IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 26 July 2022, accepted 29 August 2022, date of publication 8 September 2022, date of current version 20 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205016

RESEARCH ARTICLE

CO_2 Emission Efficiency as a Measurable Non-Functional Requirement: An Emission Estimation Framework

BEHRAD BABAEE[1], DAMON DAYLAMANI-ZAD[2], AND KEN TUNE[1]
[1]Aerospike, EC2N 1AR London, U.K.
[2]College of Engineering, Design and Physical Sciences, Brunel University London, UB8 3PH London, U.K.
Corresponding author: Behrad Babaee (behrad@aerospike.com)

ABSTRACT Environmental concerns and the impact of technology on climate change are now a global concern. To this effect, reducing CO_2 emission is one of the factors that has been the focus of researchers, activists, and governments. This has been included in the UN's 2030 Agenda for Sustainable Development. The emissions produced due to running software have remained largely unquantified or neglected in Carbon Accounting. This paper proposes "CO_2 Emission Efficiency" as a measurable Non-Functional requirement. We propose a framework for estimating software's CO_2 emissions and compare two well-known databases, Apache Cassandra and Aerospike. The paper presents the method, process, and comparisons. The paper then considers the costs of each of these technologies and concludes that reducing CO_2 emissions not only has a positive impact on the environment but can also be cheaper and reduce costs.

INDEX TERMS Aerospike, Apache Cassandra, CO_2 emissions, Carbon Accounting, net-zero target, non-functional requirements, AWS, efficiency.

I. INTRODUCTION

Concern over mankind's impact on the environment and the unchecked effects of climate change is the dominant global issue of our time. Burning 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 - its consumption impact is substantial and growing.

The impact is more significant 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. This also falls in with the UN's 17 Sustainable Development Goals (SDGs), which are an urgent call for action by all countries - developed and developing - in a global partnership [4].

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.

II. RELATED STUDY

Until recently, emissions of running software platforms were largely unquantified and mostly neglected in Carbon Accounting [5]. This is changing however, and perhaps the best evidence of this is that the three major cloud technology providers (AWS, Azure, and GCP) are now reporting the CO_2 emissions produced by the consumption of their resources

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

by individual customers [6], [7], [8]. Therefore, there are no technical reasons for not accounting for these emissions in consumer CO_2 balance sheets, and there will likely 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 framework 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?
2) What is the cost of a CO_2 Emission Efficient technology compared to its rivals?

A. SCALABILITY

Before the advent of the Internet and the rise in availability of distributed, scalable computing resources, efficient use of computing resources was mandatory [9]. 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 [10].

Scalable resources allow us to address performance-related non-functional requirements such as latency, throughput, capacity, and growth effectively, but not efficiently [10]. 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 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.

B. EFFICIENCY

In computer science, efficiency is generally evaluated in terms of CPU usage [11], [12], [13]. An algorithm is more efficient than another if it uses fewer CPU cycles to solve a problem. Therefore, 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 Total Cost of Ownership (TCO). In the context of this research, we only consider the total cost of infrastructure ownership as TCO.

To illustrate the point, let us 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 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.

III. METHODOLOGY: EMISSION ESTIMATION FRAMEWORK

This article proposes a framework for estimating CO_2 emission of software systems and their TCO, consolidated into CO_2 Emission Efficiency. This approach can be applied to various software systems and technologies.

This framework has three stages; 1) Technology and workload selection, 2) Resource estimation, and 3) Emission and cost estimation. Once the technology/technologies are selected, their requirements are used to estimate the physical resources needed to operate the software. The resource requirements are then used to estimate the operational cost of the software and the corresponding CO_2 emission. The proposed framework is presented in Fig. 1.

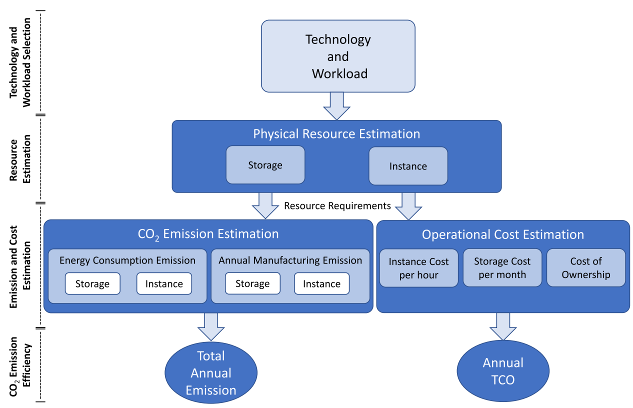

FIGURE 1. Proposed framework for estimating CO_2 emission.

