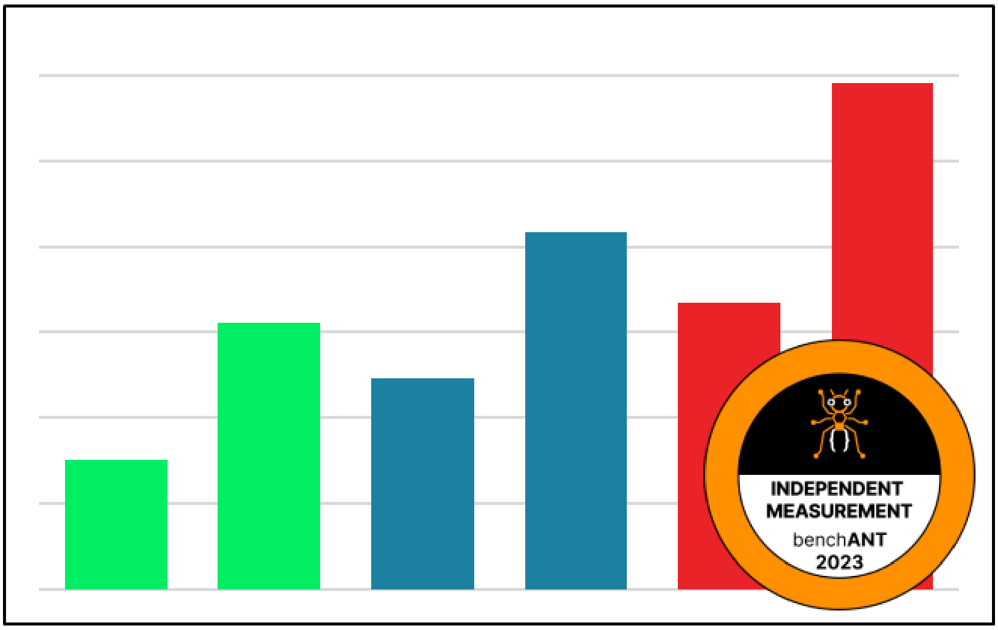
117 Independent Database and Cloud Performance Benchmarks

All the Things Database and Cloud Providers won't tell you!

By **The benchANT Team**

- *Database & Cloud Performance Engineering & Consulting–*



Introduction – Why Performance Matters!

Welcome to a groundbreaking whitepaper that unveils the untold truths behind database and cloud performance benchmarks. In an era dominated by data-driven decision making and cloud adoption, understanding the intricate details of database and infrastructure performance is crucial for organizations seeking to gain a competitive edge. This whitepaper, based on real-world project experience, will empower you with valuable, untold insights.

Enter performance benchmarks—a scientific approach to measuring, comparing, and evaluating the performance of different databases and cloud platforms. These benchmarks act as a compass, guiding you towards informed decision-making and ensuring optimal utilization of your resources.

While database vendors and cloud providers entice you with claims of unparalleled performance and linear scalability, they conveniently omit critical information that could impact your business's efficiency and cost-effectiveness. This whitepaper is designed to expose these hidden truths, empowering you with knowledge that will level the playing field.

117 Comprehensive Benchmarks: Empowering You with Real-World Data

Drawing from a vast array of real-world projects, this whitepaper features an astounding 117 independent performance benchmarks. Each benchmark scrutinizes a specific aspect of database and cloud performance, providing you with practical insights and real-world examples. These benchmarks cover a wide range of crucial parameters, including latency, throughput, scalability, costs, and performance/cost ratios.

By leveraging the power of these benchmarks, you can make data-driven decisions that align with your business goals. Rather than relying on marketing hype or biased vendor claims, you can understand the inhomogeneity of database and cloud infrastructure products and learn how to measure what is important.

Knowledge is power, and the insights contained within these benchmarks will equip you with the information necessary to make intelligent, evidence-based decisions. Embrace transparency, demand performance, and position your tech stack for the growing amount of data in the front position.

Let's dive in.

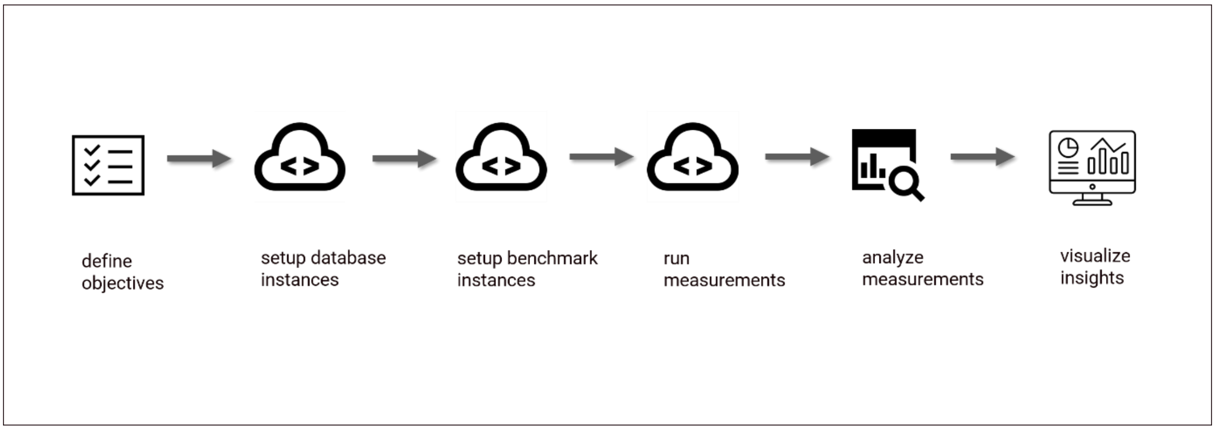
Daniel

Table of Contents

1. Database Performance – First steps

Database performance is a selling point since the rise of database management systems in 1990. There is nearly no database producer who is not doing marketing with words like "fastest", "scaling" or "real-time". Also, nearly every vendor is publishing performance benchmark reports, done by their own engineering team or third parties.

At benchANT, our mission is to bring independent and reliable performance data for databases, Database-as-a-Service products, and cloud resources to everyone. Our approach is based on a scientifically approved benchmarking methodology and toolset and enables reliable and efficient measurement automation. More information about benchANT and our benchmarking process is available on our website (https://benchant.com/).



The most important things when doing database benchmarks are:

- Relevance
- Reproducibility
- Fairness
- Verifiability
- Usability

In the following chapters we show you a large amount of highly interesting information for the most important KPIs for database performance measurement. Based on that data, we explain why and what is important.

And sometimes, you will see surprising results.

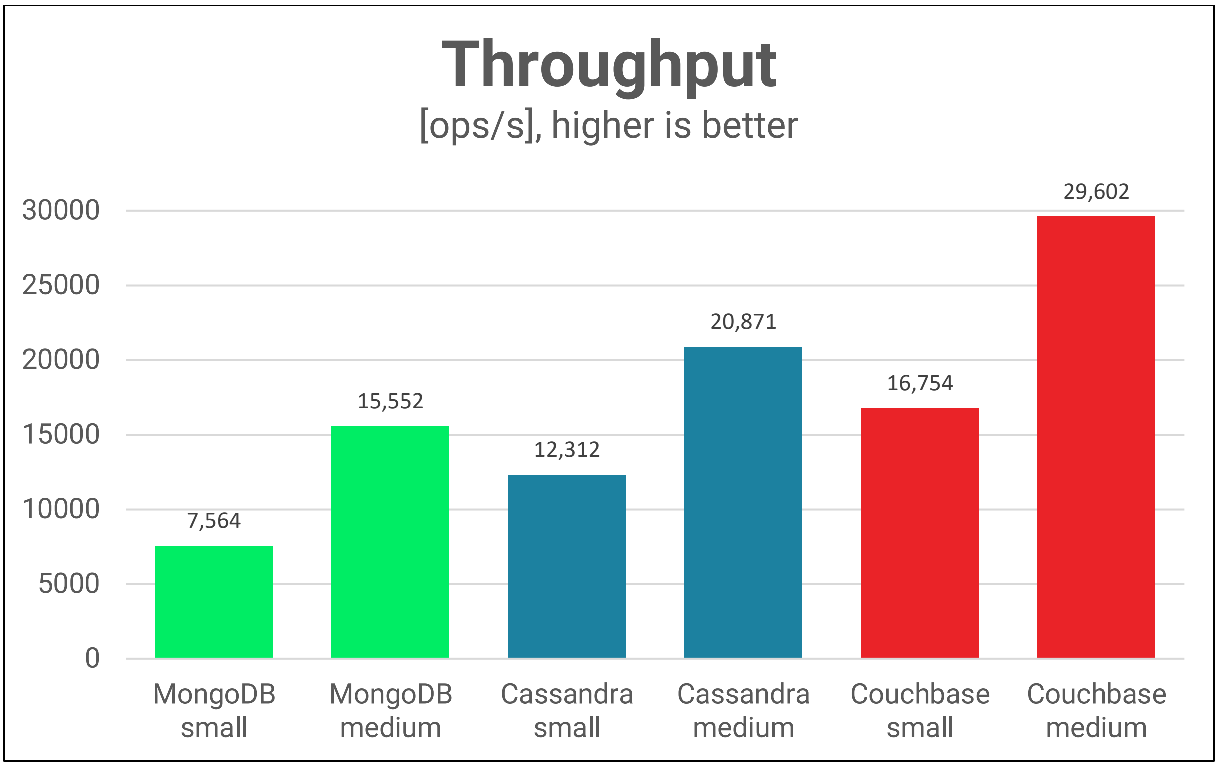
Be curious if the DBMS and Cloud provider can keep their promises!

1.1. Throughput on the Data Autobahn – Popular NoSQL Databases

Data-intense application need to handle not only a huge amount of stored data, but also many database operations per second to store and read data. This KPI, the so-called **Throughput**, is measured in database operations (or transactions) per second.

Let's compare the throughput of some modern popular NoSQL databases with a simple CRUD workload when hosted on AWS EC2.

Target Technology	NoSQL databases on Cloud infrastructure
Databases	a) MongoDB CE v5 b) Apache Cassandra v4 c) Couchbase Server CE v7
Infrastructure	AWS EC2
Scaling	a) small: m5.large (2 vCPUs, 8 GB RAM, single-node) b) medium: m5.xlarge with doubled workload threads
Workloads	YCSB: 50% read , 50% write, simple operations (no joins, no aggregations, simple search)



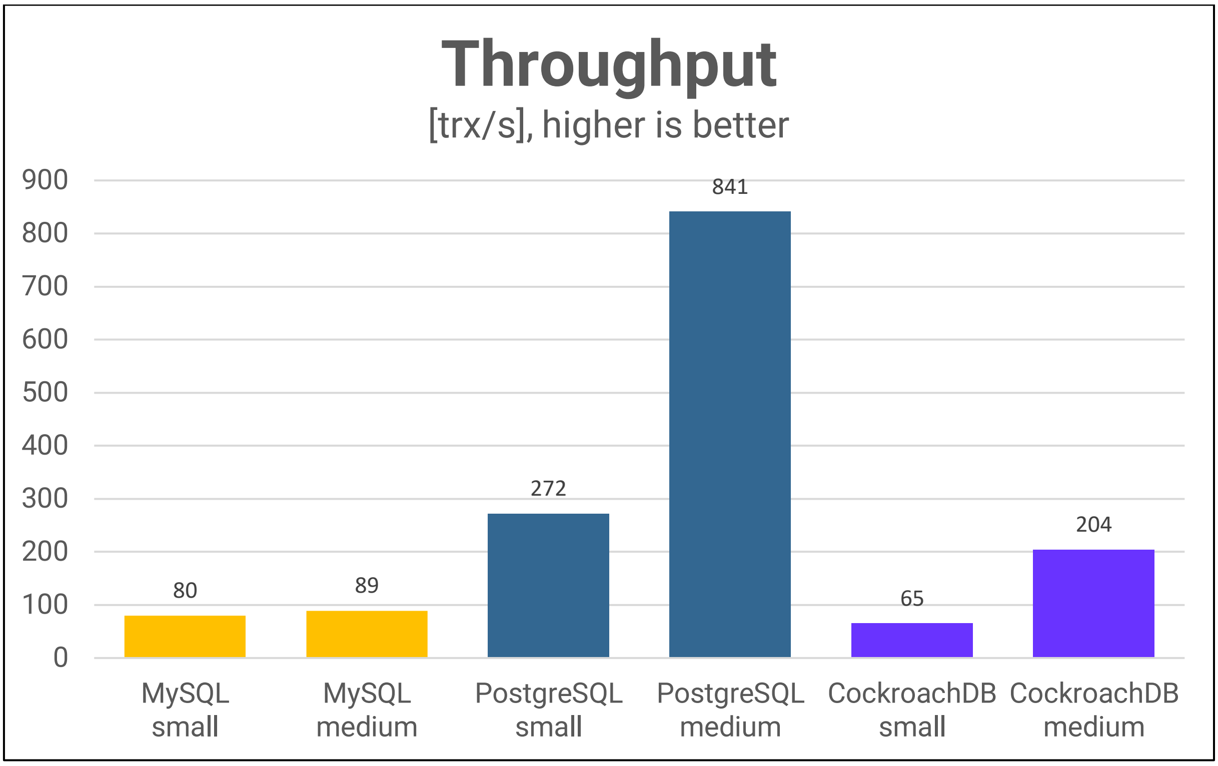
- ➔ Database technologies show up a wide throughput range: best is twice as high as last) for same workload.
- ➔ Scaling-up infrastructure resources rises the ability to handle more operations.
- ➔ Knowledge of maximum throughput is essential for database management.

