White Paper

AEROSPIKE

# Minimising $CO_2$ Production from IT Systems

## $CO_2$ Emission Efficiency as a measurable Non-Functional requirement

# Contents

# The Case for Monitoring $CO_2$ Emissions of Software Platforms

Concern over mankind's impact on the environment and the unchecked effects of climate change is the dominant global issue of our time. The burning of fossil fuels is the principal contributor to the creation of planet-heating greenhouse gases[1]. Any industry reliant on electrical power will therefore potentially contribute to global warming. This includes the IT sector - the impact its consumption has is substantial and growing.

The impact is greater than you might think. Harvard researchers expect that by 2030 information and computing technology will account for as much as 20% of global energy demand[2]. Environmental costs are not limited to the running costs - equipment manufacturing is significant in its own right.

IT infrastructure manufacturers already have a larger carbon footprint than more obviously polluting industries. For example, a recent Bloomberg article noted that "Intel's factories used more than three times as much water as Ford Motor Co.'s plants and created more than twice as much hazardous waste."[3]

The concern is registered at all levels of society, including the boardroom. Two-thirds of FTSE100 companies have voluntarily committed to net-zero targets, meaning that $CO_2$ Emission reduction is front and centre for executives around the world. Given the significance of the emissions produced by IT infrastructure, decision-makers need to consider "$CO_2$ Emission Efficiency" as a Non-Functional Requirement for IT systems.

Until recently, emissions of running software platforms were largely unquantified and mostly neglected in Carbon Accounting. This is changing however and perhaps the best evidence of this is that the three major cloud technology providers are now reporting the $CO_2$ emissions produced by the consumption of their resources by individual customers (AWS, Azure, and GCP). There are, therefore, no technical reasons for not accounting for these emissions in consumer $CO_2$ balance sheets and likely there will be pressure to do so. Bare metal emissions will likely not be far behind.

To help IT decision-makers reduce the environmental impact of their software platforms, this article introduces a model for estimating and comparing the efficiency of similar technologies. In the end, the reader can expect to have answers to the following two questions:

1. What is the impact of the $CO_2$ emission savings that a switch between two nominally similar technologies can cause? (Hint: remarkably significant)

---

[1] https://www.ipcc.ch/site/assets/uploads/2018/02/ipcc_wg3_ar5_chapter5.pdf page5, 78%

[2] https://arxiv.org/abs/2011.02839

[3] https://www.bloomberg.com/news/articles/2021-04-08/the-chip-industry-has-a-problem-with-its-giant-carbon-footprint

2. What is the cost of a $CO_2$ Emission Efficient technology compared to its rivals? (Hint: significantly cheaper)

## Introduction

Before the advent of the Internet and the rise in availability of distributed, scalable computing resources, efficient use of computing resources was mandatory. When faced with the scale of the Internet, efficiency concerns gave way to scalability - first and foremost, the user base had to be provided for.

Scalable resources allow us to address performance-related non-functional requirements such as latency, throughput, capacity, and growth effectively, but not efficiently. As the access to virtually unlimited resources has never been easier, scaling has become the solution to almost every performance-related requirement, with efficiency often being neglected. Times are changing however.

- Enterprises are committing to net-zero emission targets which forces them to reduce resource and energy consumption.
- Business owners are increasingly alert to the often substantial cost of ownership of software platforms.
- Platform owners are concerned with the operational complexity of managing sometimes vast infrastructure estates.
- Limits on growth such as network and computing speed mean developers must once again address efficiency when designing and writing software.

Scaling resources without considering efficiency is ultimately not sustainable. This opens the door to revisiting efficiency. This time around, we are talking about the efficiency of scalable applications.

## Efficiency

In computer science, efficiency is generally evaluated in terms of CPU usage. An algorithm is more efficient than another if it uses fewer CPU cycles to solve a problem. Reducing CPU time is often the main focus of software developers. Efficiency and speed have therefore become synonymous in the computer industry.

But efficiency is not only about speed. More efficient software also requires less hardware, uses less energy, and has a lower TCO. To illustrate the point, let's consider a scenario where we need to perform a task 1000 times per second. Suppose there are two algorithms for performing that task:

- Algorithm A, which requires 10 CPU cycles.
- Algorithm B, which requires 100 CPU cycles.

Assuming that a CPU core can execute 1000 cycles per second, algorithm A requires 10 cores to run 1000 times per second, and algorithm B needs 100 cores to fulfil the same requirement. So if A is used:

- Each individual task would complete 10x faster.
- 10x fewer resources would be required.
- The cost of the infrastructure would be 10x lower.
- Emissions produced by the underlying infrastructure would be 10x lower.