In stage 1, the technology and the workload, tw, are identified. Then in stage 2, using best practices for the technology, the minimum physical resources required for handling the workload, R_{tw}, are estimated, which is presented in (1).

The minimum physical resource estimation is a function of the workload, and depends on the data size, $|data_{tw}|$, the processing power, $|comp_{tw}|$, the memory complexity, $|mem_{tw}|$, and overheads such as networking, monitoring, and cooling, $|overheads_{tw}|$ (1).

$$R_{tw} = |data_{tw}| + |comp_{tw}| + |mem_{tw}| + |overheads_{tw}| \quad (1)$$

In practice, the physical resources are determined by the number of instances of a specific type, $n(nodes)$, and the capacity of the data storage equipment for storing the data size, $C(storage)$, needed to accommodate the technology and the workload. Hence, in most cases in this framework, (1) is in practice equivalent to (2).

$$R_{tw} = n(nodes) + C(storage) \quad (2)$$

In stage 3, we estimate the technology's emission and the cost of ownership. The emission is the sum of the instance emission, em_ins_{tw}, and storage emission, em_str_{tw}, of the minimum physical resources required as presented in (3).

$$em_{tw} = em\_ins_{tw} + em\_str_{tw} \quad (3)$$

Emission of instances, em_ins_{tw}, is the sum of the energy consumption emission, $em\_ins\_con_{tw}$, and the manufacturing emissions, $em\_ins\_mf_{tw}$, which is presented in (4).

$$em\_ins_{tw} = em\_ins\_con_{tw} + em\_ins\_mf_{tw} \quad (4)$$

The instance energy consumption emission calculations are based on [14], [15], and [16] and presented in (5), where IPC is Instance Power Consumption measured in Watts at 50% load, PUE is Data center Power Usage Effectiveness and ECI is Electricity Carbon Intensity of the instance region inr. Accordingly, H represents the computing hours. PUE is calculated as the total annual energy entering the data center building, E_{dc}, divided by the annual energy consumed to operate devices of the IT room (E_{ITR}).

$$\begin{aligned} em\_ins\_cons_{tw} &= \left(\frac{IPC_{inr}}{1000} \times PUE_{inr} \times ECI_{inr}\right) \times H \\ PUE &= \frac{E_{dc}}{E_{ITR}} \end{aligned} \quad (5)$$

Instance manufacturing emission is estimated using the manufacturer's published data or other publicly available data sources, $em\_ins\_mf_{node}$ per $node$ per year multiplied by the number of nodes, $n(nodes)$. This is presented in (6)

$$em\_ins\_mf_{tw} = em\_ins\_mf_{node} \times n(nodes) \quad (6)$$

The estimated emission of storage, em_str_{tw}, is equal to the sum of the manufacturing emission, $em\_str\_mf_{w}$, the energy consumption emission, $em\_str\_cons_{tw}$, and the emissions produced by the other extra resources, $em\_str\_ex_{tw}$, required by the storage technology (7).

$$em\_str_{tw} = em\_str\_con_{tw} + em\_str\_mf_{tw} + em\_str\_ex_{tw} \quad (7)$$

The storage energy consumption emission is estimated and presented in (8), where $str\_eng\_con_{TB}$ is the manufacturer's published storage consumption per terabyte of disk per year, em_eng_{region} is emissions produced per unit of energy used in the region, and d is the size of required disk.

$$em\_str\_con_{tw} = str\_eng\_con_{TB} \times em\_eng_{region} \times d \quad (8)$$

The storage manufacturing emission is estimated and presented in (9), where $em\_str\_mf_{TB}$ is the manufacturer's published emission per terabyte of disk per year, and d is the size of required disk.

$$em\_str\_mf_{tw} = em\_str\_mf_{TB} \times d \quad (9)$$

The storage emission produced by the other resources is only relevant if an external storage technology is used. In case the specification of the storage technology is known, (10) can be used to estimate the emission for extra resources. Otherwise, a percentage of energy consumption and manufacturing emissions of the disks, S_{tw}, can be factored in as a surplus.

$$em\_srt\_ex_{tw} = (em\_str\_con_{tw} + em\_str\_mf_{tw}) \times S_{tw} \quad (10)$$

The annual total cost of ownership (TCO) for each technology is estimated based on the cost of running an instance per node added to external storage cost such as EBS. This is demonstrated in (11). This is based on the number of nodes required, the hourly cost of running an instance (HC_i), the number of hours in a year (h), the monthly cost of external storage, such as EBS, costs (MC_{ov}) and the number of months in a year (m).