1.2. Transactional Throughput for Relational Databases

For relational databases the throughput is often measured in transactions per second/per hour. Hereby, multiple database operations are bundled into database transactions.

Let's compare the throughput of some open-source relational databases for a more complex and transactional workload.

Target Technology	Relational databases on Cloud infrastructure
Databases	a) PostgreSQL v13 b) MySQL v8 c) Cockroach v21
Infrastructure	AWS EC2
Scaling	a) small: m5.large (2 vCPUs, 8 GB RAM, single-node) b) medium: m5.xlarge with doubled workload threads
Workloads	Sysbench 1.0: OLTP Mix with non-simple operations, grouped to transactions, no batch processing



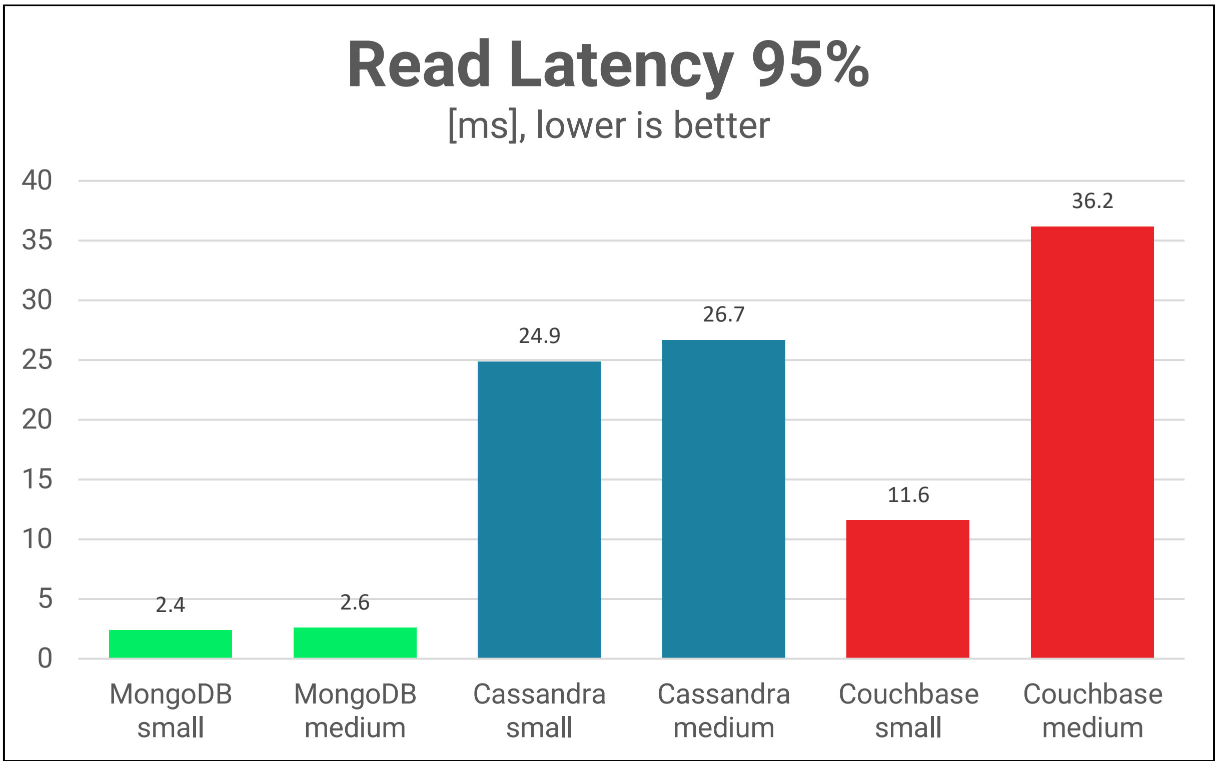
- ➔ Database throughput differences exist for relational databases, too.
- ➔ Far less transactions can be handled for this more complex workload, compared to the one in the previous chapter.
- ➔ Workload complexity and characteristics have impact on the throughput capabilities of database technologies.

1.3. Latency – The Real-Time Experience

While throughput is very important to handle all operations, many applications also need a real-time experience with low latency. As applications often consist of several layers, the database latency is only one part of the total latency. Database technologies usually show different latencies for specific database operations, like writing, reading, updating....

Let's have a look at the NoSQL scenario from chapter 1.3 but focus on read latency.

Target Technology	NoSQL databases on Cloud infrastructure
Databases	a) MongoDB CE v5 b) Apache Cassandra v4 c) Couchbase Server CE v7
Infrastructure	AWS EC2
Scaling	a) small: m5.large (2 vCPUs, 8 GB RAM, single-node) b) medium: m5.xlarge with doubled workload threads
Workloads	YCSB: 50% read, 50% write, simple operations (no joins, no aggregations, simple search)



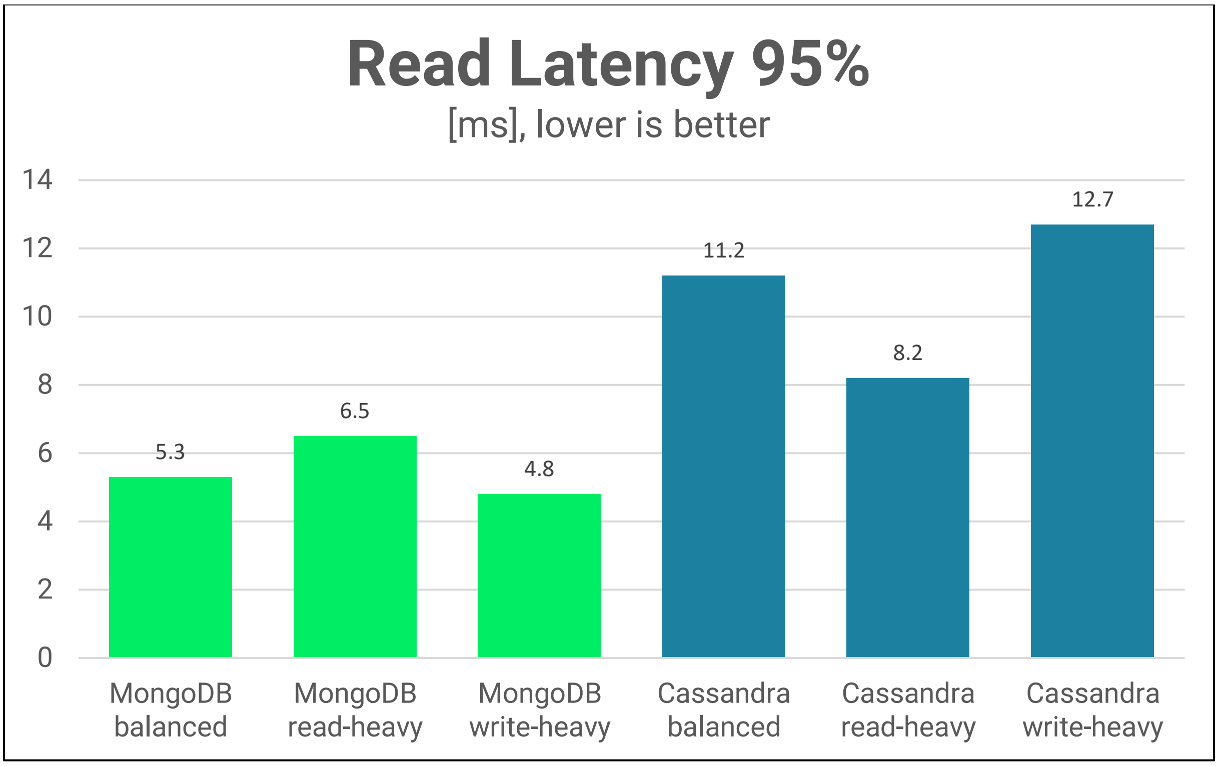
- ➔ The read latency ranking is different compared to the throughput, remember lower latency is better.
- ➔ For growing application workload, latency should stay on same level, but not grow.
- ➔ Read latency 95% means, that 95% of all operations complete faster than this number.

1.4. Database Strengths – MongoDB vs Cassandra

Already in the previous chapters one can see that databases have specific strengths and that the workload has an impact on the performance.

Let's visualize these strengths with three different workload types for read latency.

Target Technology	NoSQL databases on Cloud infrastructure
Databases	a) MongoDB CE v5 b) Apache Cassandra v4
Infrastructure	AWS EC2
Resources	3-node cluster, i3.xlarge (4 vCPUs, 30.6 GB RAM), NVME
Workloads	a) YCSB balanced: 50% read, 50% write, simple operations b) YCSB read-heavy: 80% read, 20% write, simple ops. c) YCSB write-heavy: 20% read, 80% write, simple ops.

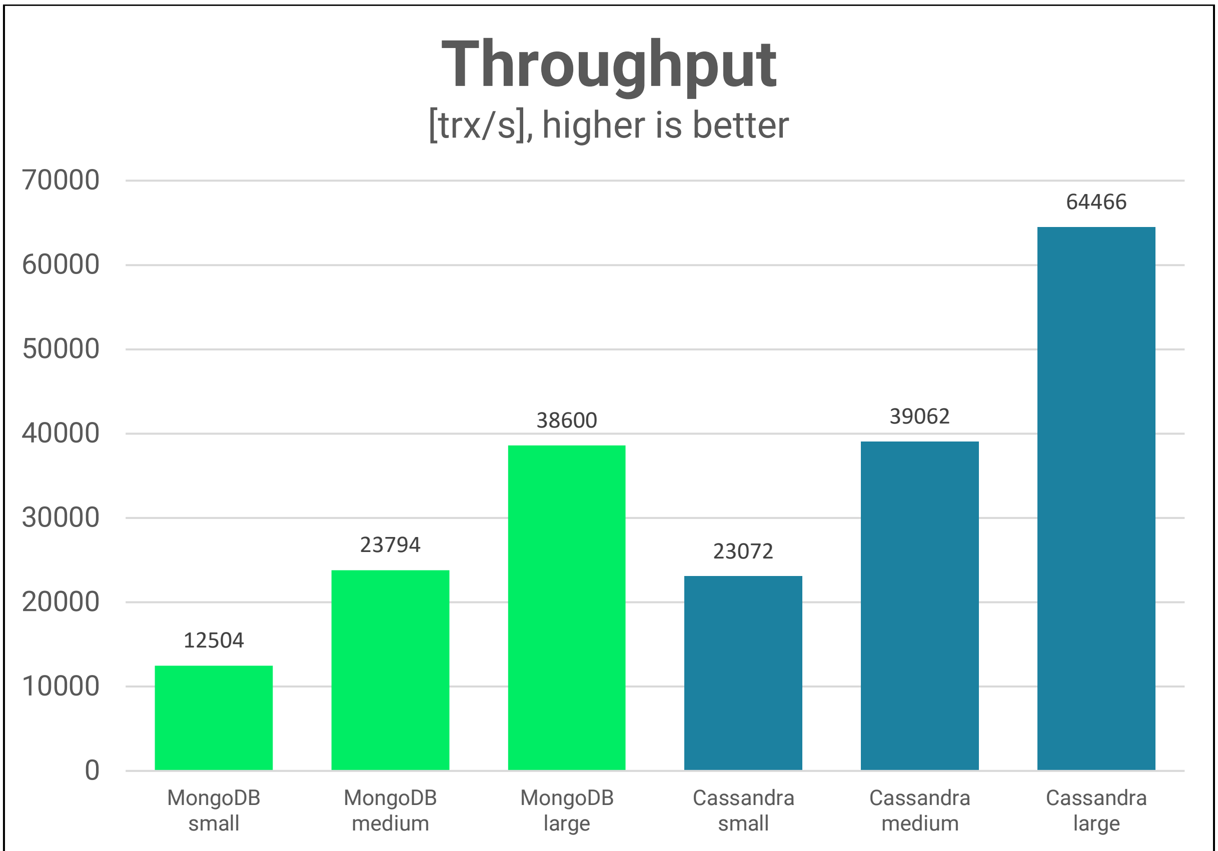


- ➔ The distribution of the database operations has an impact on the latency for read operations.
- ➔ MongoDB shows lowest read latency for write-heavy, Cassandra for read-heavy workloads.
- ➔ Measurements tells you something about the strength of databases.

1.5. When scaling matters – MongoDB vs Cassandra

The performance of present workload is one thing, but what will it be in the future? The ability to scale, as linear as possible, is important for growing applications. A simplified scalability model based on compute power assumes that the scalability factor is reflected by the increased compute capacity from the small to large cluster size, i.e. the theoretical throughout scaling factor is 400% from small to large.