## Selecting Two Technologies to Compare

My area of expertise is databases. Therefore, I decided to compare two technologies I am most knowledgeable about: Apache Cassandra and Aerospike. This article estimates and compares the $CO_2$ emission differences of these nominally similar technologies.

I expect that the same model I introduce in this article could be used for comparing other software platforms.

## Comparison Methodology

To compare these databases we need to specify fixed workloads and then compare the resulting emissions. The workload choices are:

1. Handling a specific level of throughput.
2. Managing a specific volume of data.

A throughput-oriented test is problematic as it in turn necessitates a large number of choices - respectively read/write balance, record size, data model, overall throughput and testing client type and number. It can be seen that any one choice may be seen to be favouring one platform over the other. For that reason, the comparison chosen is volume-based, with the volume being 1 PB. This is a reasonable order of magnitude choice as:

- A large organisation often has hundreds of databases. The data size in each database will range from a few hundred gigabytes to a few hundred terabytes.
- In addition to production environments, organisations will also make use of non-production environments such as Test, Staging, UAT, and Pre-Production.

Most enterprises will therefore have petabytes, if not tens or even hundreds of petabytes of data in their databases. Additionally, the databases subject to this study are linearly scalable, therefore the emissions for a 1PB use case can be readily converted to those arising from larger or smaller data volumes.

For each of the database technologies under investigation, I discuss the following for a 1 PB use case:

1. **Physical Resource Estimate**: Minimum recommended hardware if using AWS.
2. **Emissions Estimate:** This includes emissions produced during the hardware manufacturing process and emissions generated by operational use. (Scope 1, 2, and 3)
3. **Operational Cost**: The monetary cost of deploying versus the AWS platform[4].

I chose the AWS platform as I made use of benchmark data already sourced using the platform. Also, the $CO_2$ emissions of AWS EC2 instances were available. Finally, this allows the reported results to be reproduced and tested.

# Physical Resource Estimate

In this section, I determine the required AWS hardware to store 1 PB of raw data when using the vendors' best practice guides.

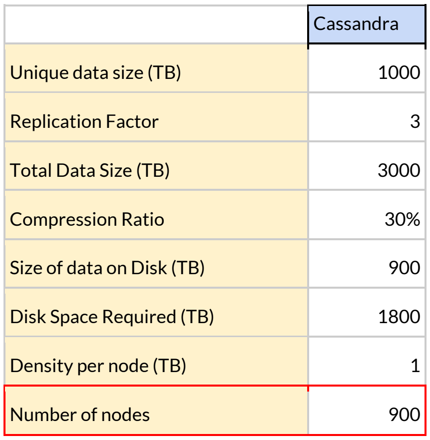
## Apache Cassandra

### Assumptions

- **Data Density**: The main contributor to the Apache Cassandra project (DataStax) recommends storing no more than 1 terabyte of data per node of Apache Cassandra[5].
- **Replication Factor**: Cassandra requires 3 copies of the data to remain consistent and available in case of a node failure.
- **Compression Ratio**: Assume the data can be compressed to 30% of its original size.
- **Operational Requirements**: Cassandra requires 50% of the disk to be empty.

### Calculations

Based on the above, for a 1PB use case



| | Cassandra |
|---|---|
| Unique data size (TB) | 1000 |
| Replication Factor | 3 |
| Total Data Size (TB) | 3000 |
| Compression Ratio | 30% |
| Size of data on Disk (TB) | 900 |
| Disk Space Required (TB) | 1800 |
| Density per node (TB) | 1 |
| Number of nodes | 900 |

---

[4] As the world's most popular cloud computing platform, AWS is a common logical choice when making relative cost comparisons.

[5] https://docs.datastax.com/en/cassandra-oss/planning/planning/ossCapacityPlanning.html

### Disk choices

Apache Cassandra can use two types of AWS resources to store data.

- Local NVMe drives (a.k.a. ephemeral storage)
- Elastic Block Storage (EBS) - network-attached virtual drives

Both options are considered in this analysis as there is a meaningful difference between the results of these two approaches.

The Apache Cassandra website recommends c4.4xlarge instances on EC2 with EBS storage[6]. As the density is 1 TB and Cassandra needs 50% empty disk space, the size of the EBS volume must be at least 2 TB. Also recommended is an additional .5 TB capacity for snapshots[7], commitlog[8], hinted handoffs[9], and other Cassandra overheads.