$$TCO_{tw} = n(nodes) \times HC_i \times h + C(storage) \times MC_{ov} \times m \quad (11)$$

Finally, the CO_2 Emission Efficiency is presented, using (12), as a combination of emission per workload and cost per workload.

$$CO_2EmEffic_{tw} = \left[\frac{em_{tw}}{workload}, \frac{TCO_{tw}}{workload}\right] \quad (12)$$

IV. APPLYING THE FRAMEWORK: DATABASES USE CASE

Almost all software solutions deal with some kind of data. Data requirements form an integral and essential part of any requirement analysis for software systems [17], [18], [19]. Data would require storage and processing, making it a suitable use case to examine the proposed framework. To this effect, the following sections will adopt two distributed database technologies for estimating their CO_2 emission, their TCO, and compares the CO_2 emission efficiencies.

In this use case, we compare the two software platforms, Apache Cassandra and Aerospike.

For each of the database technologies under investigation, the following stages are discussed for a fixed workload:

1) **Physical Resource Estimate**: Minimum recommended hardware if using AWS.
2) **Emission and Cost Estimation**
   a) **Emissions Estimate**: This includes emissions produced during the hardware manufacturing process and emissions generated by operational use.

b) **Operational Cost**: The monetary cost of deploying versus the AWS platform.

A. PHYSICAL RESOURCE ESTIMATION FOR DISTRIBUTED DATABASES

To estimate physical resource for a distributed database, the size of the unique data, replication factor (RF), compression ratio (CR) and density per node is required. Using this information, the total data size, the size of data on disk, and finally the number of nodes required can be deduced. These are calculated using (13), (14) and (15).

$$|total\_data| = |unique\_data| \times RF \tag{13}$$

$$|data\_on\_disk| = |total\_data| \times CR \tag{14}$$

$$n(nodes) = \frac{|data\_on\_disk|}{density\_per\_node} \tag{15}$$

B. OPERATIONAL COST ESTIMATION FOR DISTRIBUTED DATABASES

As the world's most popular cloud computing platform, AWS is a common logical choice when comparing relative costs. Hence, the AWS platform was chosen to use benchmark data already sourced using the platform. We used AWS on-demand pricing for the Ireland region, a popular AWS region for UK businesses. Also, the CO_2 emissions of AWS EC2 instances were available. Finally, this allows the reported results to be reproduced, examined and evaluated.

V. STAGE 1: TECHNOLOGY AND WORKLOAD SELECTION

The framework's first stage is selecting the technology and the workload for the estimation and comparison. As mentioned, the use case for this article would be database technologies. In this section, we will discuss the selection process for the technologies and their corresponding workload.

A. THE TECHNOLOGY

The research team has expertise in databases and extensive experience with well-known databases Apache Cassandra and Aerospike. This article estimates and compares the CO_2 emission differences of these nominally similar technologies as a use case. Based on this selection, the methodology has been revised, Fig. 2, to reflect the specific aspects of these technologies. However, the framework introduced in this article could be used for comparing other software platforms.

B. THE WORKLOAD

To compare these databases, we need to specify fixed workloads and compare the resulting emissions and costs. 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 favour one platform over the other.

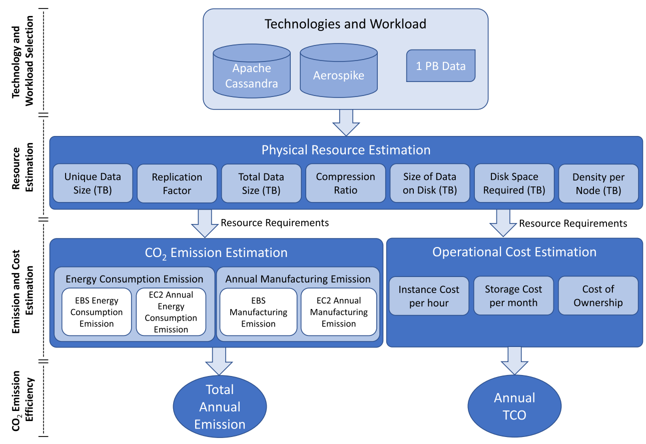

FIGURE 2. Applying framework to estimate CO_2 emission for Apache Cassandra and Aerospike.

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 1 PB use case can be readily converted to those arising from larger or smaller data volumes.

VI. STAGE 2: PHYSICAL RESOURCE ESTIMATE

This section determines the required AWS hardware to store 1 PB of raw data using the vendors' best practice guides. Assumptions, calculations and disk choices for the three chosen databases are presented and discussed.