Target Technology	NoSQL databases on Cloud infrastructure
Databases	a) MongoDB CE v5 b) Apache Cassandra v4
Infrastructure	AWS EC2
Resources	a) small: 3-node cluster, i3.xlarge, NVMe storage b) medium: doubled resources/instances c) large: quadrupled resources/instances
Workloads	YCSB balanced: 50% read, 50% write, simple operations (small: 100, medium: 200, large: 400 threads)

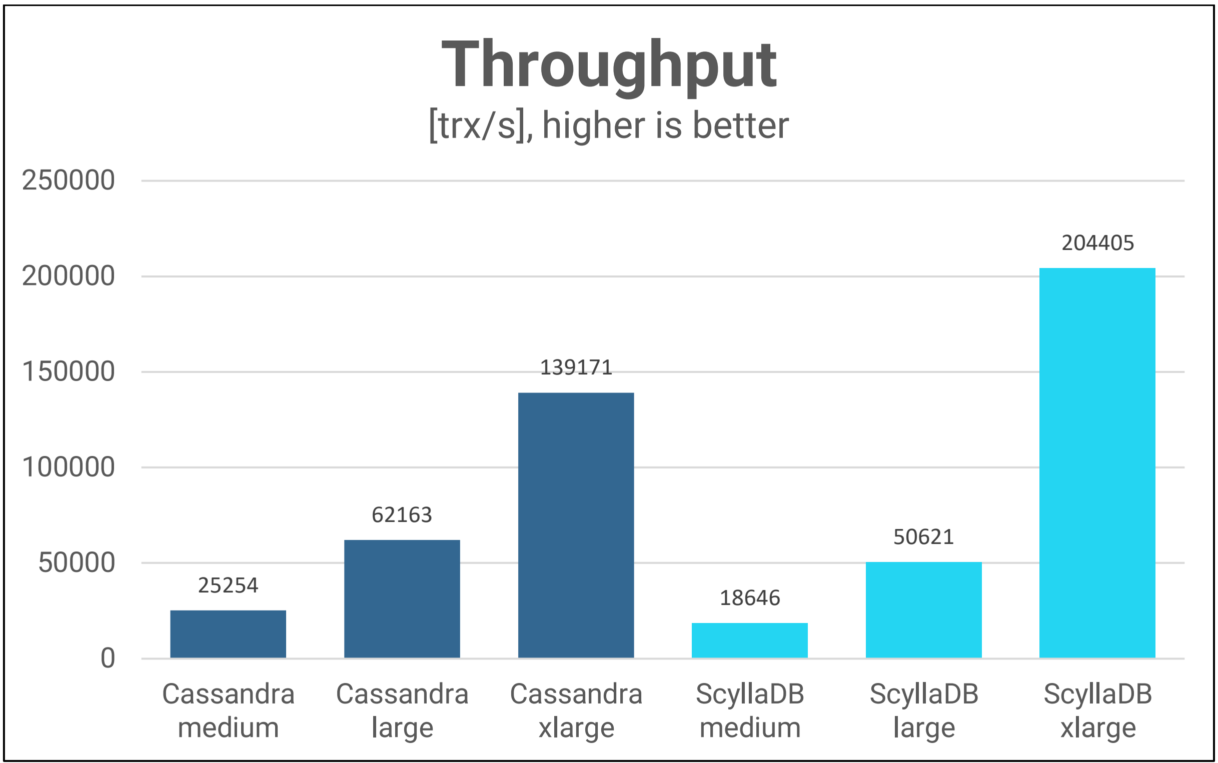


- ➔ While resources doubled and quadrupled, MongoDB gained 90% and 200% more performance.
- ➔ While resources doubled and quadrupled, Cassandra gained 69% and 180% more performance.
- ➔ Scaling is very database specific, but seldom linear, even if marketing claims it.

1.6. Looking Beyond the Mainstream – A Positive Example

While many architects only know the most popular databases, far over 300 database technologies exist. One of them is ScyllaDB, a relatively new NoSQL database that adopts many concepts of Apache Cassandra and enhances them with its close-to-the metal design. It is built specifically for applications that require high throughput and predictable low latency.

Target Technology	NoSQL databases on Cloud infrastructure
Databases	a) Apache Cassandra v4 b) ScyllaDB v4.5
Infrastructure	AWS EC2
Resources	a) medium: 3-node cluster, m5.xlarge b) large: 3-node cluster, m5.2xlarge c) xlarge: 9-node cluster, m5.2xlarge
Workloads	YCSB balanced: 50% read, 50% write, simple operations (medium: 100, large: 200, xlarge: 600 threads)

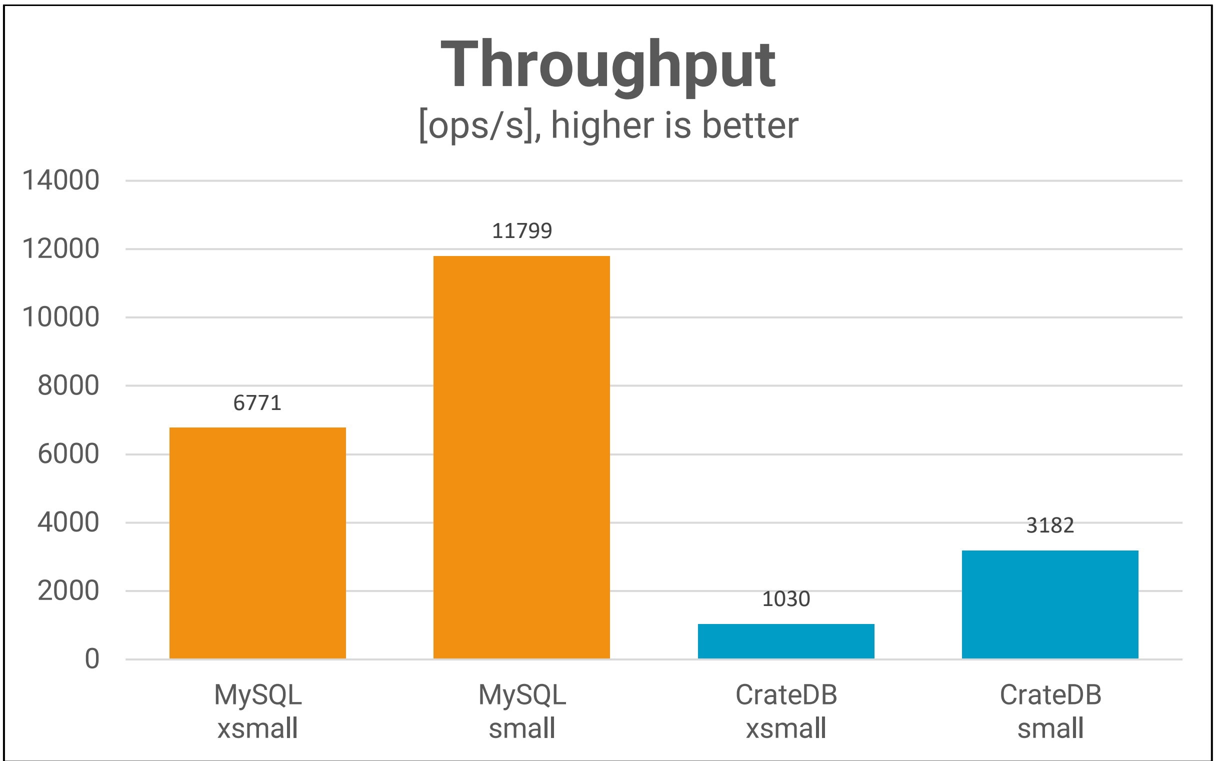


- ➔ Cassandra has slight performance advantages for medium and large scaling sizes.
- ➔ ScyllaDB shows its performance strengths for intense and large-scale workloads.
- ➔ There are some hidden champions out there, they just need to be found.

1.7. Looking Beyond the Mainstream – A Less Positive Example

Among the more than 300 databases, there are of course numerous products that are not yet 100% technically mature and whose performance leaves much to be desired in comparison to established database solutions.

Target Technology	NoSQL databases on Cloud infrastructure
Databases	a) MySQL v8 b) CrateDB v4.7
Infrastructure	AWS EC2
Resources	a) xsmall: single-node, m5.large b) small: single-node, m5.xlarge
Workloads	YCSB: balanced 50% read, 50% write, simple operations



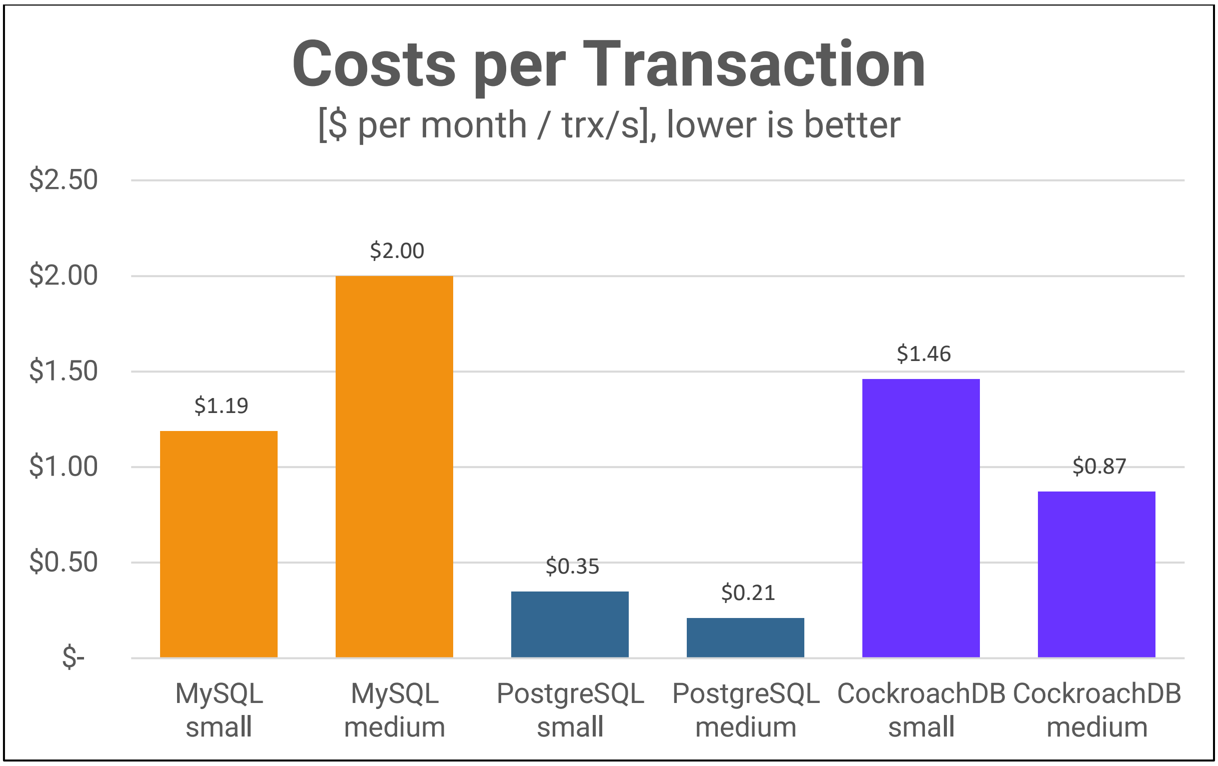
- ➔ MySQL outperforms CrateDB 6x (xsmall) and 3,7x (small) for a simple CRUD workload.
- ➔ CrateDB is a time-series database that should handle simple CRUD workload fast. But it seems that this workload is not a good match. Better results for CrateDB have been published here using an analytical workload: https://benchmark.clickhouse.com/Some technologies need more maturity, before they can have industrial relevance. Or just are not a good fit for certain workloads.

1.8. Costs? Who is interested in Performance/Costs?

While techies are looking for performance and scalability, you should never forget the cost perspective, especially if you want to impress your manager.

Costs per transaction is one of the most important key metrics for businesses.

Target Technology	Relational databases on Cloud infrastructure
Databases	a) PostgreSQL v13 b) MySQL v8 c) Cockroach v21
Infrastructure	AWS EC2
Scaling	a) small: m5.large (2 vCPUs, 8 GB RAM, single-node) b) medium: resources x2, workload x2
Workloads	Sysbench 1.0: OLTP Mix with non-simple operations, grouped to transactions, no batch processing



- ➔ The costs for a transaction vary from database to database and in the scaling size. Optimizing the costs per transactions leads to an efficient and successful data infrastructure.
- ➔ For a fair cost calculation, it is important to consider all relevant costs, not only the resource costs (as we have done it here 😉)
- ➔ Costs are not an important KPI, but Performance per Cost is a real one!

2. The Purpose of Databases – Handling Your Workload

While the first chapter gave us some first insight about some databases and shows the enormous potential of such performance measurements, it also gave us a hint about one further benchmarking fact: workload matters.

