

Benchmark Report of PostgreSQL Tests on AMD EPYC Rome and Milan

Version : 1.1
Author : Equinix Research Team
No. : 20223342/RPT/II/EQUINIX
Date : 06 April 2022

Copyright © 2022 PT Equinix Business Solutions. All rights reserved.

Copyrights

Copyright © 2007-2022 PT Equinix Business Solutions. All rights reserved.

This document is owned by PT Equinix Business Solutions. This document contains confidential information, which is protected by law. No part of this publication could be photocopied, reproduced or translated into another language without permitted or the express written consent of PT Equinix Business Solutions.

Data and information regarding the proposal and its offer is for limited use and are not disclosed. The information contained in this document is subject to change at any time without prior notice.

Document Modification

Version	Date	Comments / Changes	Name
1.0	19-08-2021	Initial Version	Equinix Research Team
1.1	06-04-2022	Revise Benchmark comparison results on dual sockets	Equinix Research Team

Table Of Contents

Copyrights	1
Document Modification	2
Table Of Contents	3
Executive Summary	4
Methodology	5
Construction	5
Software Installation	7
Benchmarking	7
Benchmark Result	10
Terminology	18
Software Terminology	18
RDBMS	18
PostgreSQL	18

Executive Summary

AMD EPYC has proven its ability in computing data intensive applications such as RDBMS. PostgreSQL as the most advanced and fastest Database Management System. It delivers tremendous results, compared to any dual socket servers in TPC-B and TPC-C Benchmark. TPC (Transaction Performance Council) is the world council dedicated as the ultimate reference for RDBMS benchmark (<https://tpc.org>)

Benchmarking PostgreSQL on top of AMD Zen 3 7543 has resulted in more than 3,021 TPS per Core for TPC-B, equivalent to 193,344 TPS for 64 core AMD EPYC processors. For TPC-C, it resulted in 45,952 TPM per Core, equivalent to 2,940,928 TPM for 64 core AMD EPYC processors.

On TPC-B, compared with AMD Zen 2 7742 performance, for a single processor, 7543 has 28% improvement. This improvement can be achieved as easily as replacing the socket to Zen 3 one. Meanwhile on TPC-C, 7543 has 48% improvement.

We do not recommend using dual sockets both on Zen 2 and Zen 3 in PostgreSQL, since it decreases the performance per core basis by around 64% for Zen 2 and 40% for Zen 3. In the other words, the performance using dual socket does not increase any significant improvement: 2% on Zen 2 and 18% on Zen 3.

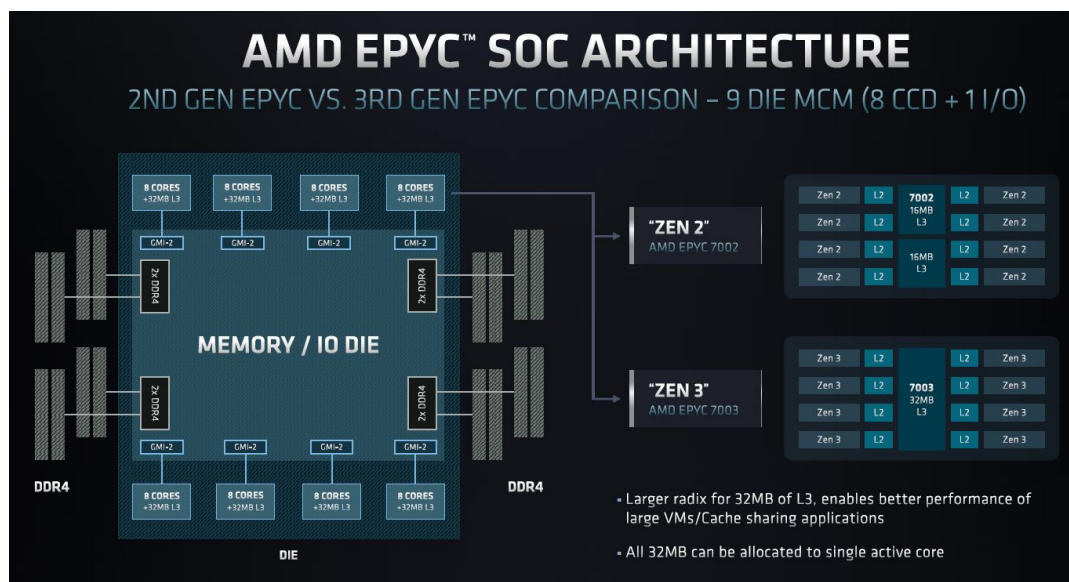
AMD EPYC Zen 3 is the most recommended processor for High Performance Transaction System. This benchmark has already proven its ability the magic of Infinity Fabric, which is able to synchronize all cores to run in synergy. It answers almost all business needs for a High Performance Database System. For OLTP requirement, a great number of concurrency processes is a must feature to cater high loads of increasing user or customer accessing the system at the same time.