A. APACHE CASSANDRA

1) 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 [20].
- **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.

2) CALCULATIONS

The physical resources required for 1 PB of data and the assumptions above are calculated for each technology to determine the number of nodes required for the data on a specific database system. Table 1 presents the required resources for 1 PB data on Apache Cassandra, and the calculated required number of nodes using equations (13), (14) and (15).

TABLE 1. Calculations for Apache Cassandra.

	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

3) 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 [21]. 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 [22], commitlog [23], hinted handoffs [24], and other Cassandra overheads.

When storing the data on ephemeral storage, i3 instances with attached NVMes are recommended [21]. 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 storing 1PB of data are summarised in Table 2.

TABLE 2. Resulting instance type choices for storing 1PB of data with Apache Cassandra.

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

The recommendations available on Apache Cassandra and DataStax respective websites regarding the instance types might not the best options. Using more modern equivalent instance types like c6g and i3en, the cost and emissions could be reduced by up to 20% without affecting the performance. To avoid subjectivity however, as the research aims to be reference-able and reproducible, the standard recommendations were used. Although an additional 20% saving is significant, it does not change the conclusions of this article.

B. AEROSPIKE

1) ASSUMPTIONS

- **Data Density**: The theoretical per node limit for disks using Aerospike is 256 TB. In practice, the limit is the disk capacity that can be attached to a single node.
- **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 its original size.
- **Operational Requirements**: Aerospike requires 50% of the disk to be empty to minimise write amplification.

2) CALCULATIONS

As mentioned, Aerospike has a theoretical density limit of 256 TB per node. Yet, AWS does not provide instances with NVMe disks larger than 60 TB. Since Aerospike requires 50% empty disk space, the highest practical density per node is limited to 30 TB. These can be used to estimate the hardware requirements using (13), (14) and (15) summarised in table 3.

TABLE 3. Calculations for Aerospike.

	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

Additionally, in August 2021, Aerospike published a research study demonstrating the hardware required to store 1PB on AWS [25]. It found that 20 × i3en.24xlarge instances were required, matching the estimation above.

3) DISK CHOICES

Detail of the i3en.24xlarge instance type is presented in Table 4.

TABLE 4. Resulting instance type choice for storing 1PB of data with Aerospike.

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

C. PHYSICAL RESOURCE ESTIMATION DISCUSSION

Table 5 compares the required node resources required for each database. It shows that each Aerospike node requires significantly more resources than the Apache Cassandra nodes. However, table 6, the total amount of hardware used in the entire solution is nevertheless clearly significantly smaller for Aerospike for all of CPU, RAM, and disk volume.

TABLE 5. Required resources for storing 1PB.

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

TABLE 6. Total amount of hardware used in the entire solution for each database.

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

VII. STAGE 3A: ESTIMATING THE EMISSIONS PRODUCED BY THE INFRASTRUCTURE

By establishing the underlying resources required by each solution, it is possible to estimate the CO_2 emissions each solution produces over one year.

A. CO_2 EMISSIONS

Currently, none of the cloud providers allowed forecasting of CO_2 emissions based on estimated consumption. Their tools only enable 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.

Due to these limitations, we have adopted the previous research done for evaluating carbon footprint [26], power consumption [14], [16] and Carbon Emissions dataset [27]. This series of articles explains why and how they decided to independently estimate the CO_2 emissions of AWS infrastructure. Teads Engineering has published a tool [28] that estimates the manufacturing and energy consumption emissions of different EC2 instances in each region. This approach has been based on previous work, which presents comparative evaluations of power models in data-centers [29] and cloud servers [30].

It is important to acknowledge the limitations of the dataset [27]. 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.

Since one of the essential components that databases use is storage, the storage emissions were estimated using data that hardware manufacturers publish and included in the 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. For example, according to the calculator, the emissions produced by the energy consumption of instances in eu-west-1 (Ireland) are around 2.5 times lower than those in the me-south-1 (Bahrain) region. In this analysis, Ireland was used as the basis for estimations as it is the most popular AWS region for UK businesses.

TABLE 7. CO_2 emissions of EC2 instance types used in each solution calculated using the estiamtor tool.

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

TABLE 8. Manufacturing CO_2 emissions of a typical SSD drive.

SSD Manufacturing Emission Estimation	
Samsung 8TB SSD Manufacturing Emissions per year ($kgCO_2eq$)	204.67
SSD Manufacturing Emissions per 1TB per year ($kgCO_2eq$)	25.58

TABLE 9. Manufacturing CO_2 emissions of EBS.

