Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Load Balancing and Carbon Emission Control for Sustainable Cloud

[1]Mr. Ramnaresh Lodhi, [2]Dr. Shahnawaz Ansari, [3]Dr. Anil Pimpalapure

[1] Research Scholar, Eklavya University Damoh

[2] Professor, Eklavya University Damoh

[3] Professor, Eklavya University Damoh

(Received: 22 September 2023 Revised: 24 October Accepted: 27 November)

KEYWORDS

Voronoi Partitions, Cloud Computing, Load Balancing, Carbon Emissions.

ABSTRACT:

The system named Stratus, intended for big public cloud infrastructure, is described in the abstract. Several power plants, each with different electrical costs and carbon emissions, power this system. For every request, the system attempts to direct traffic to the data center that is closest in terms of geographic distance, least expensive power, and least amount of carbon emissions. This is accomplished by modeling the networking and computing components as graphs and using Voronoi partitioning to identify the data center for routing according to the priorities set by the cloud operator. The significance of cloud computing services is emphasized in the introduction, along with the difficulties associated with load balancing, latency, operating expenses, and carbon emissions in data centers. The advantages of distributed servers globally in terms of lower latency and operating expenses are mentioned. It also covers the possible long-term effects of carbon emission rules as well as the mounting worry over carbon emissions from data center electricity.

1 INTRODUCTION

To solve these issues, a number of projects and plans have been put up, such as the utilization of locally produced clean energy and load balancing based on electricity pricing and carbon intensity. Nevertheless, network traffic-related carbon emissions are not fully taken into account in the current recommendations. It's also necessary to take into account elements like service level agreements (SLAs), cooling techniques, and the price of electricity.

The study presents Stratus, a graph-based method that uses Voronoi partitions to regulate cloud operation parameters, in order to overcome these issues. The paper offers several contributions: a distributed algorithm for minimizing average request time, electricity cost, and carbon emissions; data on carbon intensity, electricity prices, and round trip times for different geographical regions; and the development of a model that details carbon emissions, electricity costs, and request time for computational and networking aspects. The performance of the distributed method is assessed in the study under different conditions.

The paper's overall goal is to offer a thorough approach to cloud operation optimization that takes into account a variety of variables, including latency, operating costs, and environmental impact.

An overview of current proposals and research initiatives focused on various elements of cloud computing operations optimization, with a particular focus on electricity costs and carbon emissions, can be found in the related work section.

2 LITERATURE REVIEW

Stanojevic et al. [2] provided a distributed consensus technique to decrease energy prices while preserving QoS levels, whereas Qureshi et al. [1] introduced a distance-constrained energy price optimizer. Electricity cost was articulated as a flow network problem by Rao et al. [16, 17], who also suggested control algorithms for load balancing and server power regulation. A revised

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

marginal cost method was presented by Wang et al. [18], while Mathew et al. [19] suggested an algorithm to regulate the number of servers online in order to lower energy usage.

Liu et al.'s [3] distributed algorithms make use of optimization strategies like gradient projection to reduce energy and delay costs. They also took into account the financial and environmental effects of data centers on society.

In an effort to increase power efficiency, Baliga et al. [14] and Mahdevan et al. [20] examined power usage in network switches and cloud computing infrastructure, respectively.

Certain recommendations, such those made by Moghaddam et al. [21], Doyle et al. [13], and Liu et al. [12], took carbon emissions into account when determining how to prioritize service requests. A methodology for flow optimization was created by Gao et al. [22] to balance carbon emissions, electricity costs, and average job time.

The paper suggests using Voronoi partitions, which have been used in a variety of applications, as an alternative to current methods. Voronoi partitions provide a less complicated strategy with potential advantages for dynamic demand scenarios in cloud computing, in contrast to earlier approaches that frequently entail sophisticated mathematical procedures. to explain the meanings of a graph and a Voronoi division and the ways in which cloud computing uses these ideas.

A set of vertices (V) and a set of edges (E) that each connect two vertices make up a graph (G = (V, E)). Vertices can stand in for data centers in the context of cloud computing, while edges can stand in for the networks connecting them. Weights indicating variables like latency, electricity costs, or carbon emissions related to data transmission between data centers may be assigned to edges.