Methodology

Benchmarking a system has to be done by using proper mechanism or methodology. Improper usage of testing clearly results in nothing useful. The expected result should allow us to extrapolate in some manner to the real world condition and give us proper understanding or help us visualize the situation comparable to the production environment.

a. Construction

In performing the test, there are 2 configurations used on the AMD 7742 and AMD 7543 configuration as a Server. Below is the server configuration with 1 processor on each server.



Hardware Specification:

No	Type	Specification	
1.	AMD 7742	Machine Type	GigaByte R272-Z30
2.		Processor	AMD EPYC 7742 64-Core Processor (128HT)
3.		Memory	512 GB RAM
4.		Disk 1	2 TB NVME

5.		Operating System	Debian GNU/Linux 10 (buster)
1.	AMD 7543	Machine Type	GigaByte R272-Z30
2.		Processor	AMD EPYC 7543 32-Core Processor (64HT)
3.		Memory	512 GB RAM
4.		Disk 1	2 TB NVME
5.		Operating System	Debian GNU/Linux 10 (buster)

Table 1: Hardware Specifications Single Socket

Below is the server configuration with 2 processor socket on each server.

No	Type	Specification	
1.	AMD 7742	Machine Type	GigaByte R282-Z90
2.		Processor	2x AMD EPYC 7742 64-Core Processor (256HT)
3.		Memory	512 GB RAM
4.		Disk 1	2 TB NVME
5.		Operating System	Debian GNU/Linux 10 (buster)
1.	AMD 7543	Machine Type	GigaByte R282-Z90
2.		Processor	2x AMD EPYC 7543 32-Core Processor (128HT)
3.		Memory	512 GB RAM
4.		Disk 1	2 TB NVME
5.		Operating System	Debian GNU/Linux 10 (buster)

Table 2: Hardware Specifications Dual Socket

Software involved in the test:

1. Debian GNU/Linux 10 (buster) (64-bit)
2. Database Server 11DB

3. pgBench for TPC-B tests.
4. HammerDB for TPC-C tests.

b. Software Installation

On Client Side, installs:

1. pgBench compile from source code
2. HammerDB, requires X Windows System.
3. some other utility tools such as: ntop, htop, iotop, etc.

On Server Side, installs:

1. 11DB from Source Code Equinix and compiled with GCC.

c. Benchmarking

There are 2 tools use for benchmark:

1. pgBench

Use specifically for PostgreSQL benchmark with TPC-B scenario, TPC-B methodology is explained in the Terminology Section below. There are 3 parameters that determine the load of the test. This parameter should be set initially before every test, unless the next test is using the same parameter.

1. **Scale** is the parameter for setting numbers of records on the database to be tested.
2. **Endurance** is the number of second that test will passed. Higher number of the endurance affects the growing of CPU and memory usages, and at some point, it will consume all of the resources when the load is great enough. More usage of utilization of the CPU shows more effective the result comparing to the real world condition.
3. **Concurrency**, determined number of simultaneous processes at the same time. It has linear relationship between concurrency and number of the Hardware Thread, by a factor of a constant number.

pgBench initialization with parameters:


```
#pgBench -i -s 128 -h [host_server] -p [port_number] -U [username] [dbname]
```

-i is a parameter to perform initialization

-s is a parameter for determining the scale. 1 unit of scaling is 10,000 table record.

-h is a parameter to indicate the host address of the server.

-p is a parameter to indicate the port number of the server.

-U is the parameter to specify a user that will be used.

Running pgBench with parameter:

```
#pgBench -T [duration] -c [concurrency] -p [port_number] \  
-h [host_address] -U [username] [dbname]
```

-T is a parameter to specify the duration in the second. In this case use the standard 600 second.

-c is a parameter to specify the value of concurrency to be used.

-p is a parameter to indicate the port number of the server.

-h is a parameter to indicate the host address of the server.

-U is a parameter to specify a user that will be used.

2. HammerDB

Use generally for benchmark with TPC-C scenario, TPC-C methodology is explained in the Terminology Section below. HammerDB was developed as Open Source and support many databases to be tested.

1. **Endurance** is the number of how long the benchmark will last, a similar parameter to the point 1. This number can be set by some number in seconds to determine capability of the database crunching high load at peak time.
2. **Concurrency**, the number of the simultaneous process of the database to cater high load which comes at the same time, that mimic the real world situation.