EBS Manufacturing Emission Estimation	
EBS replication factor	3
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

TABLE 10. SSD energy consumption emission estimation.

SSD 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

Using the estimator tool, equation (4), the emissions of EC2 instance types used in each solution are presented in table 7.

TABLE 11. EBS energy consumption emission estimation.

EBS Energy Consumption Emission Estimation	
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

TABLE 12. Estimated annual infrastructure emissions for each solution.

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

TABLE 13. Estimated total annual emissions for each solution.

Solution	Total Number of nodes	Total 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

B. STORAGE EMISSIONS

As discussed, the Teads estimator does not include the emissions related to storage hardware. In this section, the emissions produced by the manufacturing and usage of disk drives are estimated using data published by disk manufacturers and (7). This section also includes a rough estimate of emissions produced by EBS for Apache Cassandra on EBS.

1) MANUFACTURING EMISSIONS

Samsung has reported their CO_2 emissions for SSD Manufacturing [31], 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 $204.67 \div 8 \approx 25.58$ $kgCO_2eq$ per 1 TB per year. The results are summarised in table 8.

For EBS emission estimation, we need to consider the replication factor. The EBS replication factor is 3. Hence, the emissions produced by manufacturing each TB of SSD drives used in EBS would be $25.58 \times 3 = 76.75$ $kgCO_2eq$ per year. Since the specification of EBS architecture is not public, a 30% surplus is added (10) to the sum to account for the manufacturing emissions of the other components of EBS (processors, memory, etc.). Hence the total emission of EBS infrastructure per TB of data in a year is: $76.75 + 30\% = 99.78$ $kgCO_2eq$ per year. The final results are summarised in table 9.

2) ENERGY CONSUMPTION EMISSIONS

According to Seagate, the average annual energy usage of their SSD drives is 6.7 kWh per TB [32]. Emissions produced per kWh of energy in Ireland are equal to 0.316 $kgCO_2/kWh$ [33]. Therefore the average annual emissions per 1 TB of SSD is $6.7 \times 0.316 \approx 2.12$ $kgCO_2eq$ per year. Results are presented in table 10.

Considering the replication factor, the annual emissions caused by the energy usage of EBS SSD drives are $2.12 \times 3 \approx 6.35$ $kgCO_2eq$ 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. The final results are summarised in table 11.

C. ESTIMATED INFRASTRUCTURE EMISSIONS DISCUSSION

Putting all of the above together, the detail of the emissions of AWS resources used by each solution is presented in table 12. Based on these findings, it is possible to calculate the total emission for each database. Total emissions are presented in table 13.

The results are noteworthy. As presented, 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 (Cassandra vs Aerospike: 678 → 109). They may seem puzzling until we consider each solution's minimum amount of resources as presented in tables 5 and 6. Once these two tables are merged, as presented in table 14, the results are clearly explained.

TABLE 14. Total amount of resources required in the entire solution for each database.

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

VIII. STAGE 3B: 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 solution over a year. The cost of the required AWS resources for each solution is presented in table 15. Based on these, each solution's total cost of ownership is calculated and presented in table 16.

Based on the data presented in this section, it is evident that the solution with an order of magnitude lower CO_2 emissions is also a fraction of the cost of alternative solutions.

TABLE 15. Total cost of of each database on AWS* The EBS cost is estimated for EBS on SSD with a minimum of 10k IOPS which is DataStax's recommendation [15].

Instance	Instance Hourly Rate	EBS Monthly Rate [34]
c4.4xlarge (Cassandra on EBS)	$0.91	$258.64*
i3.4xlarge (Cassandra on NVMe)	$1.38	0
i3en.24xlarge (Aerospike)	$12.00	0

TABLE 16. Total cost of ownership for each database.

Instance	Total number of nodes	Total Instance Cost per hour	EBS Cost per month	Annual TCO
Cassandra EBS	900	$815	$232,776	$9,928,332
Cassandra NVMe	900	$1,238	$0	$10,848,384
Aerospike	20	$240	$0	$2,102,400

IX. CONCLUDING DISCUSSIONS

This research aimed to answer two questions that could help researchers and IT decision-makers understand their chosen technology stack's environmental impact.

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

The discussions presented in this article allow us to arrive at conclusions which answer these questions. We will consider these under impact and cost.

