1	98	0	29,935	9	30	100.0	56.7	89.7
16	DL-BUFFER	1	2	1	4	3	1	96	0	54,752	16	55	100.0	48.5	91.7
17	DL-BUFFER	1	4	1	7	6	1	93	0	105,138	32	107	100.0	49.4	91.8
18	DL-BUFFER	1	8	1	8	7	1	92	0	181,662	54	184	100.0	59.6	92.5
19	DL-BUFFER	1	16	1	12	10	1	88	0	284,286	85	288	100.0	63.0	93.9
20	DL-BUFFER	1	32	1	14	12	2	86	0	338,027	101	344	100.0	65.3	96.4
21	DL-BUFFER	2	64	1	30	26	3	70	0	786,876	236	798	100.0	59.4	96.1

Test	Workload	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Loaded user data [rps]	Loaded user data [MBps]	REDO data [MByte]	BuCache read [%]	FlCache read [%]	FlCache write [%]
22	DL-DIRECT	1	1	1	5	4	1	95	0	46,254	14	22	98.6	66.1	84.0
23	DL-DIRECT	1	2	1	5	4	1	95	0	92,640	28	44	99.3	62.5	86.0
24	DL-DIRECT	1	4	1	6	5	1	94	0	202,426	61	79	99.3	63.3	84.0
25	DL-DIRECT	1	8	1	7	6	1	93	0	387,664	116	121	99.3	67.9	87.7
26	DL-DIRECT	1	16	1	10	8	1	90	0	714,476	214	179	99.3	68.8	92.2
27	DL-DIRECT	1	32	1	16	14	2	84	0	1,144,732	343	225	99.3	68.5	93.5
28	DL-DIRECT	2	64	1	28	24	3	72	0	2,398,864	720	485	99.3	66.8	92.4

Fig. 4. Benchmark report for data load workloads.

This case study was run on an Oracle Engineered System which uses flash cache technology. The buffered load generates more REDO data. The direct load in workload DL-DIRECT provides much higher throughput in data load. In the last test of each workload, the load is doubled but distributed over two database servers. In both cases, the system scales well.

4.4 Data Analytic Workloads

System Architects and capacity planners require performance metrics from database services regarding their ability to search for data. Data analytics applications are typically based on “full table scan” operations. The performance of “full table scans” depends on the position of the data in the storage hierarchy and the technology used to boost scanning performance.

Peakmarks provides workloads to test different data locations (storage, memory) and to test different boost technologies (smart scan, in-memory column store) (Table 4).

Table 4. Peakmarks workloads to determine data analytics performance.

Workload	Action	Key performance metric	Unit
DA-LOW	Full table scan with an aggregate of low complexity	Data scan throughput	[MBps]

The benchmark report in Fig. 5 shows the performance of a database service for data analytic workloads. The key performance metric for data analytics is the amount of data that can be scanned within a certain timeframe in column *Scanned user data*.

Test	Workload	Loc	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]
29	DA-LOW	STO	1	1	1	3	2	1	97	2,957,446	1,009	0.0	99.9
30	DA-LOW	STO	1	2	1	3	2	1	97	6,504,650	2,220	2.9	99.9
31	DA-LOW	STO	1	4	1	2	1	1	98	11,139,254	3,802	0.0	100.0
32	DA-LOW	STO	1	8	1	5	4	1	95	16,622,389	5,674	0.0	100.0
33	DA-LOW	STO	1	16	1	5	4	1	95	16,620,501	5,673	0.0	100.0
34	DA-LOW	STO	1	24	1	4	3	1	96	16,841,046	5,748	0.0	100.0
35	DA-LOW	STO	2	48	1	5	4	1	95	33,858,898	11,557	0.0	100.0

Test	Workload	Loc	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]
36	DA-LOW	EXA	1	1	1	2	1	1	98	38,705,238	13,211	0.0	100.0
37	DA-LOW	EXA	1	2	1	4	3	1	96	74,473,109	25,420	0.0	100.0
38	DA-LOW	EXA	1	4	1	5	4	1	95	121,551,795	41,490	0.0	100.0
39	DA-LOW	EXA	1	8	1	5	3	1	95	171,804,526	58,642	0.0	100.0
40	DA-LOW	EXA	1	16	1	6	4	2	94	221,632,126	75,650	0.0	100.0
41	DA-LOW	EXA	1	24	1	5	3	2	95	246,288,356	84,066	0.0	99.8
42	DA-LOW	EXA	2	48	1	7	5	2	93	272,648,654	93,064	0.0	99.9

Fig. 5. Benchmark report for data analytics workloads using storage.