A Voronoi partition is a partitioning of a space into regions according to how close a set of points known as seeds are to each other. Every area has every point that is closest to a specific seed relative to every other seed. Voronoi partitions can be used in the context of cloud computing to divide geographical areas into areas that are serviced by various data centers. The Voronoi partition establishes which geographic regions are serviced by each data center, which functions as a seed.

3 PROBLEM FORMULATION

The formulation of the problem is figuring out how to distribute incoming service requests around data centers to minimize a number of variables, including average request time, electricity cost, and carbon emissions. This can be accomplished by taking into account the client's location while submitting the request, each data center's cost of electricity and carbon emissions, and the network delay between the client and each data center. By taking these parameters into consideration, Voronoi partitions can be utilized to effectively determine which data center is nearest to each client, guiding the assignment of service requests accordingly.

The following definition of a graph's fundamental elements and ideas is accurate:

1. **Connectors:** Nodes, sometimes called vertices, are the constituent components that make up a graph. Nodes can stand in for a variety of things in the context of cloud computing or network architecture, including servers, data centers, and physical locations.
2. **Rounded Corners:** The links between nodes are shown by edges. They show connections or exchanges between the entities that the nodes represent. An edge in an undirected graph is bidirectional since it lacks a specified direction associated with it. In contrast, edges in directed graphs, also known as digraphs, have a direction that denotes a one-way interaction between nodes.
3. **Route:** In a graph, a path is a series of nodes joined by edges between every subsequent pair of nodes. In a graph, paths are used to represent pathways or links between various nodes.
4. **Graph with Weights:** Every edge in a weighted graph has a weight, which is a numerical number ascribed to it. These weights can stand for a number of characteristics related to the node-to-node connection, including latency, cost, distance, and carbon emissions.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

The definition given clearly describes the basic features and attributes of graphs, which are necessary for modeling and assessing a variety of systems, such as network architecture and cloud computing.

Voronoi partitions and their use are defined precisely in the following description:

1. **Voronoi Partitions:** A geometric breakdown of a space into regions based on a set of points called sites, generators, or seeds is known as a Voronoi partition, sometimes referred to as a Voronoi diagram. All of the locations in the space that are closer to one of these sites than to any other site make up each area in the partition, which is centered around one of these points. Put otherwise, every point in the space corresponds to the area that is closest to the site.
2. **Distance Metric:** A distance metric is used to determine which zone a location is located in. The concept of distance between a point and the sites is defined by this measure. Depending on the particular application, common distance metrics include Manhattan distance, Euclidean distance, and other customized metrics.
3. **How to Use:** Applications for voronoi partitions can be found in many domains, such as robotics, where they are employed to split a workspace into areas that are designated for distinct robots. Every robot is in charge of sweeping the area connected to its site or seed point. This method facilitates effective workspace coverage and multi-robot collaboration.

Voronoi partitions can be used to partition the geographical space into regions that correspond to various data centers or server locations in the context of cloud load balancing. Client queries are directed to the data center linked to the region in which the client is located, as each region revolves around a single data center. By allocating the workload among various data centers according to how close they are to clients, this strategy helps to maximize resource usage and lower latency.

In our work, the sources of cloud service requests as well as data center locations make up the set of points. The paths to various data centers are linked to each request source, and these paths are used to determine the Voronoi partitions. The partitions designate which data center is in charge of handling requests coming from particular sources.

An illustration of how request sources are divided between two data centers is shown in Figure 1. The data center with the shortest path to each source of requests is assigned to it. These divisions are represented by the Voronoi cells, which show which sources of requests each data center is in charge of fulfilling at any particular time.

All things considered, Voronoi partitions are employed to effectively assign incoming requests to the relevant data center according to their network pathways and proximity. This methodology facilitates the optimization of workload allocation among data centers and enhances the overall efficiency of the cloud infrastructure