A. IMPACT OF REDUCING CO_2 EMISSIONS

It is considering how significant 500 tonnes of CO_2 emissions per year is. Whilst 500 tonnes of CO_2 emissions is vanishingly small compared to the estimated 36.4 billion tonnes of global emissions in 2021 [35], nevertheless, there are significant constructive perspectives to this saving:

- It is estimated that each hectare of trees absorbs 10 tonnes of CO_2 per year [36]. 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 [37], 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 zettabytes [38] (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.

B. CO_2 EMISSION EFFICIENCY: EMISSION VS COST

Sometimes efficiency has a price. For example, in the case of Apache Cassandra-based platforms, the solution with the least amount of CO_2 emissions is more expensive, while the cheaper solution is more polluting. Table 17 presents the CO_2 Emission Efficiency for 1 PB workload (12) in terms of annual TCO and emissions, illustrating the trade-off between emission efficiency and costs.

TABLE 17. CO_2 emission efficiency for 1 PB workload: Cassandra disk choice storage comparison.

	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. Table 18 presents CO_2 Emission Efficiency for 1 PB workload (12), illustrating the case where emission efficiency and cost are complementary. Fig. 3 presents the CO_2 Emission Efficiency for all three options by plotting the Total Annual Emission against their Annual TCO.

TABLE 18. CO_2 emission efficiency for 1 PB workload: Cassandra vs Aerospike.

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

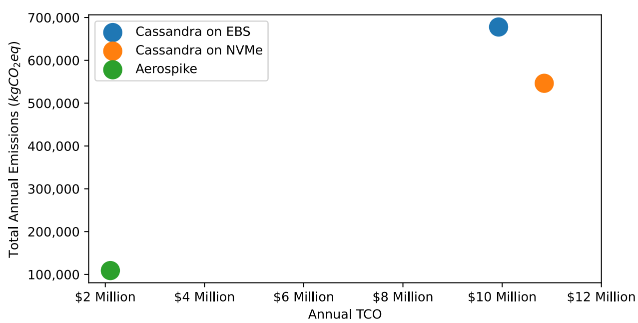

FIGURE 3. CO_2 emission efficiency: complete picture.

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

C. FINAL WORDS AND FUTURE DIRECTIONS

In this article, a case was made for the importance of considering CO_2 emissions efficiency as a Non-Functional requirement for IT systems. The data shows the considerable positive effects of good practice and, conversely, the adverse effects of ignoring CO_2 production.

It is hoped that the framework suggested here 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 reducing environmental impact would 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. In summary, this article presents three contributions:

1) A framework for estimating and comparing the CO_2 emissions of software,
2) Introducing CO_2 Emission Efficiency as a measurable non-functional requirement,
3) Demonstrating that reducing the environmental impact of software would also reduce its costs.

ACKNOWLEDGMENT

This work would not have been possible without the public dataset provided by Teads on the estimated CO_2 emissions of AWS resources. The authors also thank Benjamin Davy of Teads for providing valuable feedback on this research.

REFERENCES

[1] G. Blanco, R. Gerlagh, S. Suh, J. Barrett, H. C. De Coninck, C. D. Morejon, R. Mathur, N. Nakicenovic, A. O. Ahenkorah, J. Pan, and H. Pathak, ''Drivers, trends and mitigation,'' in *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the 5th Assessment Report of the Intergovernmental Panel on Climate Change*, O. Edenhofer, Ed. 2014, pp. 351–411.

[2] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, ''Chasing carbon: The elusive environmental footprint of computing,'' *IEEE Micro*, vol. 42, no. 4, pp. 37–47, Mar. 2022.

[3] A. Crawford, I. King, and D. Wu. (Apr. 2021). *The Chip Industry has a Problem With Its Giant Carbon Footprint*. Accessed: Jun. 23, 2022. [Online]. Available: https://www.bloomberg.com/news/articles/2021-04-08/the-chip-industry-has-a-problem-with-its-giant-carbon-footprint

[4] United Nations. (Jun. 2022). *Do you Know all 17 SDGS?* Accessed: Jun. 23, 2022. [Online]. Available: https://onetreeplanted.org/blogs/stories/how-much-co2-does-tree-absorb

[5] K. Stechemesser and E. Guenther, ''Carbon accounting: A systematic literature review,'' *J. Cleaner Prod.*, vol. 36, pp. 17–38, Nov. 2012.