"Full table scans" cause sequential storage reads on the storage system and are usually limited by the bandwidth between the storage system and server system, in this case around 6 GBps (test 34). When two database servers request sequential reads, the storage system scales well (2 cluster nodes are used in test 35 and test 42).

Test 36 to 42 show the performance when SQL offload technology can be used. Even one database server can use the full performance capabilities of the storage system, which is by factors higher than on a conventional storage system. But this technology requires specialized hardware and software (Oracle Engineered System).

The benchmark report in Fig. 6 shows the same workload, but data is stored in main memory using row store (test 43–51) or column store (test 52–60). The results are very different (by factors) and allow a fair comparison of different technologies to calculate the price-/performance ratio of each data analytics solution.

Test	Workload	Loc	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]
43	DA-LOW	ROW	1	1	1	2	2	0	98	24,461,819	8,100	100.0	0.0
44	DA-LOW	ROW	1	2	1	4	4	0	96	46,686,828	15,459	100.0	0.0
45	DA-LOW	ROW	1	4	1	8	8	0	92	87,802,201	29,073	100.0	0.0
46	DA-LOW	ROW	1	8	1	17	16	0	83	76,070,044	25,188	100.0	0.0
47	DA-LOW	ROW	1	12	1	25	25	0	75	82,624,704	27,359	100.0	0.0
48	DA-LOW	ROW	1	16	1	33	33	0	67	113,429,349	37,559	100.0	0.0
49	DA-LOW	ROW	1	24	1	50	50	0	50	143,799,960	47,615	100.0	0.0
50	DA-LOW	ROW	1	32	1	66	66	0	34	145,733,328	48,255	100.0	0.0
51	DA-LOW	ROW	1	48	1	99	99	0	1	159,741,657	52,894	100.0	0.0

Test	Workload	Loc	Nodes	Jobs	DOP	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	Scanned user data [rps]	Scanned user data [MBps]	BuCache read [%]	FlCache read [%]
52	DA-LOW	COL	1	1	1	2	2	0	98	161,621,061	53,516	100.0	0.0
53	DA-LOW	COL	1	2	1	4	4	0	96	321,493,739	106,453	100.0	0.0
54	DA-LOW	COL	1	4	1	8	8	0	92	638,947,218	211,569	100.0	0.0
55	DA-LOW	COL	1	8	1	17	16	0	83	1,190,488,863	394,195	100.0	0.0
56	DA-LOW	COL	1	12	1	25	25	0	75	1,830,025,038	605,959	100.0	0.0
57	DA-LOW	COL	1	16	1	33	33	0	67	2,478,718,508	820,755	100.0	0.0
58	DA-LOW	COL	1	24	1	50	50	0	50	3,579,861,738	1,185,366	100.0	0.0
59	DA-LOW	COL	1	32	1	66	65	0	34	4,064,304,827	1,345,775	100.0	0.0
60	DA-LOW	COL	1	48	1	99	99	0	1	4,830,584,944	1,599,506	100.0	0.0

Fig. 6. Benchmark report for data analytics workloads using main memory.

4.5 Transaction Processing Workloads

System architects and capacity planners need performance metrics from database services regarding their ability to run typical transaction processing applications. Peakmarks provides transaction processing workloads of varying complexity (light, medium, and heavy) (Table 5).

Table 5. Peakmarks workloads to determine transaction processing performance.

Workload	Action	Key performance metric	Unit
TP-LIGHT	Select/Update single row via index. Example: account, product, order, invoice, etc.	• Transaction throughput • Response time	[tps] [ms]
TP-MEDIUM	Select/Update avg 25 rows via index. Example: account postings last week; item list of order, etc.	• Transaction throughput • Response time	[tps] [ms]
TP-HEAVY	Select/Update avg 125 rows via index. Example: last month's call records of smartphones, etc.	• Transaction throughput • Response time	[tps] [ms]
TP-MIXED	Complex workload: mix of Select/Update/Insert transactions.	• Transaction throughput • Response time	[tps] [ms]

The following benchmark report in Fig. 7 shows the performance of a database service for workloads TP-LIGHT and TP-HEAVY with 80/20 select/update ratio. The percentage of update transactions can be configured from 0% to 100% in 10% steps (column *Upd*).