When storing the data on ephemeral storage, i3 instances with attached NVMes are recommended[10]. Storing 1TB of data in Cassandra requires a minimum of 16 virtual cores. i3.4xlarge is the smallest instance type in this series, offering 16vCPU or above.

The resulting instance type choices for 1TB of data are summarised below.

| **Instance** | Cores | RAM (GB) | Number of disks | Disk Size (TB) |
|---|---|---|---|---|
| c4.4xlarge on EBS | 16 | 30 | EBS | 2500 |
| i3.4xlarge on NVMe | 16 | 122 | 2 | 1900 |

**Note**: In my opinion, Apache Cassandra and DataStax websites' recommendations regarding the instance types are not the best options. Using more modern equivalent instance types like c6g and i3en, the cost and emissions could be reduced up to 20% without affecting the performance. To avoid subjectivity however, as I wanted to have referenceable decisions, I  use standard recommendations. Although an additional 20% saving is significant, it does not change the conclusions of this article.

## Aerospike

### Assumptions

- **Data Density**: The theoretical per node limit for disks when using Aerospike is 256 TB. In practice,  the limit is the disk capacity that can be attached to a single node.

---

[6] https://cassandra.apache.org/doc/latest/cassandra/operating/hardware.html
[7] https://docs.datastax.com/en/cassandra-oss/3.0/cassandra/operations/opsAboutSnapshots.html
[8] https://cassandra.apache.org/doc/latest/cassandra/architecture/storage_engine.html
[9] https://docs.datastax.com/en/cassandra-oss/2.1/cassandra/dml/dml_about_hh_c.html
[10] https://cassandra.apache.org/doc/latest/cassandra/operating/hardware.html

- **Replication Factor**: Aerospike requires 2 copies of the data to guarantee consistency and availability
- **Compression Ratio**: Assume the data can be compressed to 30% of the original size.
- **Operational Requirements**: Aerospike requires 50% of the disk to be empty to minimise write amplification.

In August 2021, Aerospike published a research study showing the (virtual) hardware required to store 1PB of data in AWS[11]. 20 * i3en.24xlarge instances were required. Results are summarised in the table below.

| | Aerospike |
|---|---|
| Unique data size (TB) | 1000 |
| Replication Factor | 2 |
| Total Data Size (TB) | 2000 |
| Compression Ratio | 30% |
| Size of data on Disk (TB) | 600 |
| Disk Space Required (TB) | 1200 |
| Density per node (TB) | 30 |
| Number of nodes | 20 |

Detail of the i3en.24xlarge instance type is given below:

| **Instance** | Cores | RAM (GB) | Number of disks | Disk Size (TB) |
|---|---|---|---|---|
| i3en.24xlarge | 96 | 768 | 8 | 7500 |

## Physical Resource Estimation Summary

| **Instance** | Cores | RAM (GB) | Number of disks | Disk Size | Required number of nodes |
|---|---|---|---|---|---|
| c4.4xlarge (Cassandra on EBS) | 16 | 30 | EBS | 2500 | 900 |
| i3.4xlarge (Cassandra on NVMe) | 16 | 122 | 2 | 1900 | 900 |
| i3en.24xlarge (Aerospike) | 96 | 768 | 8 | 7500 | 20 |

The table above shows that each Aerospike node requires significantly more resources than the Apache Cassandra nodes. But as you can see in the table below, the total amount of hardware

---

[11] https://aerospike.com/lp/running-operational-workloads/

used in the entire solution is nevertheless clearly significantly smaller for Aerospike for all of CPU, RAM, and disk volume.

| **Instance** | Total number of cores | Total Ram (TB) | Total disk (TB) |
|---|---|---|---|
| Cassandra on EBS | 14,400 | 27 | 6,750 |
| Cassandra on NVMe | 14,400 | 109.8 | 3,420 |
| Aerospike | 1,920 | 15.36 | 1,200 |

# Estimating the Emissions Produced by the Infrastructure

Now that we know the underlying resources required by each solution, we can estimate the $CO_2$ emissions that each solution produces over a one year period.

## EC2 Emissions

At the time of writing, June 2022, none of the cloud providers allowed forecasting of $CO_2$ emissions based on estimated consumption. Their tools only allow users to monitor the report of actual emissions, with a 3-month delay. Additionally, the reported numbers by the cloud providers are not comparable across providers because they are calculated using different methodologies.