Every application is unique and has an individual workload. Of course, you can categorize most applications like ERP, eCommerce shop, IoT application or AI algorithm, but still it is unique regarding

- the amount of data,
- the data set size,
- the number of (parallel) users,
- the request,
- the distribution of the requests regarding database operations
- the distribution of the requests regarding the specific data sets
- … and many more

And yes, each of these workload specifics has an impact on the performance, the best database solution for the application and the best database tuning.

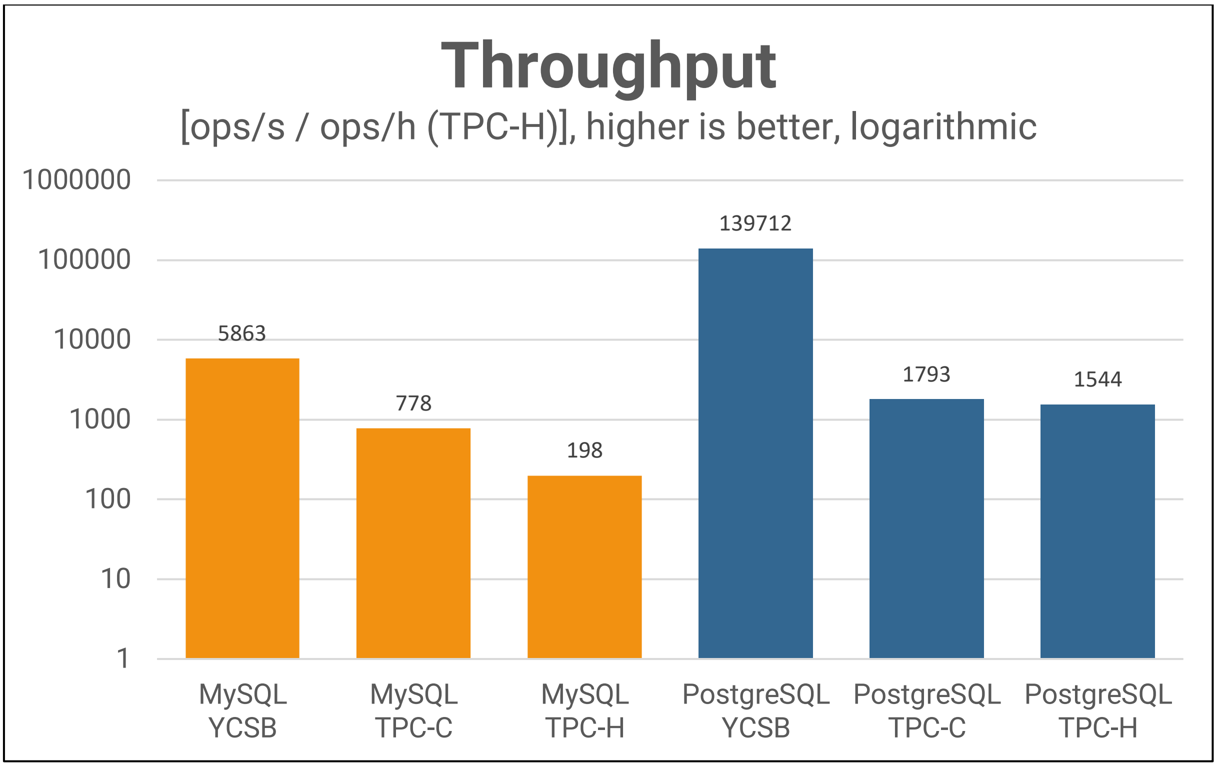
In Chapter 2, we provide you with performance data regarding workload variations – form traditional relational databases to modern database technologies.

Let’s dive in!

2.1. Handling Different Workload Types

MySQL and PostgreSQL are among the most popular databases, used for nearly every workload, from eCommerce to analytics. The following diagram shows why performance depends so much on the workload.

Target Technology	Relational databases on Cloud infrastructure
Databases	a) MySQL v8 b) PostgreSQL v13
Infrastructure	AWS EC2
Resources	16 vCPUs, 128GB RAM, NVMe storage
Workloads	a) YCSB: load phase, bulk inserts b) TPC-C: transactional eCommerce, semi-complex queries c) TPC-H: analytical workload, complex queries

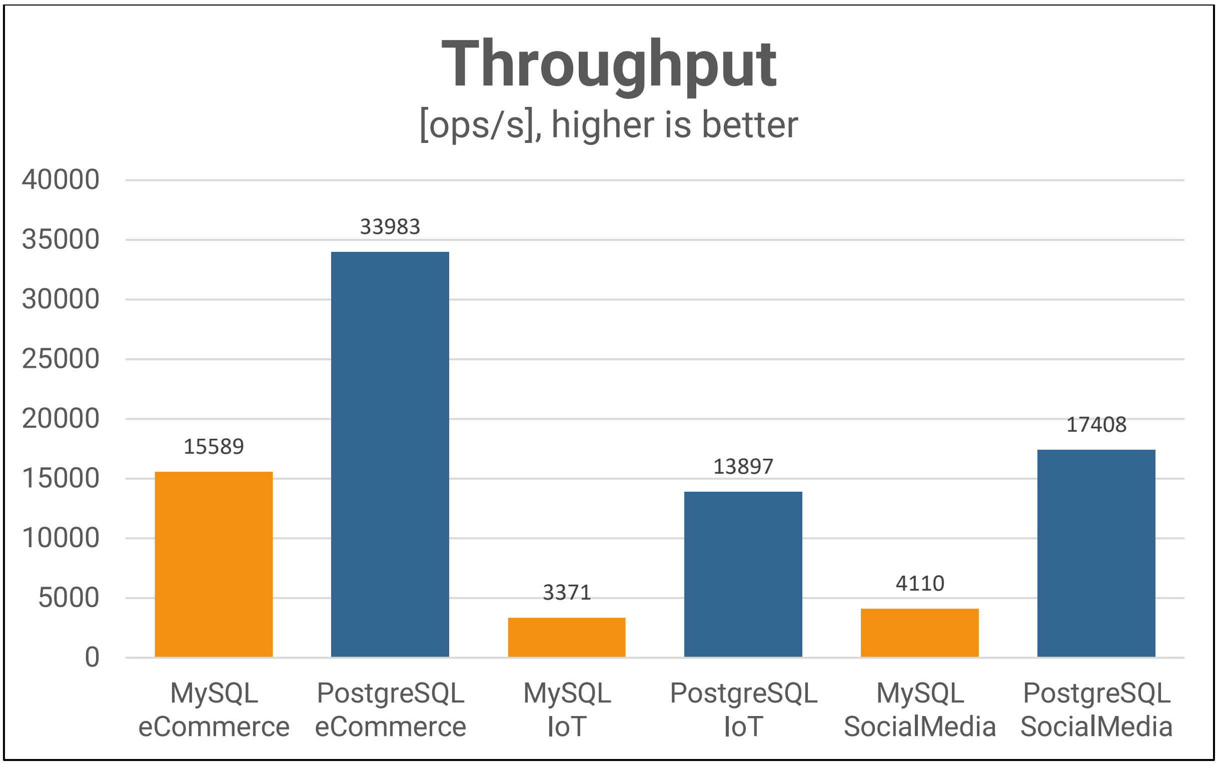


- ➔ Different workloads and complexity show different throughput results for same resources and settings. A logarithmic scale is necessary to present the results
- ➔ Complex workloads like analytics are usually not measured by ops/s, but per operation per hour due to long-lasting queries.
- ➔ Workload understanding and re-modelling is key for useful performance measurements and optimizing the database layer.

2.2. Handling Different Workload Variation - Relational

Not always are the workloads that different as in the last example, but also simple CRUD workloads with a shift in the distribution of database operations can lead to different performance results and showing the strengths and weaknesses of databases.

Target Technology	Relational DBMS on Cloud infrastructure
Databases	a) PostgreSQL v12 b) MySQL v8
Infrastructure	AWS EC2
Resources	c6i.2xlarge: 8 vCPUs, 16GB RAM, single node
Workloads	a) YCSB eCommerce: 90% read, 10% insert operations b) YCSB IoT: 80% insert, 20% read operations c) YCSB SocialMedia: 50% insert, 50% read operations

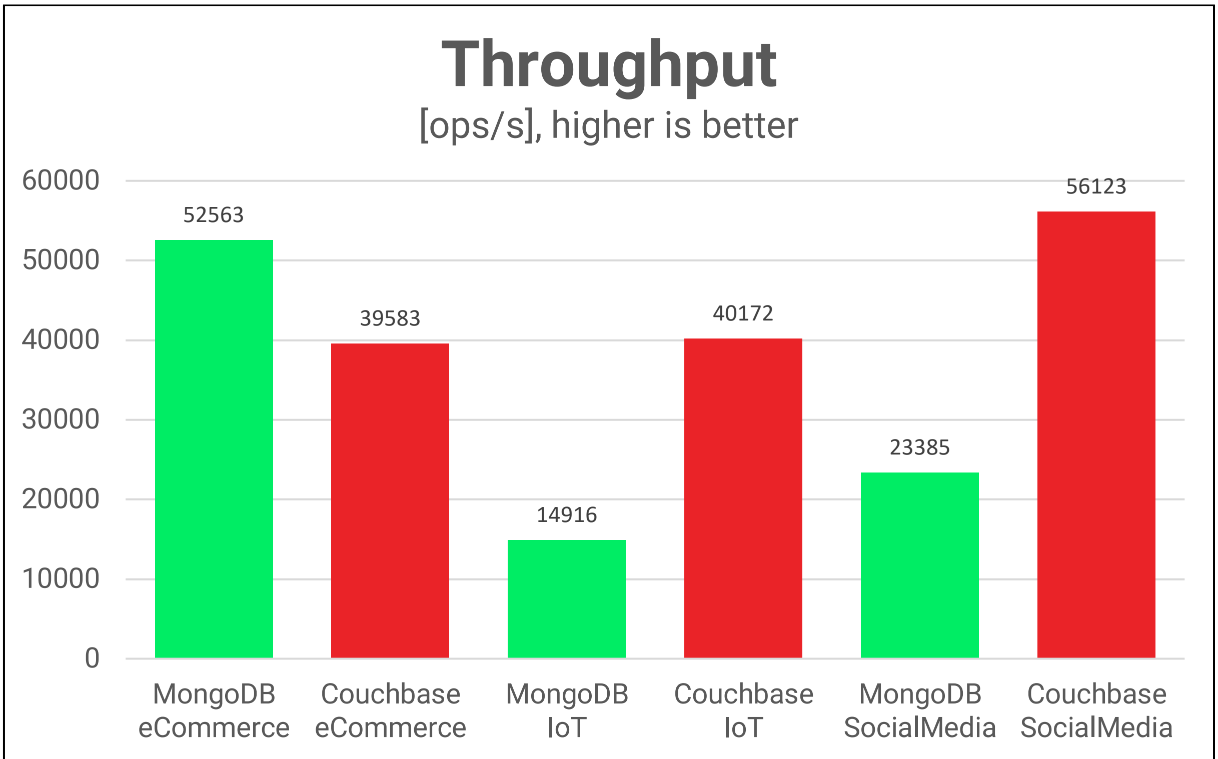


- ➔ While MySQL is good at handling a balanced non-complex eCommerce workload, the performance for write-heavy IoT workload and the social media collapsed.
- ➔ The same pattern can also be detected for PostgreSQL, but not with the same extreme decline. Still, it looks that these workloads are not the sweet-spot of these databases.
- ➔ Even \`general purpose\` databases show strength and weaknesses for specific workloads. Sometimes dedicated database solutions are a more efficient fit.

2.3. Handling Different Workload Variation - NoSQL

Not always are the workloads that different than in the last example, but also simple CRUD workloads can lead to different performance results and showing the strengths of databases.