Parameter usage scope is:

- Number of Virtual User : 64 - 256
- Number of Warehouse : 64 - 256

-
- Duration of test : 30 minutes

Benchmark Result

1. TPC-B Benchmark comparison with 1 processor socket

TPC-B Result AMD Zen 3 7543 (64)

No	Scale	Concurrency	Thread	Duration (s)	fsync	Type	QPS
1	64	64	64	600	on	RW	91,132
2	64	64	64	600	on	R/O	889,504
3	64	64	64	600	off	RW	88,626
4	64	64	64	600	off	R/O	870,751
5	128	128	128	600	on	RW	96,681
6	128	128	128	600	on	R/O	854,046
7	128	128	128	600	off	RW	87,655
8	128	128	128	600	off	R/O	847,835

TPC-B Result AMD Zen 2 7742 (128)

No	Scale	Concurrency	Thread	Duration (s)	fsync	Type	QPS
1	64	64	64	600	on	RW	93,158
2	64	64	64	600	on	R/O	1,133,981
3	64	64	64	600	off	RW	29,038
4	64	64	64	600	off	R/O	1,134,455
5	128	128	128	600	on	RW	103,756
6	128	128	128	600	on	R/O	1,572,873
7	128	128	128	600	off	RW	95,771
8	128	128	128	600	off	R/O	1,573,100

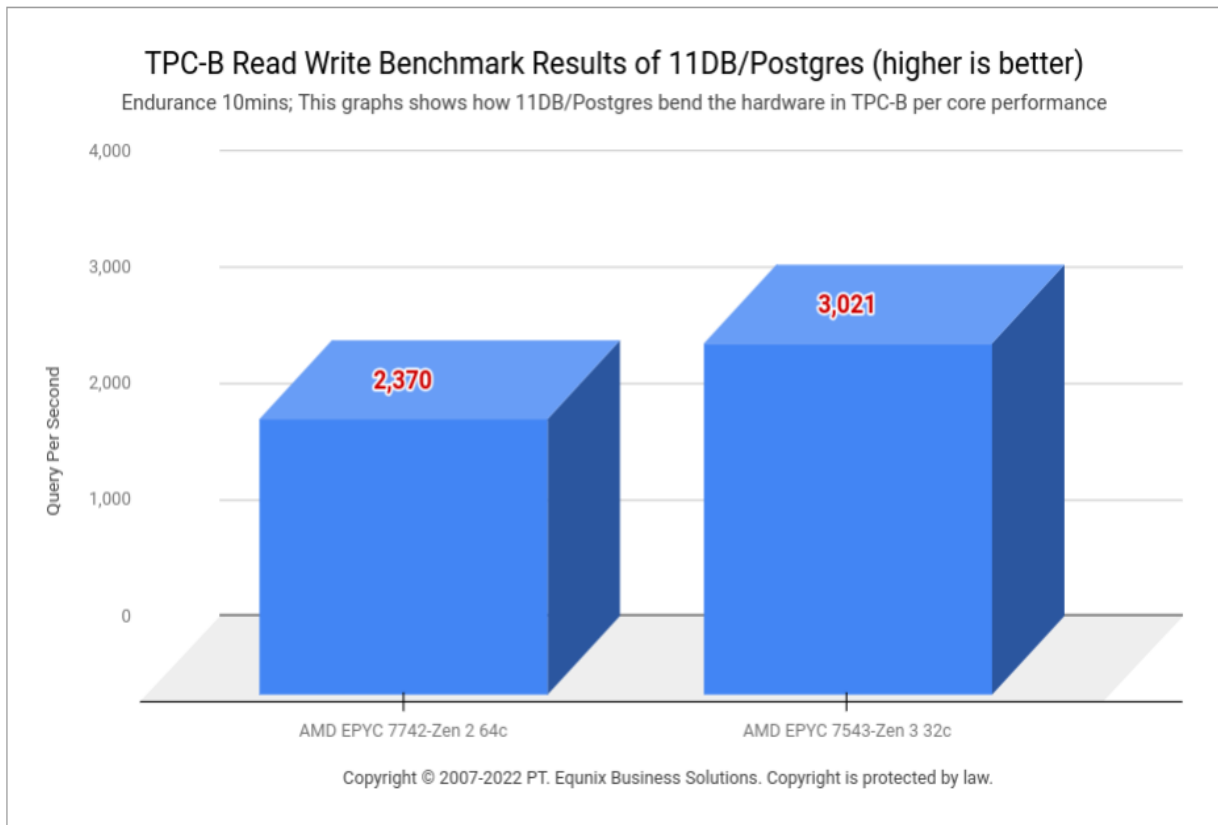


Illustration 1: Comparison TPC-B AMD Zen 2 and Zen 3 Per Core (Read Write)

Illustration 1 shows that PostgreSQL which is run on AMD Zen 3 7543 configuration has an improvement compared to AMD Zen 2 7742. On AMD Zen 2 7742, the optimum value is **2,370** QPS Per Core (151684 QPS / 64 Core) with concurrency 128. While on AMD Zen 3 7543, the optimum value is **3,021** QPS Per Core (96680 QPS / 32 Core) with concurrency 128.

$$1 + \frac{3,021 - 2,370}{2,370} = 1,28x$$

It shows about a **28%** increase in performance.