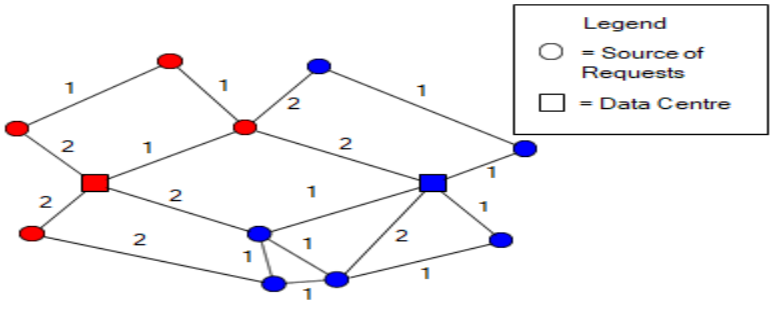

Fig. 1. Example of how sources of requests are partitioned between two data centres. Colour indicates that the node is part of a particular partition.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

A graph (G = (|Q|, |E|, |w|)) is defined by the formulation, in which (|Q|) is a set that represents the sources of requests or data centers, (|E|) is the set of edges linking these points, and (|w|) is the set of weights attached to the edges. The time (T_i) needed to service a portion of the request, the carbon emissions (G_i) related to servicing the portion, and any associated power costs (E_i) along the edge are used to calculate the weights. To determine these weights' relative relevance, the functions (R_1) and(R_2) are used.

The regions that each data center serves are represented by(N) subsets of the set (|Q|). A collection of (N) subsets of (|Q|) is produced by this partitioning, and each subset (P = {P_i}_{i=1}^N satisfies the following criteria:

1. All subsets added together equal (|Q| .
2. If (i neq j), then the intersection of any two subsets (P_i) and (P_j) is empty.
3. There are no empty subsets in any of P_i.
4. All of P_i's subsets are connected.

Next, a set of subsets is created using the Voronoi partitioning in order to minimize the sum of the three variables: average request time, power cost, and carbon emissions. The definition of the Voronoi partition (P_i) connected to data center (i) is as follows:

1. Define (U := P_i(t) cup P_j(t)), where the current partitions for data centers (i) and (j) are denoted by(P_i(t)) and (P_j(t)) correspondingly.
2. Based on the distance to the data centers (i) and (j), for each point (x) in (U), identify if it is a part of(P_i) or (P_j).
3. Using the distances computed in step 2, update the partitions (P_i(t + 1)) and (P_j(t + 1)) appropriately.

Figure 2 provides the pseudocode for this pairwise partitioning rule. This method takes into account the various aspects influencing the delivery of cloud services and guarantees the effective use of resources.

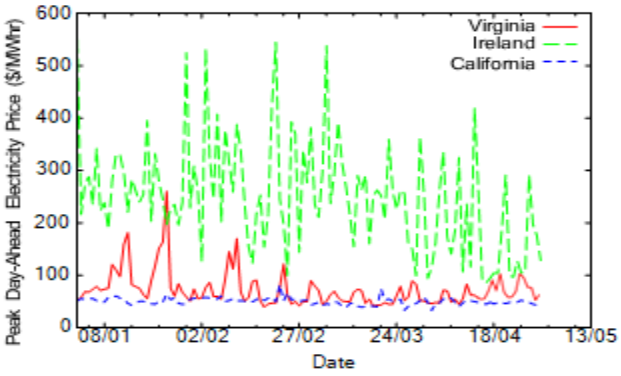

Fig 2. Peak Daily Price of electricity for supplier in the region of three centres studied

A pairwise partitioning rule can be used to find the Voronoi partitions that minimize the distance between request sources and data centers. The objective of this rule is to allocate every point to the Voronoi division linked to the closest data center. To balance the load on the data centers, a source that is equally spaced from many data centers is assigned to the partition with the fewest members.

The lowest weight of a path, (d(i, j)), connecting two points (i) and (j) in a weighted graph represents the distance between them. The total weight of all the path's edges equals the weight of the path.

The following is an expression of the goal function that seeks to reduce the distance between request sources and data centers:

[min sum_{i=1}^{N} sum_{j in P_i} d(i, j)}]

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

where: - (N) is the total number of data centers; - (P_i) is the Voronoi partition associated with data center (i); - (d(i, j)) is the distance, computed as the lowest weight of a path between point (i) and (j).

The Voronoi partitions can efficiently distribute request sources to the closest data centers, optimizing the distance between them, by minimizing this objective function. This guarantees effective load balancing and resource distribution among the cloud infrastructure's data centers.