[6] J. Barr. (Mar. 2022). *Customer Carbon Footprint Tool*. Accessed: Jun. 23, 2022. [Online]. Available: https://aws.amazon.com/blogs/aws/new-customer-carbon-footprint-tool/

[7] Microsoft Azure. (Jun. 2022). *Emissions Impact Dashboard*. Accessed: Jun. 23, 2022. [Online]. Available: https://appsource.microsoft.com/en-us/product/power-bi/coi-sustainability.emissions_impact_dashboard/

[8] Google Cloud. (Jun. 2022). *Carbon Footprint*. Accessed: Jun. 23, 2022. [Online]. Available: https://cloud.google.com/carbon-footprint

[9] T. C. Chieu, A. Mohindra, A. A. Karve, and A. Segal, ''Dynamic scaling of web applications in a virtualized cloud computing environment,'' in *Proc. IEEE Int. Conf. e-Bus. Eng.*, Oct. 2009, pp. 281–286.

[10] M. S. Rahman and H. Reza, ''Systematic mapping study of non-functional requirements in big data system,'' in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Jul. 2020, pp. 25–31.

[11] Y. Jin, Y. Wen, and Q. Chen, ''Energy efficiency and server virtualization in data centers: An empirical investigation,'' in *Proc. IEEE INFOCOM Workshops*, Mar. 2012, pp. 133–138.

[12] Y. Wen, Y. Wang, J. Liu, B. Cao, and Q. Fu, ''CPU usage prediction for cloud resource provisioning based on deep belief network and particle swarm optimization,'' *Concurrency Comput., Pract. Exp.*, vol. 32, no. 14, p. e5730, Jul. 2020.

[13] M. Wu, Q. Chen, and J. Wang, ''Toward low CPU usage and efficient DPDK communication in a cluster,'' *J. Supercomput.*, vol. 78, no. 2, pp. 1852–1884, 2022.

[14] D. Mytton, ''Assessing the suitability of the greenhouse gas protocol for calculation of emissions from public cloud computing workloads,'' *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–11, 2020.

[15] DataStax. (Jun. 2022). *Planning a DSE cluster on Amazon EC2*. Accessed: Jun. 23, 2022. [Online]. Available: https://docs.datastax.com/en/dse-planning/doc/planning/planningEC2.html

[16] B. Davy. (Mar. 2021). *Estimating AWS EC2 Instances Power Consumption*. Accessed: Jun. 23, 2022. [Online]. Available: https://medium.com/teads-engineering/estimating-aws-ec2-instances-power-consumption-c9745e347959

[17] K. Wiegers and J. Beatty, *Software Requirements*. London, U.K.: Pearson Education, 2013.

[18] A. Griva, S. Byrne, D. Dennehy, and K. Conboy, ''Software requirements quality: Using analytics to challenge assumptions at Intel,'' *IEEE Softw.*, vol. 39, no. 2, pp. 1–9, Mar. 2020.

[19] W. Maalej, M. Nayebi, and G. Ruhe, ''Data-driven requirements engineering—An update,'' in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Softw. Eng. Pract. (ICSE-SEIP)*, May 2019, pp. 289–290.

[20] DataStax. (Jun. 2022). *Capacity Planning and Hardware Selection for Apache Cassandra Implementations*. Accessed: Jun. 23, 2022. [Online]. Available: https://docs.datastax.com/en/cassandra-oss/planning/planning/ossCapacityPlanning.html

[21] Apache Cassandra. (Jun. 2022). *Hardware Choices*. Accessed: Jun. 23, 2022. [Online]. Available: https://cassandra.apache.org/doc/latest/cassandra/operating/hardware.html

[22] DataStax. (Jun. 2022). *About Snapshots*. Accessed: Jun. 23, 2022. [Online]. Available: https://docs.datastax.com/en/cassandra-oss/3.0/cassandra/operations/opsAboutSnapshots.html

[23] Apache Cassandra. (Jun. 2022). *Storage Engine*. Accessed: Jun. 23, 2022. [Online]. Available: https://cassandra.apache.org/doc/latest/cassandra/architecture/storage_engine.html

[24] DataStax. (Jun. 2022). *About Hinted Handoff Writes*. Accessed: Jun. 23, 2022. [Online]. Available: https://docs.datastax.com/en/cassandra-oss/2.1/cassandra/dml/dml_about_hh_c.html

[25] Aerospike. (Jun. 2022). *Petabyte Scale Benchmark*. Accessed: Jun. 23, 2022. [Online]. Available: https://aerospike.com/lp/running-operational-workloads/

[26] B. Davy. (Dec. 2020). *Evaluating the Carbon Footprint of a Software Platform Hosted in the Cloud*. Accessed: Jun. 23, 2022. [Online]. Available: https://medium.com/teads-engineering/evaluating-the-carbon-footprint-of-a-software-platform-hosted-in-the-cloud-e716e14e060c