Because of these limitations, I have taken advantage of the excellent work of Benjamin DAVY and his colleagues from Teads. In this series of articles, they explain why and how they decided to independently estimate the $CO_2$ emissions of AWS infrastructure.

- Evaluating the carbon footprint of a software platform hosted in the cloud
- Estimating AWS EC2 Instances Power Consumption
- Building an AWS EC2 Carbon Emissions Dataset

Teads has published a tool that estimates the manufacturing and energy consumption emissions of different EC2 instances in each region. You can make use of the tool yourself - see Carbon footprint estimator for AWS instances - Teads.

It is important to acknowledge the limitations of the dataset (this article lists them in the “Foreword on limitations” section). Due to a lack of information, the emissions produced by some components are omitted. Most notable omissions are related to the data centre facilities, networking equipment, and storage hardware.

Because one of the most important components that databases use is storage, I have estimated the storage emissions using data that hardware manufacturers publish and included them in my calculations. Nevertheless, the emissions produced by the other omitted components are not negligible. Therefore, this study's estimates can only be treated as a lower bound.

Choice of the region affects both cost and $CO_2$ emissions. Depending on the *source* of energy that an AWS region uses, the $CO_2$ emissions can vary significantly. As an example, according to to the calculator, the emissions produced by the energy consumption of instances in eu-west-1 (Ireland) are around 2.5x lower than those in the me-south-1 (Bahrain) region. In this analysis, I used Ireland as the basis for my estimations as it is the most popular AWS region for UK businesses.

Using the estimator tool, the emissions of EC2 instance types used in each solution are:

| **Instance** | EC2 Annual Manufacturing Emissions ($kgCO_2eq$) | EC2 Annual Energy Consumption Emissions ($kgCO_2eq$) |
|---|---|---|
| c4.4xlarge (Cassandra on EBS) | 117.384 | 365.73 |
| i3.4xlarge (Cassandra on NVMe) | 145.416 | 356.43 |
| i3en.24xlarge (Aerospike) | 746.352 | 3,042.43 |

## Storage Emission

As discussed, the Teads estimator does not include the emissions related to storage hardware. In this section, using data published by disk manufacturers, I estimate the emissions produced by the manufacturing and usage of disk drives. Also included is a rough estimate of emissions produced by EBS.

### Manufacturing Emissions

Samsung has reported their $CO_2$ emissions for SSD Manufacturing[12], which they believe to be the lowest in the industry. 204.67 $kgCO_2eq$ emissions per year were reported for their 8 TB SSDs. We can conclude that manufacturing 1 TB of SSD produces at least:

$204.67 \div 8 = 25.58\ kgCO_2eq$ per 1 TB per year.
Because the EBS replication factor is 3, the emissions produced by the manufacturing of each TB of SSD drives used in EBS would be:

$25.58 \times 3 = 76.75\ kgCO_2eq$ per year.

To estimate the manufacturing emissions of the other components of EBS (processors, memory, etc.) I add a 30% surplus. Hence the total emissions of EBS infrastructure per TB of data in a year are:

$76.75 + 30\% = 99.78\ kgCO_2eq$ per year.

| | |
|---|---|
| **EBS Manufacturing Emission Estimation** | |

---

[12] https://news.samsung.com/global/samsung-receives-the-industrys-first-environmental-product-declaration-certificate-for-512gb-v-nand-and-860-evo-4tb-ssd

| | |
|---|---|
| Samsung 8TB SSD Manufacturing Emissions per year ($kgCO_2eq$) | 204.67 |
| SSD Manufacturing Emissions per TB per year ($kgCO_2eq$) | 25.58 |
| Annual SSD Manufacturing Emissions per 1TB capacity on EBS ($kgCO_2eq$) | 76.75 |
| EBS additional surplus | 30% |
| Estimated Annual Manufacturing Emissions per 1TB of Capacity on EBS ($kgCO_2eq$) | 99.78 |

## Energy Consumption Emissions

According to Seagate, the average annual energy usage of their SSD drives is 6.7 kWh per TB[13]. Emissions produced per kWh of energy in Ireland are equal to 0.316 $kgCO_2$/kWh[14]. Therefore the average annual emissions per 1 TB of SSD is:

$6.7 \times 0.316 = 2.12\ kgCO_2eq$ per year.

Considering the replication factor, the annual emissions caused by the energy usage of EBS SSD drives are:

$2.11 \times 3 = 6.35\ kgCO_2eq$ per year.