Test	Workload	Upd [%]	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Trans total [tps]	Trans per cpu [tps]	Resp time [ms]	Log reads per tx [dbptx]	Phys reads per tx [dbptx]	Log writes per tx [dbptx]	Phys writes per tx [dbptx]	BuCache read [%]
61	TP-LIGHT	20	1	13	10	3	87	0	21,535	21,535	0.046	5.22	0.18	2.81	0.13	96.6
62	TP-LIGHT	20	2	20	17	3	80	0	29,990	14,995	0.066	5.22	0.14	2.81	0.15	97.2
63	TP-LIGHT	20	4	39	32	7	61	0	47,151	11,788	0.084	5.21	0.18	2.81	0.18	96.6
64	TP-LIGHT	20	8	65	53	13	35	0	64,295	8,037	0.123	5.21	0.24	2.81	0.28	95.4
65	TP-LIGHT	20	12	69	52	17	31	0	52,553	6,569	0.225	5.21	0.44	2.81	0.51	91.6
66	TP-HEAVY	20	1	14	12	2	86	0	657	657	1.516	281.37	3.73	301.79	5.19	98.7
67	TP-HEAVY	20	2	28	22	6	72	0	1,011	506	1.967	281.53	7.37	301.84	9.43	97.4
68	TP-HEAVY	20	4	49	41	8	51	0	1,620	405	2.446	281.29	6.35	301.71	12.78	97.7
69	TP-HEAVY	20	8	64	51	14	36	0	1,505	188	5.264	281.42	11.83	301.88	24.07	95.8
70	TP-HEAVY	20	12	46	24	22	54	0	551	69	21.525	282.31	43.06	302.12	60.28	84.7

Fig. 7. Benchmark report for transaction processing, 20% updates.

The key performance metrics for these workloads are transactions per second (column *Trans total*) and the response time (column *Resp time*).

The performance of these workloads depends on various factors, including the ratio of database size to buffer cache size. The higher the hit rate of the buffer cache (column *BuCache read*), the higher the transaction rate and the lower the response time of the transactions column *Resp time*). This is particularly true for low update rates, where the proportion of write operations is low in relation to the number of read operations.

The following benchmark report in Fig. 8 shows the performance of a database service with the workload "TP-MIXED" with 20% update share. This complex workload is similar to a TPC-C workload, where queries, update, and insert transactions are processed simultaneously.

Test	Workload	Upd [%]	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Trans total [tps]	Trans per cpu [tps]	Resp time [ms]	Log reads per tx [dbptx]	Phys reads per tx [dbptx]	Log writes per tx [dbptx]	Phys writes per tx [dbptx]	BuCache read [%]
71	TP-MIXED	20	4	45	36	9	55	0	13,326	3,331	0.298	21.57	0.48	20.59	0.66	97.8
72	TP-MIXED	20	8	77	60	17	23	0	17,242	2,155	0.460	23.60	0.85	22.84	1.39	96.4
73	TP-MIXED	20	12	80	61	19	20	0	13,440	1,680	0.879	29.35	1.22	29.22	2.44	95.8

Fig. 8. Benchmark report for mixed transaction processing, 20% updates.

4.6 PL/SQL Application Performance

PL/SQL is the preferred programming language for complex transaction logic and algorithms. PL/SQL code is stored in the database server. Some large applications, e.g., core banking systems, are entirely implemented in PL/SQL.

Peakmarks provides workloads to test PL/SQL code efficiency on a particular processor. These workloads are entirely CPU-bound. The key performance metrics for PL/SQL application performance are the number of executed PL/SQL operations within a certain timeframe and the execution time of PL/SQL algorithms (Table 6).

Table 6. Peakmarks workloads to determine transaction processing performance.

Workload	Action	Key performance metric	Unit
PLS-ADD	Addition of numbers.	Throughput of PL/SQL operations	[Mops]
PLS-BUILTIN	Datatype-specific operations, including SQL built-in functions, based on core banking and telco billing applications.	Throughput of PL/SQL operations	[Mops]
PLS-PRIME	Calculation of first N prime numbers.	Algorithm processing time	[s]
PLS-FIBO	Calculation of Fibonacci number N using a recursive algorithm.	Algorithm processing time	[s]
PLS-MIXED	Datatype-specific operations, including SQL built-in functions.	Throughput of PL/SQL operations	[Mops]

The following benchmark report in Fig. 9 shows the performance of a database service with the workload "PLS_MIXED" with different numerical datatypes.

```
                                       CPU  CPU  CPU  CPU      Operations  Operations  Elapsed
                                      busy user  sys idle           total     per cpu     time
Test Workload      Type    N Nodes  Jobs  [%]  [%]  [%]  [%]          [Mops]      [Mops]      [s]
---- ------------ ---- ---- ----- ----- ---- ---- ---- ---- --------------- ----------- --------
  74 PLS-MIXED       0    0     1     4   50   50    0   50          337.07       84.27      180
  75 PLS-MIXED       0    0     1     8   99   99    0    1          401.76       50.22      180
  76 PLS-MIXED       0    0     1    12  100   99    0    0          418.11       52.26      181
```

Fig. 9. Benchmark report for PL/SQL code with mixed datatypes.