Target Technology	NoSQL on Cloud infrastructure
Databases	a) MongoDB v4.4 b) Couchbase v7
Infrastructure	AWS EC2
Resources	c6i.2xlarge: 8 vCPUs, 16GB RAM, 3-node-cluster
Workloads	a) YCSB eCommerce: 90% read, 10% insert operations b) YCSB IoT: 80% insert, 20% read operations c) YCSB SocialMedia: 50% insert, 50% read operations



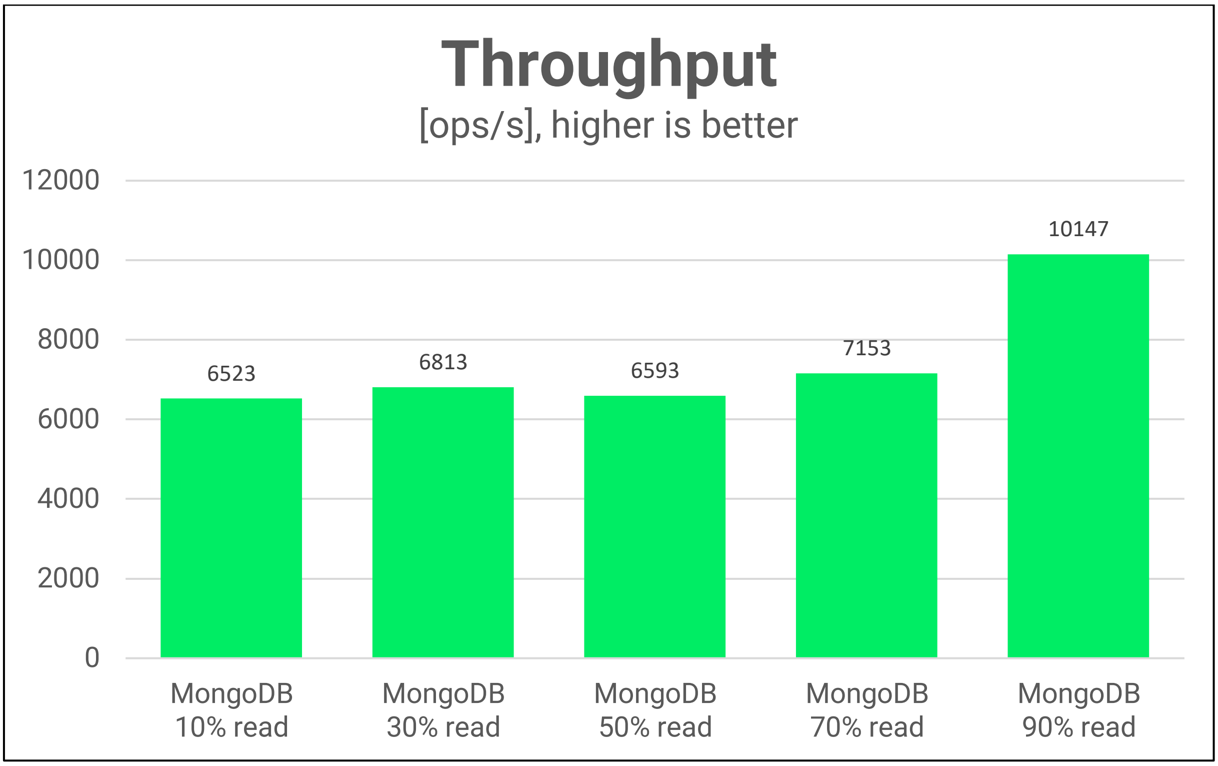
- ➔ MongoDB shows highly different throughput results regarding the workload specifications. While the write intense workload is not handled efficiently it looks more promising for read-intense workloads.
- ➔ Couchbase's performance results are not that dependable on the workload, but even here you can see an increase of 40% throughput for the social media workload.
- ➔ Even slight changes in workloads can have a significant impact on the resulting performance of a database. And the strengths of databases are varying a lot.

2.4. A Shift in the Read-Write Ratio for MongoDB

In the last example, the performance sensitivity of MongoDB due to variation in the workload regarding the read-write-ratio was already visible, here comes an in-depth analysis of this phenomenon.

It is important to understand, that shifts of the workload due to new features can have an overall impact on the database performance.

Target Technology	NoSQL on Cloud infrastructure
Databases	MongoDB v4.4
Infrastructure	AWS EC2
Resources	m5.large: 2 vCPUs, 8GB RAM, 3-node setup
Workloads	YCSB simple CRUD: read-write distribution variable



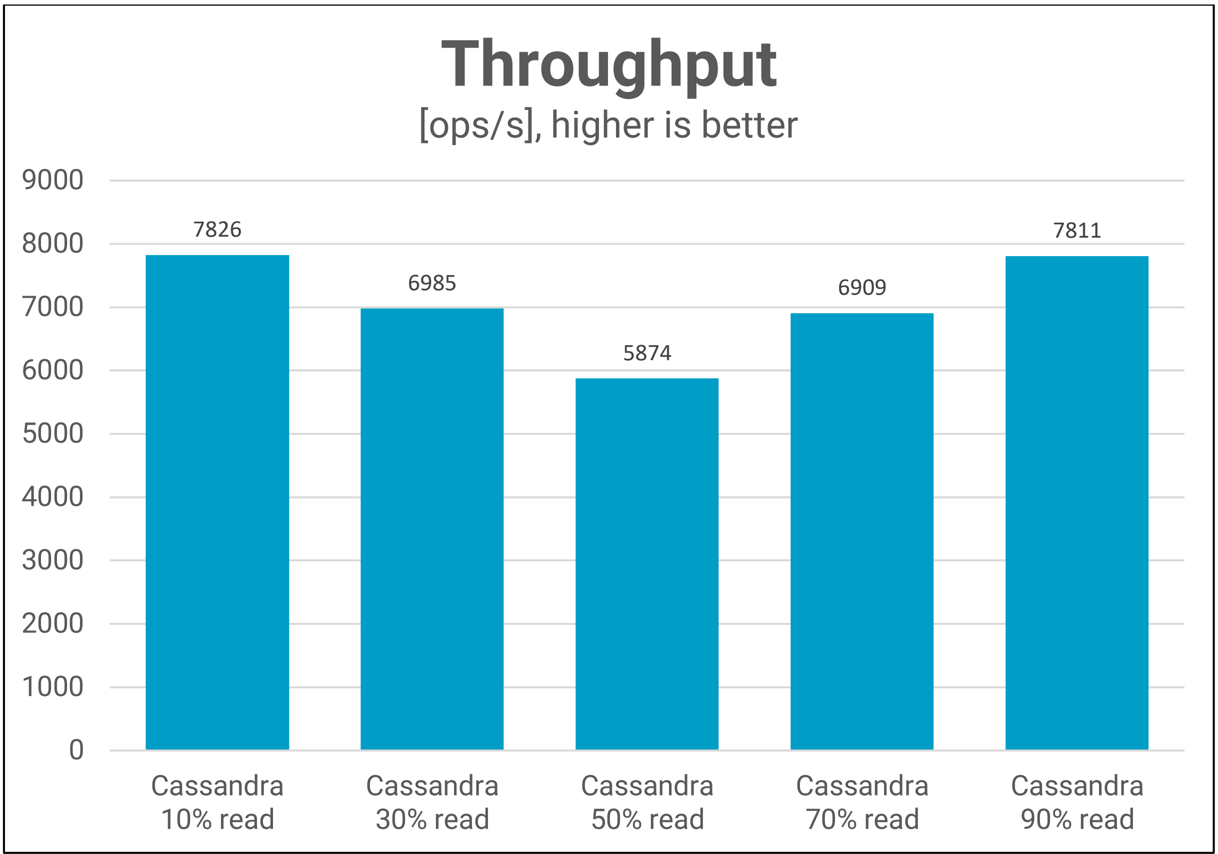
- ➔ While the read percentage is below 70% the throughput results are nearly on the same level. Yet, with an increasing ratio of read operations and lower write operations the possible throughput of MongoDB increases significantly.
- ➔ Write operations look more performance intensive at MongoDB. This is normal for nearly any database, but the impact at MongoDB is significantly high with over 50%.
- ➔ But not only the throughput, also the latency for read and write operations can (and does) change with workload shifts like these.

2.5. A Shift in the Read-Write Ratio for Cassandra

The same scenarios as above, we also did for Apache Cassandra to find out if more write operations always deliver lower throughput, due to more cost intense internal operation, or if this is also database specific.

The results were more than surprising.

Target Technology	NoSQL on Cloud infrastructure
Databases	Cassandra v4.0
Infrastructure	AWS EC2
Resources	m5.large: 2 vCPUs, 8GB RAM, 3-node setup
Workloads	YCSB: simple CRUD, read-write distribution variable



- ➔ Cassandra shows a totally different behavior on the variation of the read-write distribution compared to MongoDB.
- ➔ While the balanced 50r/50w workload shows the worst performance, the performance increases for read intense but also for write intense workloads by nearly 30%
- ➔ It is nearly impossible to predict the specific behavior, but performance measurements can help to identify this easily.

3. The Infrastructure Impact

In the last two chapters, we were digging into database performance and the impact on the workload on performance, but we never questioned the underlying resources. Which impact do have

- Different cloud provider
- different VM types
- different storage types
- IOPS
- or self-hosted infrastructure

Cloud does not equal Cloud

If you are assuming that Cloud resources from AWS or Azure, or european cloud providers like IONOS cloud and Open Telekom Cloud are delivering identical performance for comparable virtual machines, we will prove you wrong.

While pricing is nearly equal for many cloud providers for comparable resources, the performance of the technical cloud solution, regarding hardware, software virtualization or provisioning rules, is extremely divers.

Finding the right resources for your application can increase the performance and especially the performance per costs significantly.

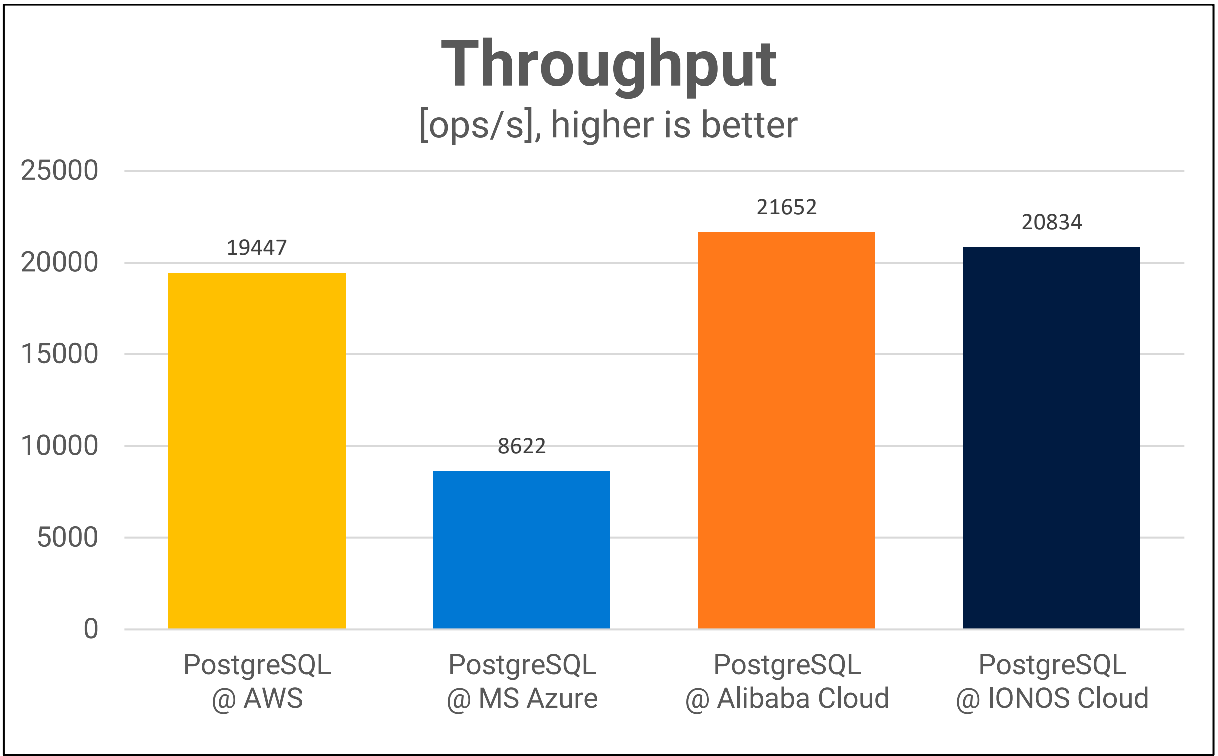
Let’s dive into some measurements on infrastructure layer.

3.1. Cloud Providers Impacting PostgreSQL Performance

A first good example, how the underlying cloud infrastructure has an impact on the outcoming performance of the database running on the infrastructure.

Measuring this performance differences is not only important for performance, but even more for performance per cost comparisons before choosing cloud resources.