And if we similarly assume a 30% surplus for the other components of EBS (processors, memory, etc.). The total annual energy usage of EBS per TB is:

$6.35 + 30\% = 8.26\ kgCO_2eq$ per year.

| **EBS Energy Consumption Emission Estimation** | |
|---|---|
| Average Annual Energy usage of SSD ((kWh) | 6.7 |
| $CO_2$ Emissions per kWh energy usage in Ireland (kg/kWh) | 0.316 |
| Annual SSD Energy Consumption Emissions per TB ($kgCO_2eq$) | 2.12 |
| EBS replication Factor | 3 |
| Annual SSD Energy Consumption Emissions for EBS per TB ($kgCO_2eq$) | 6.35 |
| Estimated Annual Energy Consumption Emissions for EBS per TB ($kgCO_2eq$) | 8.26 |

---

[13] https://www.seagate.com/gb/en/global-citizenship/product-sustainability/nytro-1551-sustainability-report/

[14] https://www.cloudcarbonfootprint.org/docs/methodology/

## Estimated Infrastructure Emissions Summary

Putting all of the above together, the detail of the emissions of AWS resources used by each solution is:

| Instance | Annual EC2 Instance Manufacturing Emissions ($kgCO_2eq$) | Annual EC2 Instance Energy Consumption Emissions ($kgCO_2eq$) | Annual Storage Manufacturing Emissions ($kgCO_2eq$) | Annual Storage Energy Consumption Emissions ($kgCO_2eq$) |
|---|---:|---:|---:|---:|
| c4.4xlarge (Cassandra on EBS) | 117.384 | 365.73 | 249.44 | 20.64 |
| i3.4xlarge (Cassandra on NVMe) | 145.416 | 356.43 | 97.22 | 8.05 |
| i3en.24xlarge (Aerospike) | 746.352 | 3,042.43 | 1,535.03 | 127.03 |

Hence for each solution the total emissions would be:

| Solution | Number of nodes | Tota Annual EC2 Manufacturing Emissions ($kgCO_2eq$) | Total Annual EC2 Energy Consumption Emissions ($kgCO_2eq$) | Total Annual Storage Manufacturing Emissions ($kgCO_2eq$) | Total Annual Storage Energy Consumption Emissions ($kgCO_2eq$) | Total Annual Emissions ($kgCO_2eq$) |
|---|---:|---:|---:|---:|---:|---:|
| Cassandra on EBS | 900 | 105,646 | 329,156 | 224,497 | 18,578 | 677,878 |
| Cassandra on NVMe | 900 | 130,874 | 320,785 | 87,496 | 7,241 | 546,397 |
| Aerospike | 20 | 14,927 | 60,849 | 30,701 | 2,541 | 109,017 |

The results are surprising. As you can see, the choice of hardware can reduce the CO_2 emissions of a solution by 20% (Cassandra: 678 → 546). But perhaps more interestingly, nominally similar solutions may have order of magnitude differences for CO_2 emissions ( Apache Cassandra Vs Aerospike: 678 → 109). If you find this result puzzling, let me remind you of the amount of resources each solution requires:

| Instance | Total Number of nodes | Total Number of cores | Total Ram (TB) | Total Disk (TB) |
|---|---:|---:|---:|---:|
| Cassandra on EBS | 900 | 14,400 | 27 | 6,750 |
| Cassandra on NVMe | 900 | 14,400 | 109.8 | 3,420 |
| Aerospike | 20 | 1,920 | 15.36 | 1,200 |

# Cost of the Infrastructure on AWS

As explained initially, more efficient software requires less hardware, produces fewer emissions, and is cheaper. So far, we have seen that one of the solutions requires less hardware and produces fewer emissions. This section investigates the cost of running each of these solutions over a year.

The cost of the required AWS resources for each of these solutions are:

| **Instance** | Instance Hourly Rate | EBS Monthly Rate[15] |
|---|---|---|
| c4.4xlarge (Cassandra on EBS) | $0.91 | $258.64* |
| i3.4xlarge (Cassandra on NVMe) | $1.38 | 0 |
| i3en.24xlarge (Aerospike) | $12.00 | 0 |

*: The EBS cost is estimated for EBS on SSD with a minimum of 10k IOPS which is DataStax recommendation[16].

Therefore, the total cost of ownership for each solution is:

| | Number of Nodes | Total Instance Cost per Hour | EBS Cost per month | Annual TCO |
|---|---|---|---|---|
| Cassandra on EBS | 900 | $815 | $232,776 | $9,928,332 |
| Cassandra on NVMe | 900 | $1,238 | $0 | $10,848,384 |
| Aerospike | 20 | $240 | $0 | $2,102,400 |

The solution with order of magnitude lower $CO_2$ emissions is also a fraction of the cost of alternative solutions.