4.7 Database Service Processes

In the case of Oracle, the performance of the log writer background process is critical. It is responsible for transaction logging and database recovery after system failures. The latency of transaction logging can have a significant impact on the response time of user transactions.

Optionally, the log writer is also used for database replication to synchronize standby databases. This technology is very popular for disaster recovery solutions. Replication can take place in synchronous or asynchronous mode. The data transfer between primary and standby databases can optionally be encrypted and/or compressed. With synchronous replication, local transactions have to wait until the standby databases have also applied the transaction log. This may delay local transaction processing considerably.

To analyze the performance behavior of the log writer process in all possible situations, Peakmarks offers two different workloads. One workload analyzes log writer latency, and the other one's log writer throughput (Table 7).

Table 7. Peakmarks workloads to determine log writer performance.

Workload	Action	Key performance metric	Unit
LGWR-LAT	Small insert transactions with 1, 25 or 125 rows per transaction and commit wait	• Transaction throughput • Response time • REDO sync time	[tps] [ms] [μsec]
LGWR-THR	Large insert transaction with 2'000 rows per transaction and commit wait	Log writer throughput	[MBps]

The following benchmark report in Fig. 10 shows the performance of a database service with the workload "LGWR-LAT" and different transaction sizes (column *TX size*) of 1 and 25 rows per transaction.

Test	Workload	TX size [rpt]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Response time [ms]	REDO blocks [rbps]	REDO data [MByte]	REDO sync [ms]	FlCache write [%]
77	LGWR-LAT	1	1	1	12	9	3	88	0	6,762	0.147	29,240	13	0.058	0.0
78	LGWR-LAT	1	1	2	24	17	7	76	0	11,293	0.176	49,077	21	0.070	0.0
79	LGWR-LAT	1	1	4	41	31	10	59	0	15,719	0.253	66,757	30	0.118	0.0
80	LGWR-LAT	1	1	8	65	52	13	35	0	27,992	0.285	114,960	53	0.138	0.0

Test	Workload	TX size [rpt]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Response time [ms]	REDO blocks [rbps]	REDO data [MByte]	REDO sync [ms]	FlCache write [%]
81	LGWR-LAT	25	1	1	12	9	3	88	0	1,270	0.785	70,581	35	0.242	0.0
82	LGWR-LAT	25	1	2	25	21	4	75	0	2,573	0.769	143,785	70	0.154	0.0
83	LGWR-LAT	25	1	4	46	40	7	54	0	4,267	0.925	239,056	117	0.192	0.0
84	LGWR-LAT	25	1	8	84	76	8	16	0	6,089	1.298	339,946	167	0.232	0.0

Fig. 10. Benchmark report for log writer latency.

The following benchmark report in Fig. 11 shows the performance of a database service with the workload "LGWR-THR" with large transactions.

Test	Workload	TX size [rpt]	Nodes	Jobs	CPU busy [%]	CPU user [%]	CPU sys [%]	CPU idle [%]	CPU iow [%]	Transactions total [tps]	Response time [ms]	REDO blocks [rbps]	REDO data [MByte]	REDO sync [ms]	FlCache write [%]
85	LGWR-THR	2000	1	1	14	12	3	86	0	115	8.631	115,085	57	0.740	0.0
86	LGWR-THR	2000	1	2	27	23	4	73	0	205	9.615	205,126	102	0.802	0.0
87	LGWR-THR	2000	1	4	52	48	4	48	0	321	12.390	322,406	160	0.596	0.0
88	LGWR-THR	2000	1	8	95	88	6	5	0	449	17.545	450,739	223	0.914	0.0

Fig. 11. Benchmark report for log writer throughput.

4.8 Order in Which Workloads Are Executed

We run the workloads in a logical order and start with the server and storage workloads. For example, if the storage workloads do not perform satisfactorily, other workloads that are heavily dependent on storage performance will also deliver disappointing results.

The following order has proven workable

- Workloads for Server Systems.
- Workloads for Storage Systems.
- Workloads for Database Background Processes.
- Workloads for Data Load.
- Workloads for Data Analytics.
- Workloads for Transaction Processing.
- Workloads for PL/SQL application programs.