Target Technology	PostgreSQL on Cloud infrastructure
Databases	PostgreSQL
Infrastructure	a) AWS EC2 b) MS Azure c) Alibaba Cloud d) IONOS Cloud
Resources	Single-node, comparable general-purpose VMs with 4 vCPUs and 16 GB RAM, standard SSD storage
Workloads	YCSB: 50% read, 50% write, simple operations

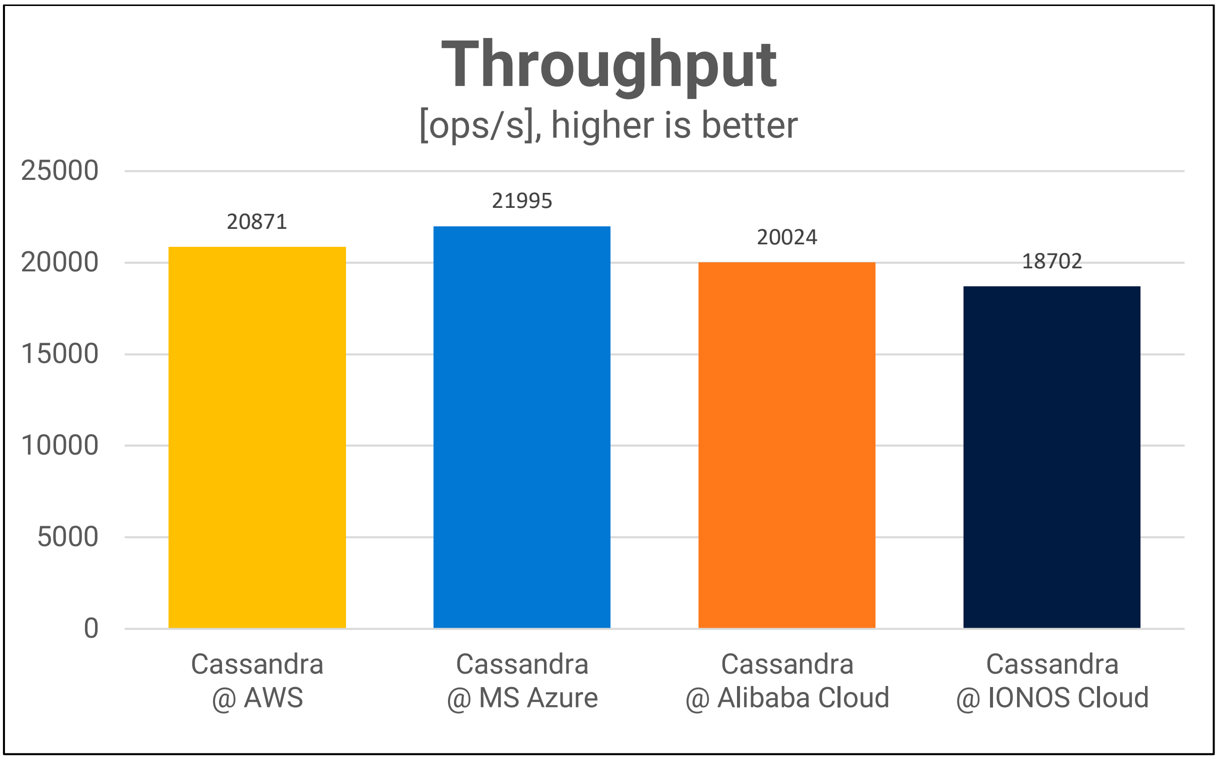


- ➔ The performance of PostgreSQL on Alibaba Cloud is more than 10% better, and on IONOS Cloud more than 5%, better than on AWS EC2.
- ➔ The performance on the Microsoft Azure infrastructure is way lower in this example. You would need significantly higher and more expensive VMs for similar performance as in the other cases, which is not efficient and unsatisfying.

3.2. Cloud Providers Not Impacting Cassandra

The above example shows an extreme performance impact on PostgreSQL, but this is not necessarily the same for other databases. This example shows the performance impact for the same workload and infrastructure resources for Apache Cassandra.

Target Technology	Apache Cassandra on Cloud infrastructure
Databases	PostgreSQL
Infrastructure	a) AWS EC2 b) MS Azure c) Alibaba Cloud d) IONOS Cloud
Resources	Single-node, comparable general-purpose VMs with 4 vCPUs and 16 GB RAM, standards SSD
Workloads	YCSB: 50% read, 50% write, simple operations



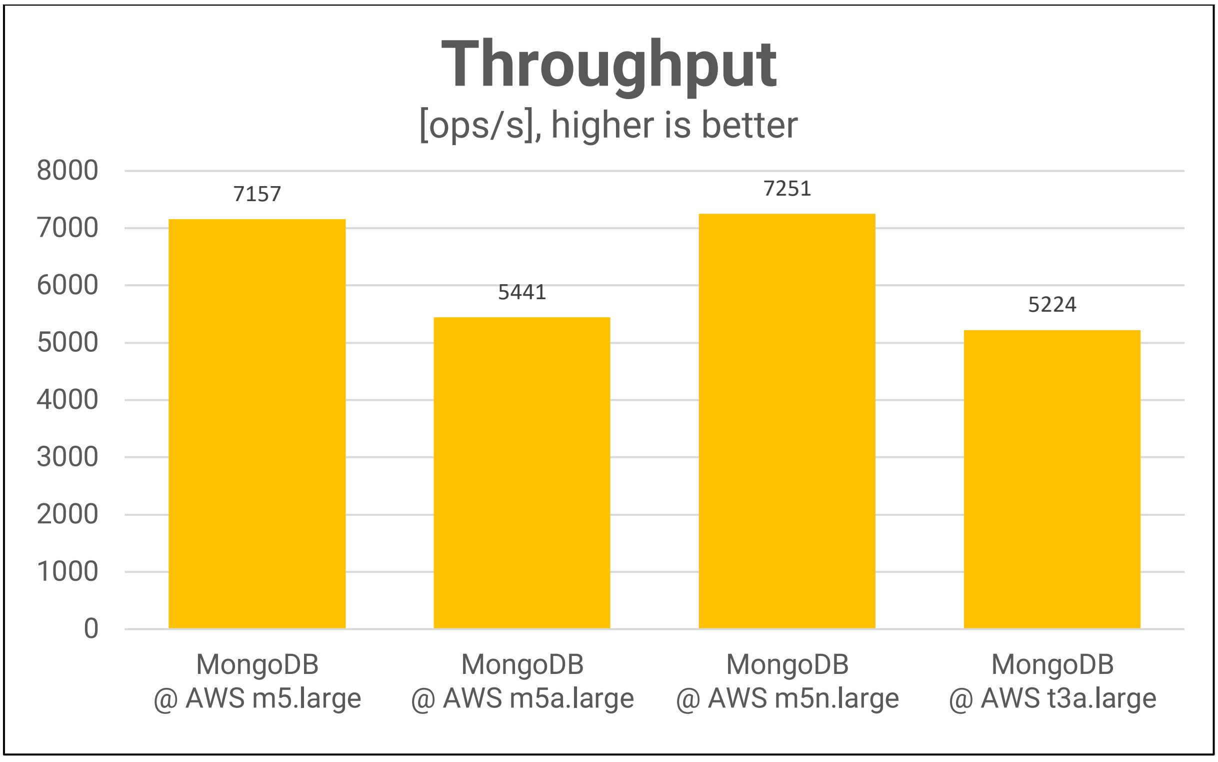
- ➔ The performance impact of the cloud resources on Cassandra are below 10%.
- ➔ Only IONOS cloud shows some dropping performance numbers. But on the other hand, Microsoft Azure shows strongest performance for this scenario.
- ➔ Some databases are more dependent on the underlying resources and some cloud resources are not working efficient with some database technologies.

3.3. Performance Differences of VM Types

Beside the differences of Cloud providers, their hardware and virtualization, there are also big differences in the available Virtual Machine types at one provider.

Is this relevant for the database performance? Sure!

Target Technology	MongoDB on Cloud infrastructure
Databases	MongoDB v4.4
Infrastructure	AWS EC2
Resources	a) m5.large: 2 vCPUs, 6 GB RAM b) m5a.large: 2 vCPUs, 6 GB RAM, AMD c) m5n.large: 2 vCPUs, 6 GB RAM, network-optimized d) t3a.large: 2 vCPUs, 6 GB RAM, AMD, burst
Workloads	YCSB: simple CRUD, 80% read / 20% writes



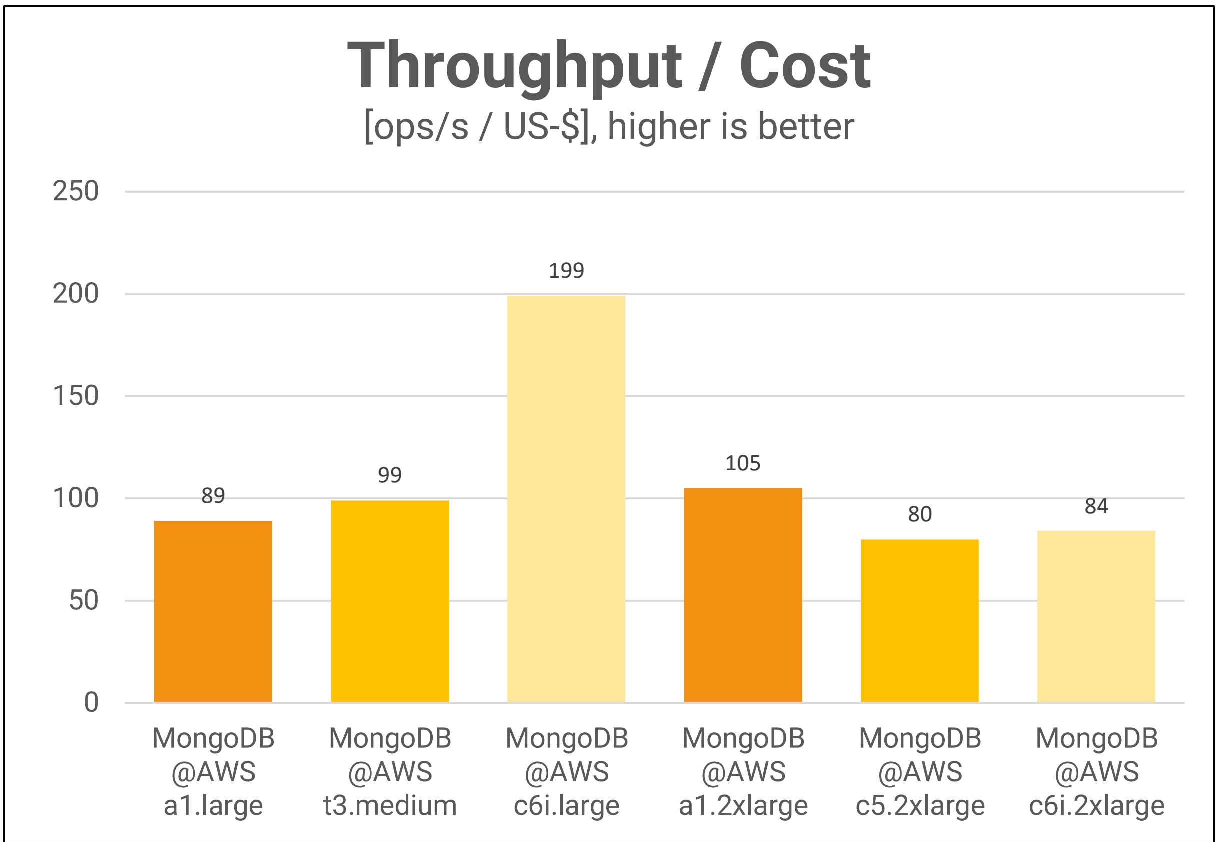
- ➔ The performance differences of similar VM types of AWS EC2 vary up to 30% for the MongoDB throughput.
- ➔ Every database or workload can have different requirements to CPU, RAM or even network bandwidth, storage or IOPS.
- ➔ Throughput and costs should always be considered together when selecting the right cloud resources to find an efficient solution- from technical and business perspective.

3.4. Price/Performance of ARM Resources for Databases

The number of VM types at public cloud providers are complemented with ARM Graviton processors since 2021. ARM VMs are usually lower priced due to lower hardware costs.

Yet, do these resources work properly with databases? What's their price/performance?

Target Technology	PostgreSQL on Cloud infrastructure
Databases	MongoDB v4.4
Infrastructure	AWS EC2
Resources	Single-node, ARM Graviton vs. different Intel VMs at two scaling sizes: 2 vCPUs, 4 GB RAM and 8 vCPUs, 16 GB RAM
Workloads	YCSB: eCommerce, simple CRUD, 90% read, 10% write, latest



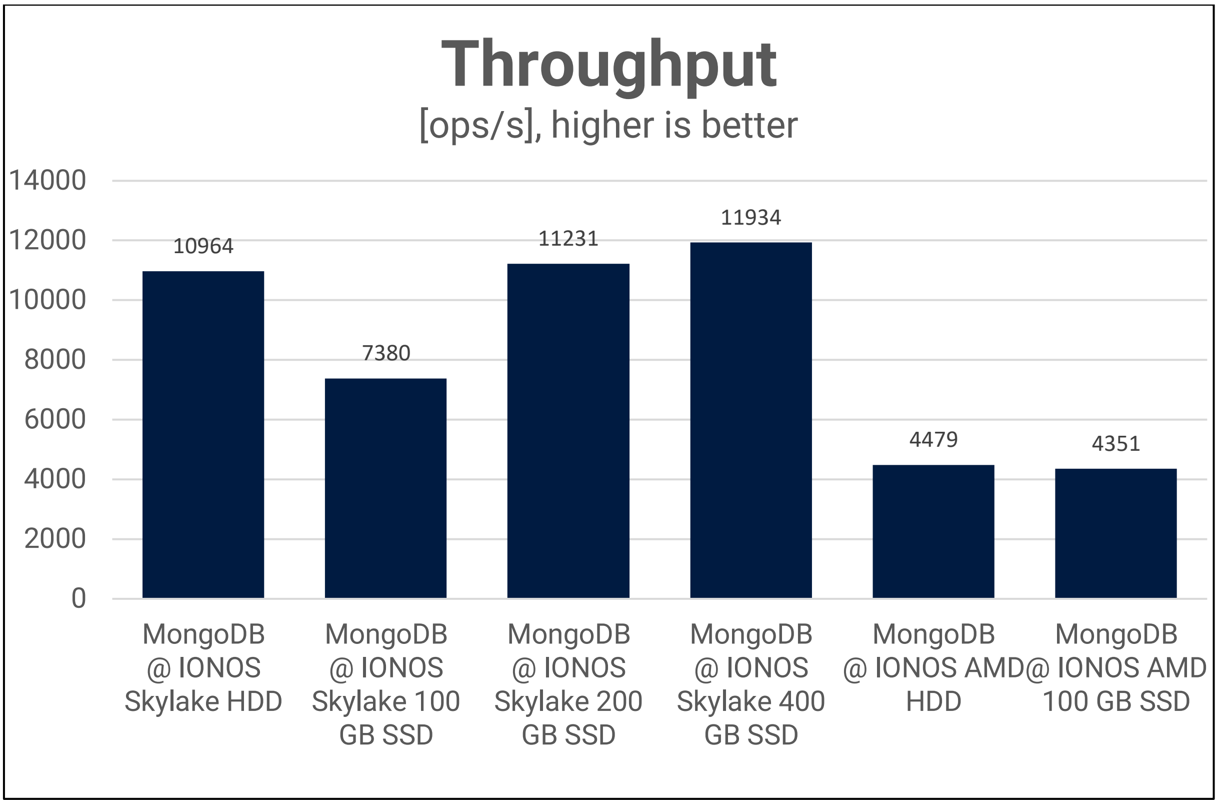
- ➔ While the absolute performance (not visible) of the ARM VMs (dark orange) is lower than the other AWS VMs, their performance per cost ratio is fine, especially for larger instances. The small c6i instance delivers an incredible price/performance ratio.
- ➔ The differences between the small and the larger ARM graviton instances is significantly.
- ➔ Note: Not every database has a driver for ARM instances, yet.