2. TPC-C Benchmark comparison with 1 processor socket

TPC-C Result AMD Zen 3 7543 (64)

No	Warehouses	User	Duration (s)	NOPM	TPM	fsync
1	64	32	30	615,320	1,415,076	off
2	64	64	30	604,972	1,426,019	off
3	64	32	30	608,224	1,403,205	on
4	64	64	30	602,867	1,394,219	on
5	128	64	30	631,345	1,452,364	off
6	128	128	30	628,794	1,448,582	off
7	128	64	30	639,469	1,470,477	on
8	128	128	30	596,069	1,373,997	on

TPC-C Result AMD Zen 2 7742 (128)

No	Warehouses	User	Duration (s)	NOPM	TPM	fsync
1	128	128	30	817,616	1,879,224	on
2	128	128	30	832,505	1,908,743	on
3	128	128	30	863,906	1,979,870	on
4	128	128	30	760,141	1,746,491	on

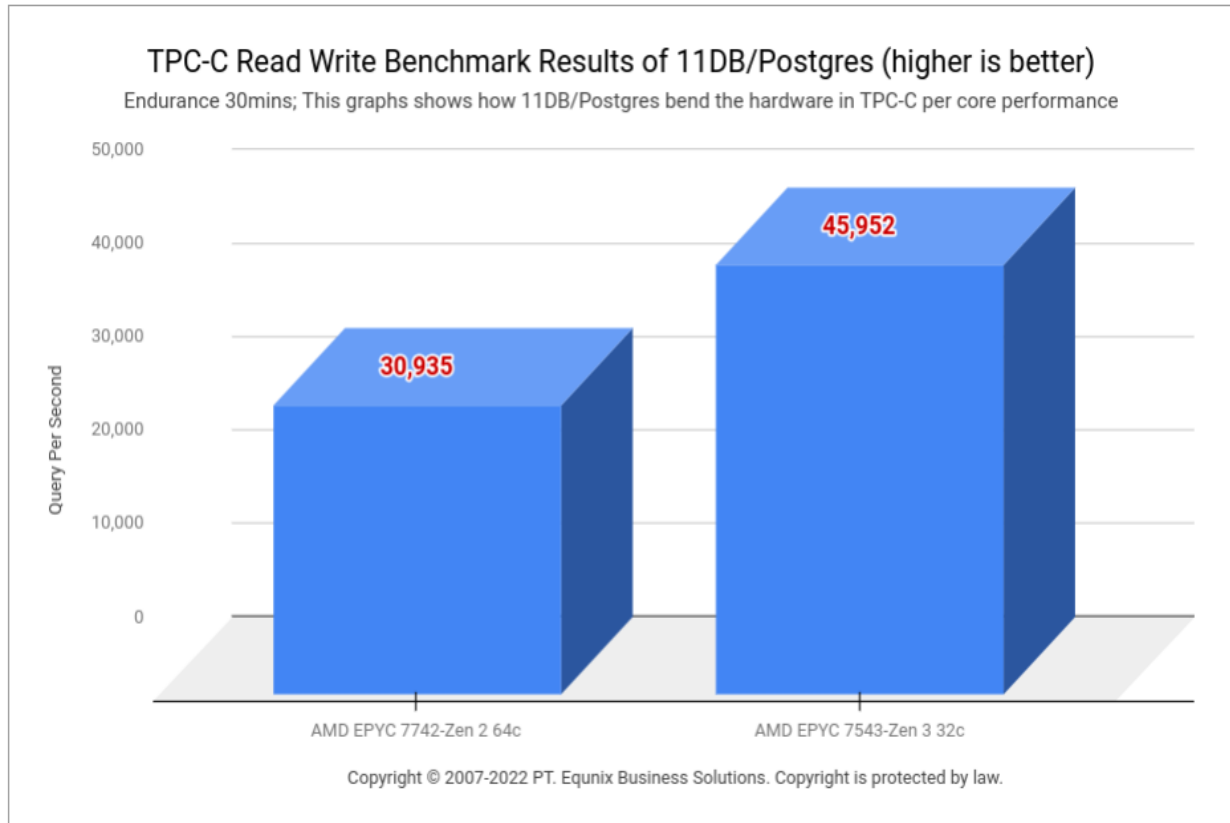


Illustration 2: Comparison TPC-C AMD Zen 3 32 Core and Zen 3 64 Core (Read Write)

Illustration 2 shows that PostgreSQL which is run on AMD Zen 3 7543 configuration has an improvement compared to AMD Zen 2 7742. On AMD Zen 2 7742, the optimum value is **30,935** QPS Per Core (1,979,870 QPS / 64 Core) with concurrency 128. While on AMD Zen 3 7543, the optimum value is **45,952** QPS Per Core (1,470,477 QPS / 32 Core) with concurrency 128.

$$1 + \frac{45,952 - 30,935}{30,935} = 1,48x$$

It shows about a **48%** increase in performance.