5 Case Study

The Peakmarks benchmark software offers a fast and comprehensive performance analysis of Database Cloud Services. The results are understandable key performance metrics for representative database operations and provide a reliable foundation for price/performance comparisons and capacity planning.

Here is a summary of performance metrics of a database service with 8 processors, 32 GByte main memory, flash storage, and a 250 GByte database. The min/max values describe the system behavior in all load situations (best case, worst case).

The whole benchmark took less than 24 h. The customer selected those workloads from all that are important to him. The parameters for the workloads were chosen to reflect the customer's current environment best (Tables 8, 9, 10, 11, 12, 13 and 14).

Table 8. Peakmarks key performance metrics for server component.

Workload	Description	Cloud service A
SRV-MIXED	Min/Max query throughput per CPU in [qps]	12,458/17,673
SRV-SCAN	Min/Max scan throughput per CPU in [MBps]	2,118/3,986

[qps] queries per second
[MBps] megabytes per second

Table 9. Peakmarks key performance metrics for storage component.

Workload	Description	Cloud service A
STO-READ	Max sequential read throughput in [MBps]	2,014
STO-RANDOM 100% read	Max random read throughput in [dbps] at service time in [µs]	180,632 @ 387
STO-RANDOM 50% read	Max random read throughput in [dbps] at service time in [µs]	55,043 @ 546

[dbps] database blocks per second
[MBps] megabytes per second
[µs] microseconds

Table 10. Peakmarks key performance metrics for data load.

Workload	Description	Cloud service A
DL-BUFFER	Buffered data load, max data load rate in [MBps]	68
DL-DIRECT	Direct data load, max data load rate in [MBps]	136

[MBps] megabytes per second

Table 11. Peakmarks key performance metrics for data analytics.

Workload	Description	Cloud service A
DA-LOW default storage	Max data scan rate in [MBps]	1,755
DA-LOW storage offload	Max data scan rate in [MBps]	–
DA-LOW row store	Max data scan rate in [MBps]	16,828
DA-LOW column store	Max data scan rate in [MBps]	96,078

[MBps] megabytes per second

Table 12. Peakmarks key performance metrics for transaction processing (80% read, 20% update).

Workload	Description	Cloud service A
TP-LIGHT 1 row per tx	Max transaction rate in [tps] at response time in [ms]	73,0976 @ 107
TP-MEDIUM 25 rows per tx	Max transaction rate in [tps] at response time in [ms]	8,041 @ 971
TP-HEAVY 125 rows per tx	Max transaction rate in [tps] at response time in [ms]	1,775 @ 4,570
TP-MIXED	Max transaction rate in [tps] at response time in [ms]	20,924 @ 708

[tps] transactions per second
[ms] millisecond

Table 13. Peakmarks key performance metrics for PL/SQL application code.

Workload	Description	Cloud service A
PLS-ADD PLS_INTEGER	Min/Max throughput per CPU in [Mops]	167/381
PLS-BUILTIN NUMBER	Min/Max throughput per CPU in [Mops]	6.10/11.46
PLS-BUILTIN VARCHAR2	Min/Max throughput per CPU in [Mops]	1.98/4.38
PLS-MIXED all data types	Min/Max mixed operations per CPU in [Mops]	50.22/84.27

[Mops] million operations per second

Table 14. Peakmarks key performance metrics for database service processes.

Workload	Description	Cloud service A
LGWR-LAT 1 row per tx	Max log writer transaction rate in [tps] at service time in [µs]	27,992 @ 285
LGWR-LAT 25 rows per tx	Max log writer transaction rate in [tps] at service time in [µs]	6,089 @ 1,298
LGWR-LAT 125 rows per tx	Max log writer transaction rate in [tps] at service time in [µs]	1,583 @ 4,969
LGWR-THR	Max log writer throughput in [MBps]	223
DBWR-THR	Max database writer throughput in [dbps]	67,592

[Mops] million operations per second
[µs] microseconds
[tps] transactions per second
[dbps] database blocks per second

References

1. Bermbach, D. (ed.): Cloud Service Benchmarking. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55483-9_15
2. Crolotte, A.: Issues in benchmark metric selection. In: Nambiar, R., Poess, M. (eds.) TPCTC 2009. LNCS, vol. 5895, pp. 146–152. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10424-4_11
3. Huppler, K.: The art of building a good benchmark. In: Nambiar, R., Poess, M. (eds.) TPCTC 2009. LNCS, vol. 5895, pp. 18–30. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10424-4_3