[27] B. Davy. (Sep. 2021). *Building an AWS EC2 Carbon Emissions Dataset*. Accessed: Jun. 23, 2022. [Online]. Available: https://medium.com/teads-engineering/building-an-aws-ec2-carbon-emissions-dataset-3f0fd76c98ac

[28] Teads Engineering. (Jun. 2022). *Carbon Footprint Estimator for AWS Instances*. Accessed: Jun. 23, 2022. [Online]. Available: https://engineering.teads.com/sustainability/carbon-footprint-estimator-for-aws-instances/

[29] L. Ismail and H. Materwala, ''Computing server power modeling in a data center: Survey, taxonomy, and performance evaluation,'' *ACM Comput. Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[30] W. Lin, F. Shi, W. Wu, K. Li, G. Wu, and A.-A. Mohammed, ''A taxonomy and survey of power models and power modeling for cloud servers,'' *ACM Comput. Surveys*, vol. 53, no. 5, pp. 1–41, Oct. 2020.

[31] Samsung Newsroom. (Jun. 2018). *Samsung Receives the Industry's First Environmental Product Declaration Certificate for 512Gb V-NAND and 860 EVO 4TB SSD*. Accessed: Jun. 23, 2022. [Online]. Available: https://news.samsung.com/global/samsung-receives-the-industrys-first-environmental-product-declaration-certificate-for-512gb-v-nand-and-860-evo-4tb-ssd

[32] Seagate. (2019). *Nytro 1551 Sustainability Report*. Accessed: Jun. 23, 2022. [Online]. Available: https://www.seagate.com/gb/en/global-citizenship/product-sustainability/nytro-1551-sustainability-report/

[33] Cloud Carbon Footprint. (2022). *Methodology*. Accessed: Jun. 23, 2022. [Online]. Available: https://www.cloudcarbonfootprint.org/docs/methodology/

[34] Amazon AWS. (2020). *AWS Simple Monthly Calculator*. Accessed: Jun. 23, 2022. [Online]. Available: https://calculator.s3.amazonaws.com/index.html

[35] I. Tiseo. (Jun. 2022). *Annual CO2 Emissions Worldwide From 1940 to 2020*. Accessed: Jun. 23, 2022. [Online]. Available: https://www.statista.com/statistics/276629/global-co2-emissions/

[36] R. Bernet. (Oct. 2021). *How Much CO2 Does a Tree Absorb?* Accessed: Jun. 23, 2022. [Online]. Available: https://sdgs.un.org/goals

[37] B. Gallagher. (Oct. 2020). *The Amount of Data in the World Doubles Every Two Years*. Accessed: Jun. 23, 2022. [Online]. Available: https://medium.com/callforcode/the-amount-of-data-in-the-world-doubles-every-two-years-3c0be9263eb1

[38] Statista Research Department. (May 2022). *Volume of Data/Information Created, Captured, Copied, and Consumed Worldwide From 2010 to 2025*. Accessed: Jun. 23, 2022. [Online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created/

BEHRAD BABAEE received the M.Sc. degree (Hons.) in advanced computing and cryptography from the University of Bristol. He has worked for database companies like Aerospike and Datastax. He also has plenty of experience with other scalable solutions for streaming, computing, and caching. He is a Software Engineer at Aerospike with experience in delivering mission-critical solutions to clients in multiple verticals, including financial services, oil and gas, advertisement, media, telecommunications, retail, and tech. His research interests include delivering highly scalable, high throughput, and low latency systems.

DAMON DAYLAMANI-ZAD received the B.Sc. degree in software engineering from the University of Tehran, and the M.Sc. degree in multimedia computing and the Ph.D. degree in electronic and computer engineering from Brunel University London. He is a Senior Lecturer in AI and games with the College of Engineering, Design and Physical Science, Brunel University London, where he is an EPSRC Research Fellow. His research interests include applications of artificial intelligence, machine learning, collaborative games, serious gaming, user modelling and personalization, and application of evolutionary algorithms in creative computing. He is a fellow of the British Computing Society. He has published his research findings widely in journals, edited books and presented his work at several conferences including those hosted by the IEEE.

KEN TUNE received the B.A. degree in maths from Cambridge University and the M.Sc. degree in computer science from Imperial College, London. He spent eight years as a Consultant at MarkLogic, a document database, reaching Senior Principal level, being responsible for guidance and implementation of over 20 separate deployments. He is a Senior Solution Architect with Aerospike. He has released several open source libraries relating to Aerospike. He has deep knowledge of finance, having worked for Hambros Bank, HSBC and Markit Group with experience including risk management, derivatives pricing, and major system integration.