# Conclusion

At the beginning of this article, I promised to answer two questions that could help IT decision-makers understand the environmental impact of their chosen stack of technology.

1. What is the impact of the $CO_2$ emission savings that a switch between two nominally similar technologies can cause?
2. What is the cost of a $CO_2$ Emission Efficient technology compared to its rivals?

Let's revisit these.

## Impact

It is worth asking how significant 500 tonnes of $CO_2$ emissions per year is.

500 tonnes of $CO_2$ emissions is vanishingly small compared to the estimated 36.4 billion tonnes of global emissions in 2021[17]. Nevertheless, there are significant constructive perspectives about this saving:

---

[15] https://calculator.s3.amazonaws.com/index.html
[16] https://docs.datastax.com/en/dse-planning/doc/planning/planningEC2.html
[17] https://www.statista.com/statistics/276629/global-co2-emissions/

- It is estimated that each hectare of trees absorbs 10 tonnes of $CO_2$ per year[18]. Therefore, reducing 500 tonnes of $CO_2$ emissions per year is equivalent to planting 50 hectares of trees.
- If, as predicted, stored data volumes double every two years[19], the amount of $CO_2$ emission saving would also be doubled every two years if the more efficient technology is used.
- The estimated global size of stored data in 2022 is 97 zetabytes[20] (97 million petabytes). Depending on the portion of the data stored in databases globally, there is an opportunity to reduce millions of tonnes of $CO_2$ emissions just by switching to more efficient data management solutions.

## Cost

Sometimes efficiency has a price. For example, in the case of Apache Cassandra, the solution with the least amount of $CO_2$ emissions is more expensive, and the cheaper solution is more polluting than the alternative.

| | Total Annual Emissions ($kgCO_2eq$) | Annual TCO |
|---|---:|---:|
| Cassandra on NVMe | 546,397 | $10,848,384 |
| Cassandra on EBS | 677,878 | $9,928,332 |

However, the efficiency of platforms based on different underlying technologies can be dramatically different. As we saw in this comparison, a more efficient software solution can be many times cheaper and less polluting.

| | Total Annual Emissions ($kgCO_2eq$) | Annual TCO |
|---|---:|---:|
| Cassandra on NVMe | 546,397 | $10,848,384 |
| Aerospike | 109,017 | $2,102,400 |

Although the purpose of this article is not to compare the “performance” of these technologies, it is worth mentioning that the expected latency of Apache Cassandra is in the region of a single-digit millisecond (<10ms), whereas Aerospike works in less than a millisecond latency range (<1ms). Yet another substantial improvement.

---

[18] https://onetreeplanted.org/blogs/stories/how-much-co2-does-tree-absorb

[19] https://medium.com/callforcode/the-amount-of-data-in-the-world-doubles-every-two-years-3c0be9263eb1

[20] https://www.statista.com/statistics/871513/worldwide-data-created/

## Last Word

In this article, I make the case for the importance of considering $CO_2$ emissions efficiency as a Non-Functional requirement. The data shows the considerable positive effects of good practice, and conversely the negative effects of ignoring $CO_2$ production.

I hope the framework I suggest can be used to compare other similar technologies. Going beyond this, we might consider a universal $CO_2$ emissions metric that measures different technologies based on $CO_2$ emissions. Such a metric would allow IT decision-makers to incorporate environmental considerations when choosing technology components. It would also encourage technology vendors to reduce the environmental impact of their products.

The results also show that when reducing environmental impact, you can also reduce costs - we don't necessarily have to choose between one or the other. We can satisfy our budgets and our consciences at the same time.

## About Aerospike

The Aerospike Real-time Data Platform enables organizations to act instantly across billions of transactions while reducing cloud infrastructure up to 80%. The Aerospike data platform powers real-time applications with predictable sub-millisecond performance up to petabyte scale with five-nines uptime with globally distributed, strongly consistent data. Applications built on the Aerospike Real-time Data Platform fight fraud, provide recommendations that dramatically increase shopping cart size, enable global digital payments, and deliver hyper-personalized user experiences to tens of millions of customers. Customers such as Airtel, Experian, Nielsen, PayPal, Snap, Wayfair and Yahoo rely on Aerospike as their data foundation for the future.

For more information, please visit https://www.aerospike.com.