The distance between every graph node and every data center is computed in order to create the first Voronoi partitions. The partition linked to the data center that produces the shortest path between the node and the data center is then expanded to include each node.

The following pseudocode can be used to summarize the procedure:

1.Initialize Partitions: Using computed distances, assign each node to the partition linked to the closest data center.

2. Partitions should be updated:

- For any set of partitions connected to data centers (i) and (j), as follows:
- For every graph region:
- Include the region in a temporary partition linked to data center (i) if the path between the region and data center (i) is shorter than the path between the region and data center (j).
- If not, include the area in a temporary division linked to data center (j).
- Apply the necessary temporary partitions to the two areas' partitions.

By ensuring that every node is assigned to the Voronoi partition linked to the closest data center, this procedure optimizes resource allocation and load balancing within the cloud infrastructure.

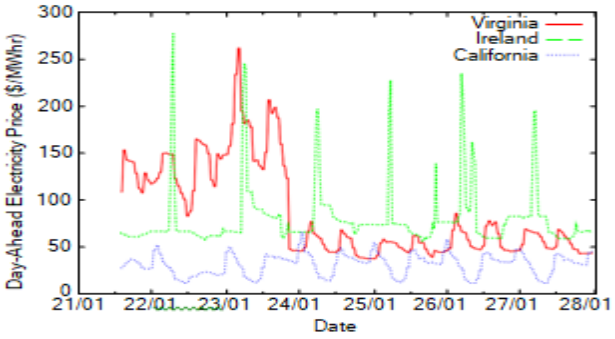

Fig. 3 Peak Daily Price of electricity for supplier in the region of three centres studied

We must take into account a number of variables, including average request time, carbon emissions, and electricity costs, in order to examine the differences in prices throughout data centers. These variables may change based on the data center's location and other outside variables. We can carry out the analysis as follows:

1. Data Collection: Compile information on typical request times, carbon emissions, and electricity costs for every data center. Performance monitoring systems, power bills, carbon emission reports, and historical records are some possible sources of this information.
2. Cost Calculation: Add the expenses for power, carbon emissions, and average request time to determine the overall cost for every data center. This could entail giving each factor a weight according to its relative relevance.
3. Comparison : To find differences, compare the overall expenses of several data centers. Tables or graphs can be used to illustrate the differences in this comparison.
4. Analysis: Examine the elements influencing the price fluctuations. Costs can be affected by a variety of factors, including demand patterns, infrastructural efficiency, energy sources, and geographic location.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

5. Optimization: Look at possible cost-saving measures like splitting up the workload among data centers, implementing energy-saving technology, or settling on more favorable electricity contracts.
6. Sensitivity Analysis: Perform sensitivity analysis to determine how changes in input parameters (such as the price of power or the factors affecting carbon emissions) will affect the total expenses. This can assist in determining the main causes of cost variances.

We can learn more about the differences in prices throughout data centers and find areas where cloud infrastructure management may be optimized and expenses can be reduced by carefully examining these variables.

We must take into account the local electricity provider pricing in the areas where the data centers are situated in order to examine the variations in electricity costs between data centers. Here, we'll concentrate on the states of California, Virginia, and Ireland and investigate the possible financial benefits of taking advantage of the variations in electricity rates.

1. Data Collection: Get information from regional providers in the states of Virginia, California, and Ireland regarding the cost of power. This information could be publicly accessible market data or historical electricity price information from the relevant suppliers.
2. Price Comparison: Examine the costs that various providers in each area are charging for electricity. Examine how costs have changed over time and note any notable regional variations.
3. Cost Savings Calculation: Determine how much you might save by using data centers located in areas with cheaper electricity rates. To find the overall cost of power, this computation may entail estimating each data center's electricity consumption and multiplying it by the associated electricity rates.
4. Scenario Analysis: Examine several situations to determine how fluctuating electricity prices affect overall operating expenses. Take into account variables that could impact electricity pricing, such as times of peak demand, seasonal fluctuations, and regulatory changes.
5. Strategic Decision-Making: Determine how to divide the workload among data centers strategically using the analysis as a guide. Optimizing resource use means taking into account things like dependability, performance needs, and cost savings.
6. Risk Assessment: Consider the hazards involved with using power pricing as a means of cutting expenses. Future electricity costs may be impacted by variables like supply outages, market volatility, and regulatory uncertainty.