3.5. Storage Types Performance Impact

Besides VM types, it is also important to have a suitable storage for the database technology and workload requirements. The performance increase of better, but more expensive storage solutions, can't be calculated but easily measured.

Sometimes the results are very surprising due to internal hardware limitations, like in this example of IONOS cloud, a smaller European Cloud provider.

Target Technology	MongoDB on Cloud infrastructure
Databases	MongoDB v4.4
Infrastructure	IONOS Cloud
Resources	Cross-product: HDD and SSD storage with different VM types (2vCPUs, 8 GB RAM)
Workloads	YCSB: simple CRUD, 80% read / 20% writes



➔ The HDD delivers better performance at lower price compared to the small SSDs.

➔ This surprising result is due to unlimited IOPS for the cheap HDD compared to limits for smaller SSDs at IONOS Cloud.

➔ The performance of the larger SSDs is slightly better due to higher IOPS.

4. Why Benchmarking Database-as-a-Service?

In the last three years Database-as-a-Service (DBaaS) have become the future of modern database management. DBaaS provide a fully managed solution for deployment, operations and support of various database technologies. DBaaS are offered by database providers, cloud providers and specialized DBaaS companies.

In one of our market studies we identified 25 DBaaS solutions for MySQL only and over 30 DBaaS solutions for PostgreSQL. In total, we found more than 175 commercially available DBaaS products, numbers still growing.

DBaaS Performance Does Not Only Depend on the DBMS Technology

In the last few chapters, we saw that the performance of a database installation depends on the dedicated DBMS technology, the application workload, and the underlying infrastructure resources.

This means, that Database-as-a-Service products with identical DBMs technology, the same workload hosted on the same resources show identical performance behavior. Right?

Nah, this is not the case!

The way a DBaaS technology is implemented and orchestrated does further influence the DBaaS performance.

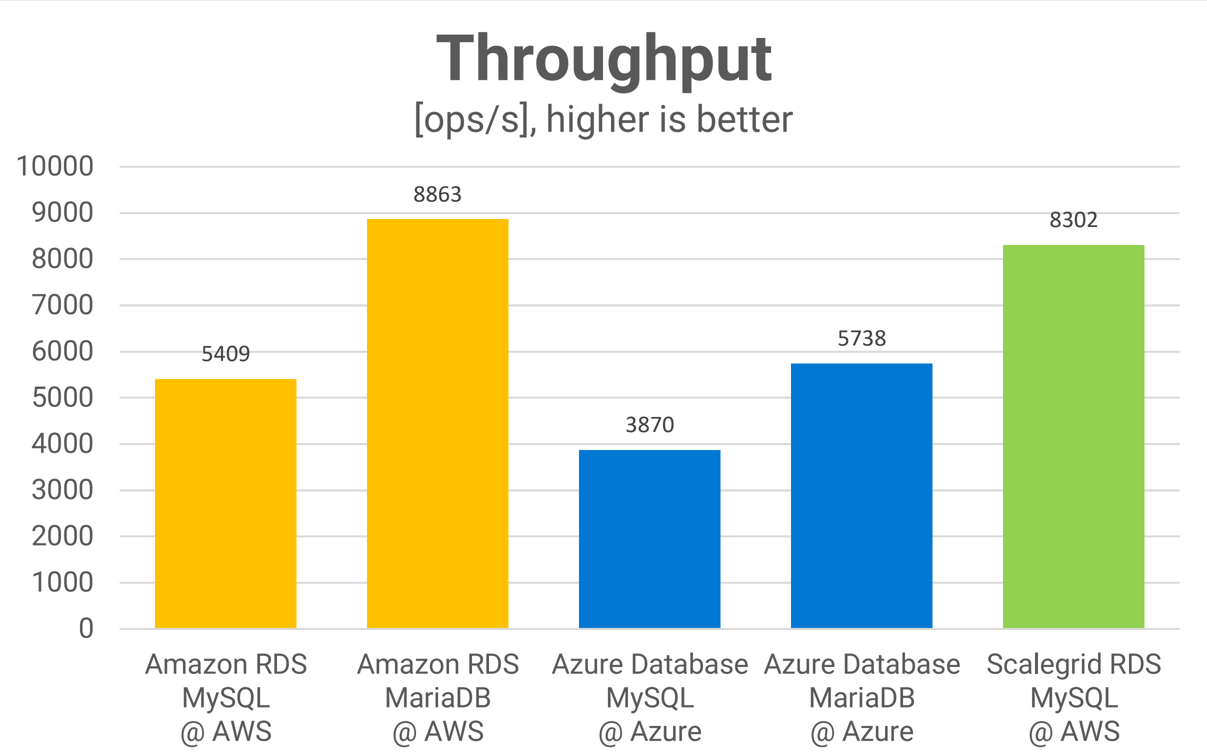
Let's compare some performance results.

4.1. MySQL and MariaDB DBaaS: Market Overview

MariaDB is a DBMs technology, based on an early MySQL fork. Both are open-source databases and many DBaaS products are implemented for these two technologies.

Yet, the performance and of course the performance per costs vary immensely.

Target Technology	Relational DBaaS
Databases	a) MySQL DBaaS b) MariaDB DBaaS
Infrastructure	a) AWS RDS b) MS Azure Databases
Resources	8 vCPUs, 64GB RAM, 2-nodes high availability (exception: MS Azure MariaDB only single-node available)
Workloads	YCSB: simple CRUD, 50% read – 50% write operations

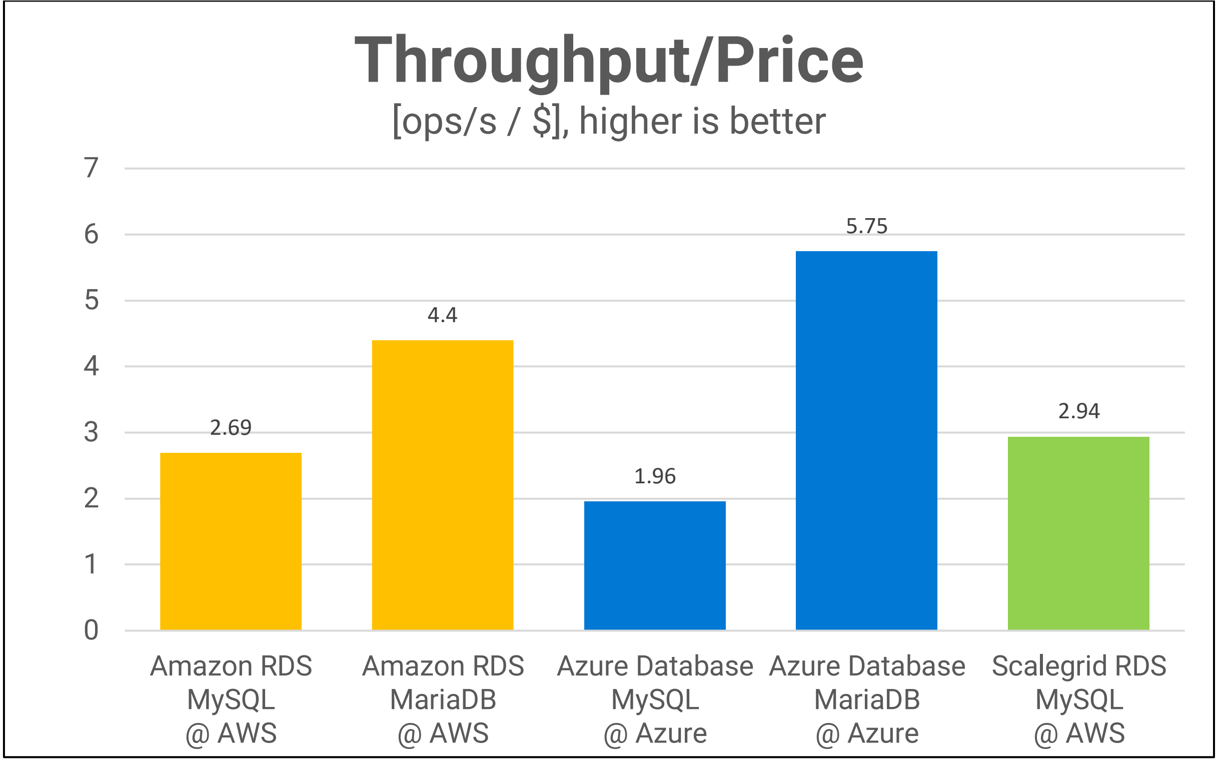


- ➔ The throughput of MySQL DBaaS highly differ, even if it is the same database technology. The differences are higher than 40% for similar DBaaS solutions.
- ➔ The same applies for MariaDB, where AWS RDS outperforms MS Azure's DBaaS solution by more than 50% for nearly identical pricing.
- ➔ Scalegrid's MySQL DBaaS delivers first class performance and can outperform the MySQL DBaaS solution of the two hyperscalers.

4.2. MySQL and MariaDB: DBaaS Performance/Cost Comparison

For the example above, a price/performance comparison is more than interesting. The different cost structures of DBaaS solutions are more than complex and divers. Besides compute costs, storage, network, backup, and support costs need to be considered for a fair comparison.

Target Technology	Relational DBaaS
Databases	a) MySQL DBaaS b) MariaDB DBaaS
Infrastructure	a) AWS b) MS Azure
Resources	8 vCPUs, 64GB RAM, 2-nodes high availability (exception: Azure MariaDB only single-node available)
Workloads	YCSB: simple CRUD, 50% read – 50% write operations

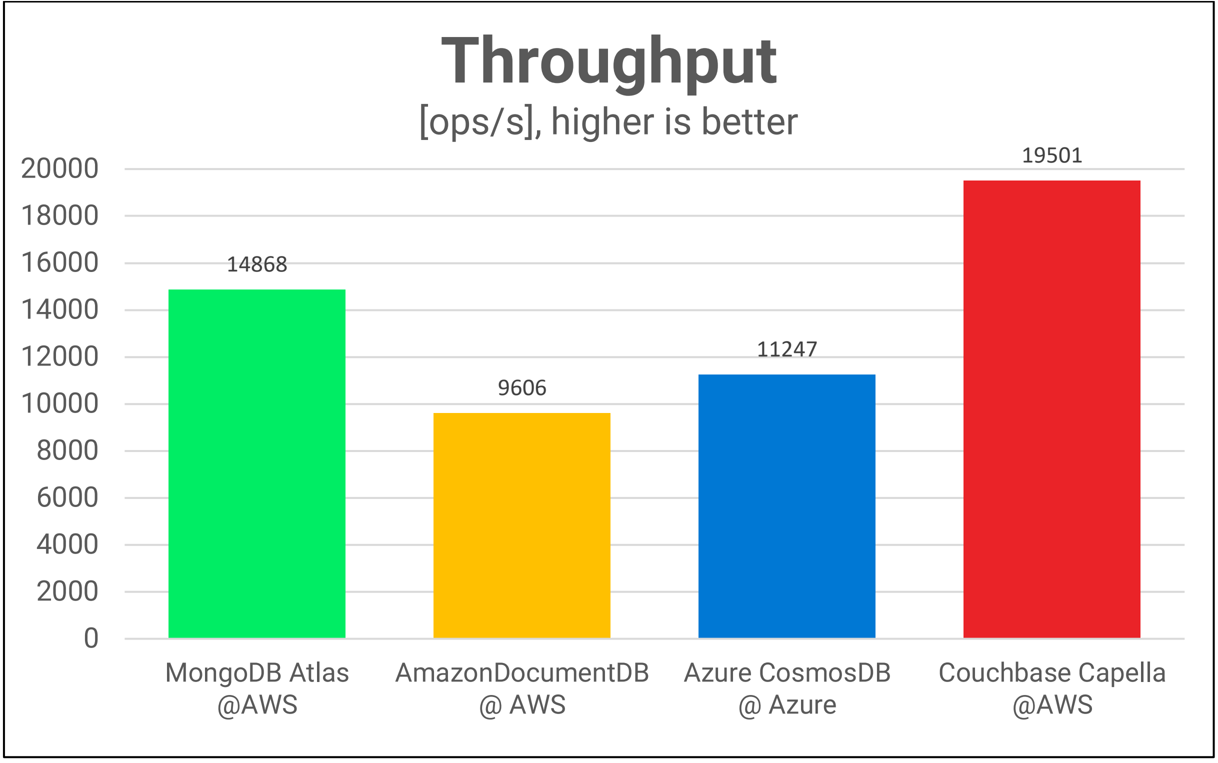


- ➔ The non-HA-setup of MS Azure Database for MariaDB provides best price/performance, but is not technical comparable without a reliability node.
- ➔ Scalegrid's MySQL DBaaS costs more than MySQL DBaaS from AWS RDS and MS Azure, but due its great performance it has still a better price/performance ratio.
- ➔ AWS RDS MariaDB provides more than 60% better price/performance than MySQL.