3. TPC-B Benchmark comparison with 2 processor sockets

TPC-B Result AMD Zen 3 7543 (128)

No	Scale	Concurrency	Thread	Duration (s)	fsync	Type	QPS
1	128	128	128	600	on	RW	102,958
2	128	128	128	600	on	R/O	1,396,416
3	128	128	128	600	off	RW	96,774
4	128	128	128	600	off	R/O	1,331,819
5	256	256	256	600	on	RW	114,607
6	256	256	256	600	on	R/O	1,376,968
7	256	256	256	600	off	RW	91,589
8	256	256	256	600	off	R/O	1,465,377

TPC-B Result AMD Zen 2 7742 (256)

No	Scale	Concurrency	Thread	Duration (s)	fsync	Type	QPS
1	128	128	128	600	on	RW	86,639
2	128	128	128	600	on	R/O	1,153,687
3	128	128	128	600	off	RW	107,938
4	128	128	128	600	off	R/O	1,145,422
5	256	256	256	600	on	RW	105,752
6	256	256	256	600	on	R/O	1,601,135
7	256	256	256	600	off	RW	124,977
8	256	256	256	600	off	R/O	1,576,313

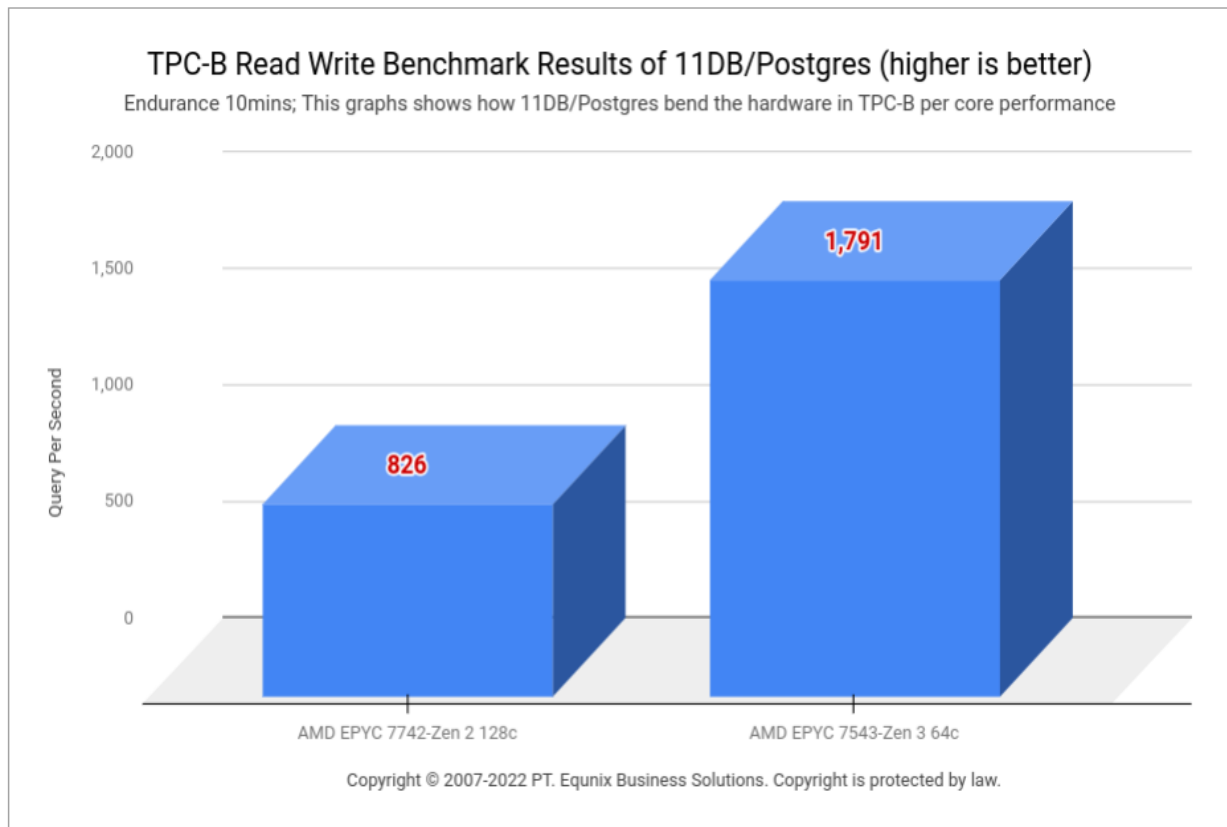


Illustration 3: Comparison TPC-C AMD Zen 3 32 Core and Zen 3 64 Core (Read Write)

Illustration 3 shows that PostgreSQL which is run on AMD Zen 3 7543 configuration has an improvement compared to AMD Zen 2 7742. On AMD Zen 2 7742, the optimum value is **826** QPS Per Core (105,752 QPS / 128 Core) with concurrency 256. While on AMD Zen 3 7543, the optimum value is **1791** QPS Per Core (114,607 QPS / 64 Core) with concurrency 256.

$$1 + \frac{1791-826}{826} = 2,16x$$

It shows about **116%** increase in performance.

Here is result comparison TPC-B AMD 7543 Zen 3 32 Core and 64 Core:

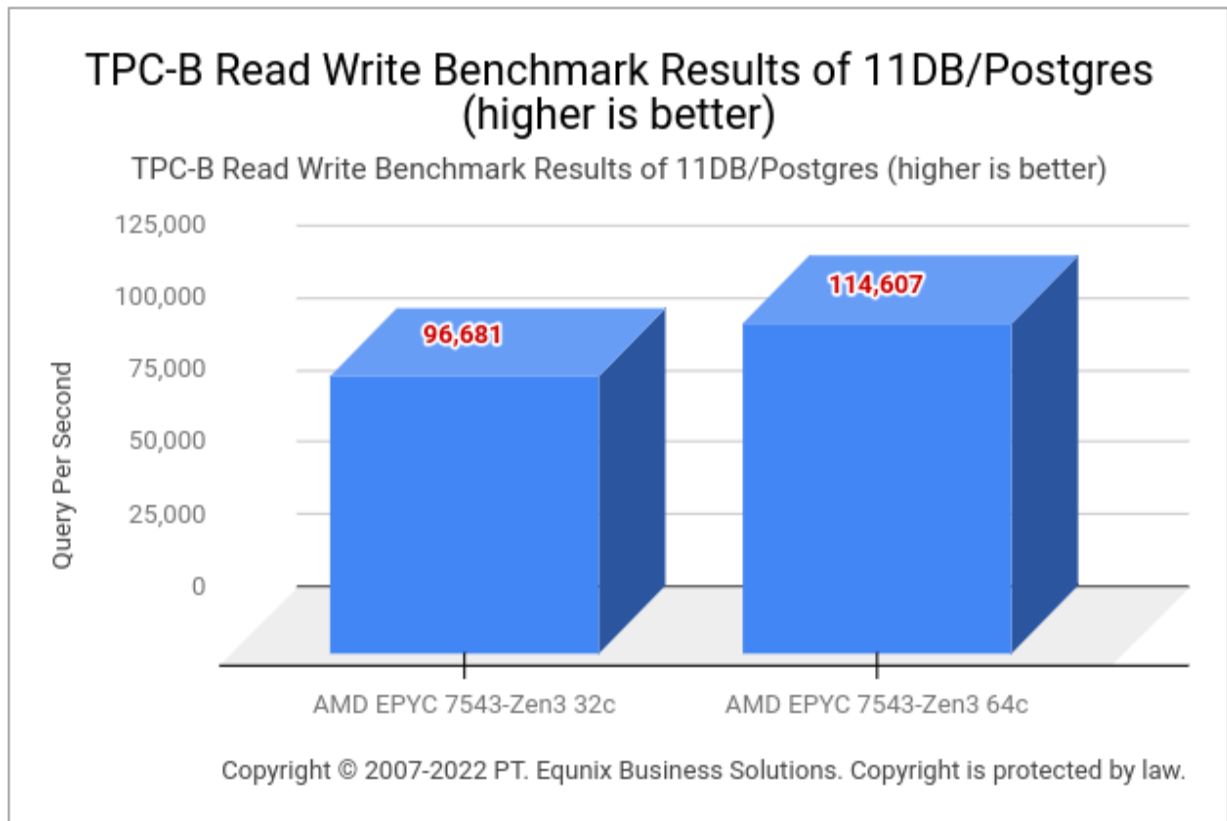


Illustration 4: Comparison TPC-B AMD Zen 3 32 Core and Zen 3 64 Core (Read Write)

Illustration 4 shows that PostgreSQL which is run on AMD Zen 3 7543 32 Core configuration has an improvement compared to AMD Zen 3 7543 64 Core. On AMD Zen 3 7543 64 Core, the optimum value is **114,607** QPS with concurrency 128. While on AMD Zen 3 7543 32 Core, the optimum value is **96,681** QPS with concurrency 128.