4.3. A Document DBaaS Market Comparison

For document-oriented DBaaS products many DBaaS products similar to MongoDB Atlas, the official DBaaS service of MongoDB, can be found on the market.

Target Technology	Document-oriented DBaaS
Databases	a) MongoDB Atlas b) AWS DocumentDB c) Azure CosmosDB d) Couchbase Capella
Infrastructure	a) AWS b) MS Azure
Resources	8 vCPUs, 32GB RAM, 3-node cluster
Workloads	YCSB: simple CRUD, 50% read – 50% write operations

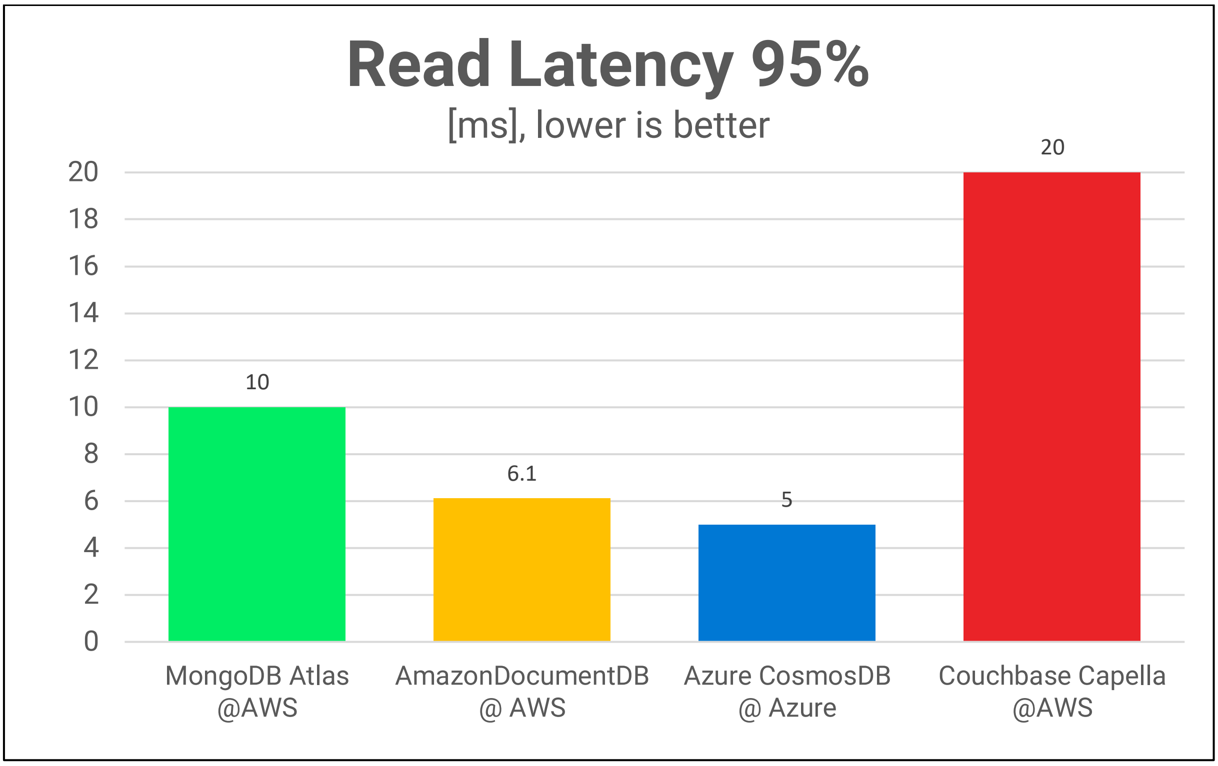


- ➔ The DBaaS products based on the long-existing MongoDB and Couchbase database implementation outperforms the cloud-native solutions of AWS and Azure.
- ➔ Couchbase Capella is 30% faster than MongoDB Atlas for this workload and resource size.
- ➔ Due to the high DBaaS costs, performance measurements for DBaaS have a high business value for selecting the right DBaaS and the right resource size.

4.4. Document DBaaS – Also Big Differences for Latency

Not only Throughput and throughput per costs are important KPIs for DBaaS, also the latencies can vary a lot, depending on the software and hardware implementation.

Target Technology	Document-oriented DBaaS
Databases	a) MongoDB Atlas b) AWS DocumentDB c) Azure CosmosDB d) Couchbase Capella
Infrastructure	a) AWS b) MS Azure
Resources	8 vCPUs, 32GB RAM, 3-node cluster
Workloads	YCSB: simple CRUD, 50% read – 50% write operations

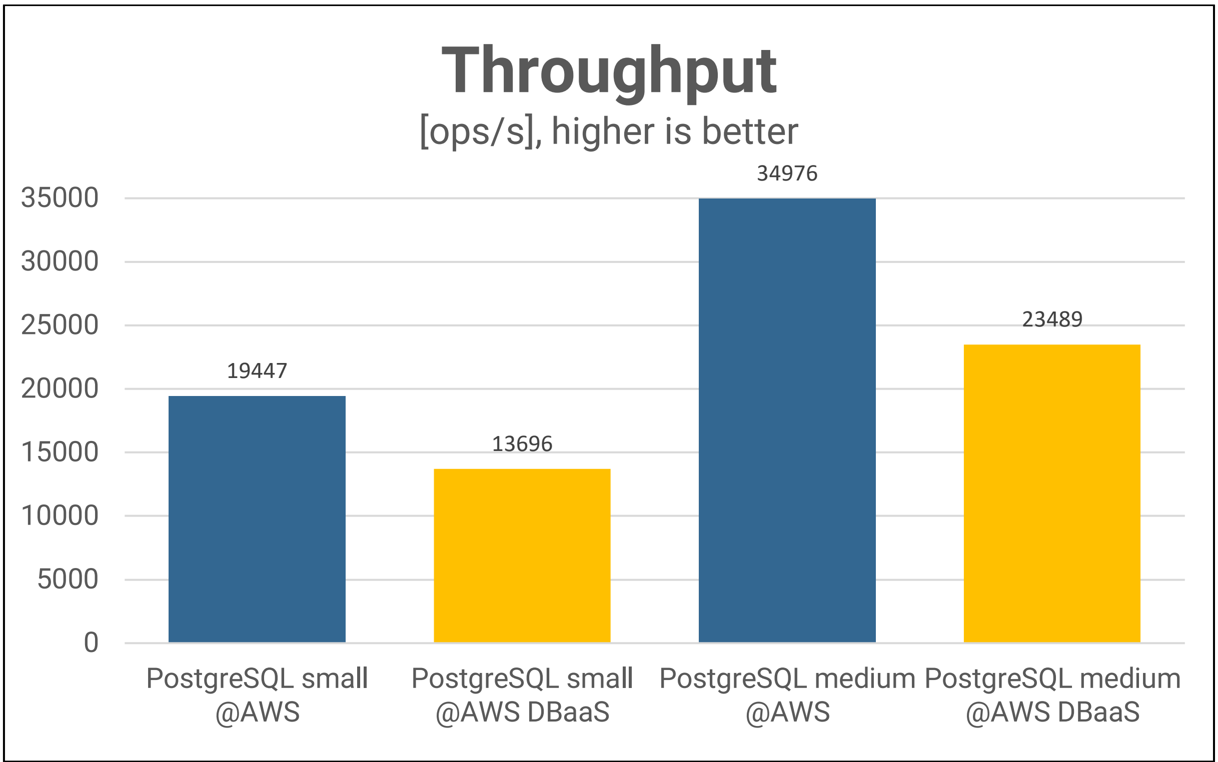


- ➔ The cloud-native DBaaS solutions of AWS and Azure show the lowest latency in this scenario.
- ➔ Couchbase Capella has over 4x higher latency than CosmosDB and yields unsatisfying results.
- ➔ Besides costs and throughput, latency can also be important for applications using a DBaaS database solution.

4.5. DBaaS vs self-managed DBMS

DBaaS products are a high-tech implementation for databases, but their performance compared to self-managed databases on similar resources is not identical. When migrating to a DBaaS product often the resource sizing needs to be adapted.

Target Technology	PostgreSQL self-managed vs DBaaS
Databases	a) PostgreSQL v13 b) AWS RDS PostgreSQL v13
Infrastructure	AWS EC2
Resources	a) Small: 2 vCPUs, 8 GB RAM b) Medium: 4 vCPUs, 16 GB RAM
Workloads	YCSB: 50% read, 50% write, simple operations



- ➔ The self-managed databases show 40% higher throughput than the DBaaS product on similar AWS resources in this scenario.
- ➔ One reason could be, that the DBaaS product has production features like backup and security, which are influencing the performance.
- ➔ Note: When migrating to DBaaS it is uncertain which resource size is necessary.

Conclusion

Wow, that are a lot of data points you fought your way through.

R – E – S – P – C –T – ! – ! – !

I hope you found some interesting and relevant metrics for your daily work. At least, you saw a lot of differences and surprising results. In many use cases performance measurements can help you to make a good data-driven decision.

If you are interested in doing measurements on your own, or struggling with some questions, or have concerns regarding the results., feel free to reach out to me.

Best

Daniel & The benchANT Team

daniel.seybold@benchant.com

About the Authors – The benchANT Team

This document is based on the work of the benchANT team. All insights come from the daily work in performance testing for clients and research. benchANT is specialized on database and infrastructure benchmarking to deliver data-driven technology recommendations and optimization suggestions for existing IT applications.

Dr. Jörg Domaschka

Jörg has a doctorate in computer science and has spent more than 15 years researching cloud computing and DevOps. His penchant for avoiding repetitive tasks drives him to develop modern, automated solutions for the daily problems in IT departments.

Dr. Daniel Seybold

Dr. Daniel Seybold gained extensive experience with distributed systems and databases during his doctoral studies. The research results lay the foundation for benchANT's goal of providing companies with an objective performance assessment of cloud and database services.

Jan Ocker

After studying physics, Jan gained experience in IT project management and data analytics before dedicating himself to (cloud) cost calculations and the Database-as-a-service market at benchANT. Jan attaches great importance to efficiency - both in IT applications and in the daily workflow.

Glossary

CRUD: CRUD are the simple database operations Create-Read-Update-Delete. In our context it describes a database workload type, which only consists of these simple database operations without joins, transactions or aggregation.

DBaaS: Database-as-a-Service (DBaaS) are fully-managed database management systems with deployment features, management tooling and support. They are offered by database and cloud providers to simplify database administration.

OLAP: On-Line-Analytical-Processing (OLAP) describes a workload type, which consists of complex database queries, usually used in analytics and business intelligence.

OLTP: On-Line-Transaction-Processing (OLAP) describes a workload type, which consists of relational and transactional database queries - similar to classical ERP and eCommerce applications.

Sysbench: Sysbench is a widely used open-source benchmarking suite, not only for databases, but also for CPU and memory testing. For database performance measurements, the Sysbench can create a transactional workload. More information can be found here: https://github.com/akopytov/sysbench

TPC-C: The TPC-C is a standardized benchmarking from the TPC council for online transaction processing (OLTP). More information can be found here: https://www.tpc.org/tpcc/

TPC-h: The TPC-h is a standardized benchmarking from the TPC council for online analytics processing (OLAP). More information can be found here: https://www.tpc.org/tpch/

YCSB: The Yahoo! Cloud Serving Benchmark (YCSB) is an open-source benchmarking suite. It is widely used to create synthetic CRUD workloads for performance measurements. More information can be found here: https://github.com/brianfrankcooper/YCSB and here https://benchant.com/de/blog/ycsb.

Disclaimer

All performance measurements were done with the automated benchmarking framework of benchANT in an automated, reliable and producible way in the last quarters.

All meta and raw data can be found on GitHub: https://github.com/benchANT

Please notify us, if you can prove some wrong measurements, we will update this document properly.

Version history

- V1 - 2023-08: release first version