$$1 + \frac{114,607 - 96,681}{96,681} = 1,18x$$

It shows about an **18%** increase in performance.

```

1 [|||||] 87.8%] 17 [|||||] 87.6%] 33 [|||||] 87.5%] 49 [|||||] 87.4%]
2 [|||||] 86.9%] 18 [|||||] 86.9%] 34 [|||||] 87.2%] 50 [|||||] 87.7%]
3 [|||||] 87.3%] 19 [|||||] 87.8%] 35 [|||||] 87.2%] 51 [|||||] 87.9%]
4 [|||||] 87.7%] 20 [|||||] 87.1%] 36 [|||||] 87.7%] 52 [|||||] 87.7%]
5 [|||||] 87.6%] 21 [|||||] 88.2%] 37 [|||||] 87.3%] 53 [|||||] 87.2%]
6 [|||||] 87.8%] 22 [|||||] 88.0%] 38 [|||||] 87.1%] 54 [|||||] 86.8%]
7 [|||||] 87.5%] 23 [|||||] 87.4%] 39 [|||||] 87.4%] 55 [|||||] 88.3%]
8 [|||||] 87.2%] 24 [|||||] 87.3%] 40 [|||||] 87.3%] 56 [|||||] 88.1%]
9 [|||||] 87.8%] 25 [|||||] 87.7%] 41 [|||||] 88.0%] 57 [|||||] 88.2%]
10 [|||||] 88.9%] 26 [|||||] 87.9%] 42 [|||||] 87.9%] 58 [|||||] 87.1%]
11 [|||||] 87.1%] 27 [|||||] 87.9%] 43 [|||||] 87.0%] 59 [|||||] 88.0%]
12 [|||||] 88.2%] 28 [|||||] 87.2%] 44 [|||||] 88.2%] 60 [|||||] 87.2%]
13 [|||||] 88.0%] 29 [|||||] 87.3%] 45 [|||||] 87.0%] 61 [|||||] 87.4%]
14 [|||||] 88.6%] 30 [|||||] 87.2%] 46 [|||||] 87.9%] 62 [|||||] 86.9%]
15 [|||||] 87.2%] 31 [|||||] 88.1%] 47 [|||||] 88.4%] 63 [|||||] 87.4%]
16 [|||||] 87.0%] 32 [|||||] 87.4%] 48 [|||||] 88.8%] 64 [|||||] 87.3%]

progress: 39.0 s, 98464.3 tps, lat 1.298 ms stddev 0.721
progress: 40.0 s, 93730.7 tps, lat 1.368 ms stddev 0.815
progress: 41.0 s, 90962.2 tps, lat 1.407 ms stddev 0.668
progress: 42.0 s, 104751.9 tps, lat 1.222 ms stddev 0.609
progress: 43.0 s, 105240.9 tps, lat 1.216 ms stddev 0.593
progress: 44.0 s, 105812.2 tps, lat 1.210 ms stddev 0.635
progress: 45.0 s, 105412.0 tps, lat 1.214 ms stddev 0.640
progress: 46.0 s, 104749.0 tps, lat 1.222 ms stddev 0.611
progress: 47.0 s, 104419.2 tps, lat 1.226 ms stddev 0.606
progress: 48.0 s, 102925.8 tps, lat 1.244 ms stddev 0.601
progress: 49.0 s, 103416.7 tps, lat 1.237 ms stddev 0.624
progress: 50.0 s, 103655.3 tps, lat 1.235 ms stddev 0.631
progress: 51.0 s, 105104.4 tps, lat 1.218 ms stddev 0.609

Every 1.0s: cat /proc/cpuin... GoldGriz: Fri Jun 25 14:58:26 2021
cpu MHz      : 2799.802
cpu MHz      : 2799.789
cpu MHz      : 2799.566
cpu MHz      : 2798.961
cpu MHz      : 2799.750

64 fsync on RW= 91132.091295
64 fsync on RO = 889503.733774
128 fsync on RW = 91589.778690
128 fsync on RO =

-- INSERT --                               3,31      All

1 [ ] 0.0%] 17 [ ] 0.7%] 33 [ ] 0.0%] 49 [ ] 0.0%]
2 [ ] 0.0%] 18 [ ] 0.0%] 34 [ ] 0.0%] 50 [ ] 0.0%]
3 [ ] 0.0%] 19 [ ] 0.0%] 35 [ ] 0.0%] 51 [ ] 0.0%]
4 [ ] 0.0%] 20 [ ] 0.0%] 36 [ ] 0.0%] 52 [ ] 0.0%]
5 [ ] 0.0%] 21 [ ] 0.0%] 37 [ ] 0.0%] 53 [ ] 0.0%]
6 [ ] 0.0%] 22 [ ] 0.0%] 38 [ ] 0.0%] 54 [ ] 0.0%]
7 [ ] 0.0%] 23 [ ] 0.0%] 39 [ ] 0.0%] 55 [ ] 0.0%]
8 [ ] 0.7%] 24 [ ] 0.0%] 40 [ ] 0.0%] 56 [ ] 0.0%]
9 [ ] 0.0%] 25 [ ] 0.0%] 41 [ ] 0.0%] 57 [ ] 0.0%]
10 [ ] 0.0%] 26 [ ] 0.0%] 42 [ ] 0.7%] 58 [ ] 0.0%]
11 [ ] 0.0%] 27 [ ] 0.0%] 43 [ ] 0.0%] 59 [ ] 0.0%]
12 [ ] 0.0%] 28 [ ] 0.0%] 44 [ ] 0.0%] 60 [ ] 0.0%]
13 [ ] 0.0%] 29 [ ] 0.0%] 45 [ ] 0.0%] 61 [ ] 0.0%]
14 [ ] 0.0%] 30 [ ] 0.0%] 46 [ ] 0.0%] 62 [ ] 0.0%]
15 [ ] 0.0%] 31 [ ] 0.0%] 47 [ ] 0.0%] 63 [ ] 0.0%]
16 [ ] 0.0%] 32 [ ] 0.0%] 48 [ ] 0.0%] 64 [ ] 0.0%]

progress: 591.0 s, 103881.6 tps, lat 1.232 ms stddev 0.604
progress: 592.0 s, 103172.0 tps, lat 1.240 ms stddev 0.750
progress: 593.0 s, 103155.4 tps, lat 1.241 ms stddev 0.637
progress: 594.0 s, 103730.6 tps, lat 1.234 ms stddev 0.612
progress: 595.0 s, 101955.3 tps, lat 1.255 ms stddev 0.673
progress: 596.0 s, 103977.4 tps, lat 1.231 ms stddev 0.607
progress: 597.0 s, 104178.3 tps, lat 1.228 ms stddev 0.596
progress: 598.0 s, 104661.8 tps, lat 1.223 ms stddev 0.655
progress: 599.0 s, 103394.1 tps, lat 1.238 ms stddev 0.621
progress: 600.0 s, 103885.8 tps, lat 1.232 ms stddev 0.614
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 128
query mode: simple
number of clients: 128
number of threads: 128
duration: 600 s
number of transactions actually processed: 58010795
latency average = 1.324 ms
latency stddev = 0.803 ms
tps = 96679.445338 (including connections establishing)
tps = 96680.624755 (excluding connections establishing)
equinix@GoldGriz:~$

Every 1.0s: cat /proc/cpuin... GoldGriz: Fri Jun 25 15:11:20 2021
cpu MHz      : 1981.087
cpu MHz      : 2137.877
cpu MHz      : 2130.244
cpu MHz      : 1980.933
cpu MHz      : 1992.340
cpu MHz      : 2115.302
cpu MHz      : 1813.835
cpu MHz      : 2230.054
cpu MHz      : 1796.064
cpu MHz      : 1794.869
cpu MHz      : 1794.236
cpu MHz      : 1795.766
cpu MHz      : 1797.461

64 fsync on RW= 91132.091295
64 fsync on RO = 889503.733774
128 fsync on RW = 91589.778690
128 fsync on RO =

-- INSERT --                               3,31      All

```

Illustration 2: CPU Usage of AMD Zen 3 75432 on TPC-B Bench

On AMD Zen 3 7543 configuration, when PostgreSQL is benchmarked, all CPU have more than 87% load. This describes that all CPU is used maximally, so PostgreSQL can achieve a huge performance.

Terminology

This chapter explains the definition and explanation of the term used in this document.

A. Software Terminology

i. RDBMS

RDBMS (Relational Database Management System) is an application for managing data in a structured and relational manner. The white-paper of Relational Data was invented by IBM Researcher Edgar F. Codd in 1970 as the foundation architecture of RDBMS today.

As RDBMS, there are 4 basic rules that should be complied with, Atomicity, Confidentiality, Isolation and Durable. These rules ensure the database can be fully functional on the transaction system. RDBMS has a Database Container which can contain: (1) Schemas, (2) Tables, (3) Triggers, (4) Stored Procedures, (5) Sequences, (6) Records, etc.

ii. PostgreSQL

PostgreSQL was derived from Ingres and Informix which was developed by UCB at 1970's up to 1990's led by Prof. Michael Stonebraker. In 1996, PostgreSQL Global Development Group has founded and became the major Developer of PostgreSQL until now. It is formed from many PostgreSQL Hacker all around the world. PostgreSQL World Conference is held every year in Ottawa, Canada in May. Equinix Business Solutions participated in the Conference as the Gold Sponsor 2014 as the contribution for the community of the Developers.

PostgreSQL has compiled 160 of 179 Conformance of SQL: 2011. The most conforming SQL database ever, with the new features of JSONB support in 2014, now PostgreSQL has the most features and joining the best of the both worlds, NOSQL and SQL.

Support for NoSQL or unstructured data meanwhile supporting Transactional and Structured data is the most awaited feature so far for many application which want flexibility as good as performance of financial related transactional.