Cloud operators may maximize their data center infrastructure's efficiency and reduce operating costs by undertaking a thorough review of electricity pricing in various regions.

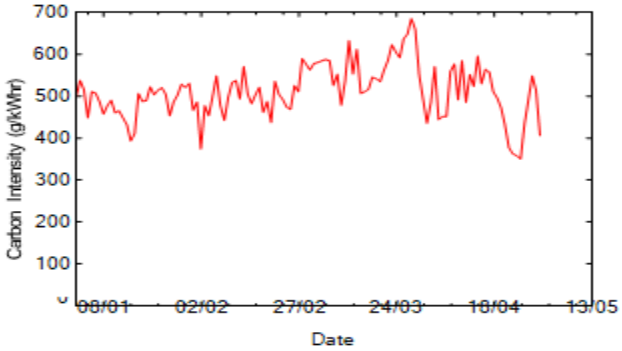

Fig. 4 Daily peak carbon intensity of electricity supplier in the region of india data centre studied

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

An examination of the financial and environmental effects of cloud infrastructure can be gained by comparing electricity prices and carbon emissions across various areas. Below is a summary of the main conclusions:

1. Electricity costs: - The Ireland region usually has the highest peak costs, with a maximum price per MWh approaching $550. The hourly variance of power costs shows that, despite Ireland's high peak prices, there are significant price changes during peak hours and less during off-peak hours.
2. Carbon Emissions: Reducing the environmental impact of cloud operations requires examining the carbon intensity of electricity providers in different areas.
   - Data on carbon intensity fluctuates over time as a result of shifts in power plant operations and electricity consumption. Environmental effect can be reduced by using real-time data.
   - Our method makes load routing decisions based on carbon intensity data instead of weather data. This is beneficial because, because of the intricate dynamics of power plant operations and the dynamics of the electricity market, weather conditions do not always directly correlate with carbon emissions.
3. Difficulties and Points to Remember: Supply and demand in the electrical grid are difficult to balance, particularly when using uncertain renewable energy sources like solar and wind power.
   - Even with the use of renewable energy sources, carbon intensity can fluctuate due to changes in power plant operations and energy demand.
   - The generation of renewable energy may not always directly correlate with carbon intensity, which emphasizes the necessity for complex algorithms to efficiently optimize load balancing and reduce environmental impact.

To summarise, cloud operators may minimise their environmental impact and optimise costs by utilising real-time data on carbon emissions and electricity prices to influence their decision-making. However, while creating efficient optimization strategies, factors like the dynamic nature of power networks and the integration of renewable energy must be carefully taken into account.

Data centers employ the confinement of aisles as a tactic to divide hot and cold airflow routes, increasing cooling effectiveness and lowering cooling expenses. This is an explanation of aisle containment's operation and how it affects cooling expenses:

1. **Aisle Containment:** Using physical barriers like Plexiglas or PVC drapes, the hot aisle (where server exhaust airflow departs) and the cold aisle (where cool air is fed to the servers) are enclosed.
   - Aisle containment increases the effectiveness of cooling systems by separating the hot and cold airflow channels. This keeps hot and cold air from mingling.
   - In order to ensure that servers receive cool air directly and to reduce hot air recirculation, cold air provided through perforated floor tiles is directed particularly to the cold aisle.
2. **Impact on Cooling Costs: -** By maximizing airflow and temperature dispersion, aisle confinement lessens the strain on cooling equipment like computer room air conditioners (CRAC) units or chiller units. Aisle confinement makes cooling systems work more effectively by keeping hot and cold air from mingling, which lessens the need for excessive cooling and cuts down on energy use. For data center operators, this means fewer energy bills and operating costs due to the decreased cooling workload.
3. **Difference in Cooling Expenses: -** Aisle containment's ability to lower cooling costs may vary according on workload, local climate, and data center design, among other things.
   - Because natural cooling techniques like free air cooling can be used, aisle containment may have a greater effect on cooling costs in areas with warmer climates or lower ambient temperatures. On the other hand, aisle containment may still be beneficial in areas with warmer temperatures overall or in warmer climes, but it may also be

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

enhanced by other cooling techniques like water cooling or additional cooling systems.

All things considered, aisle confinement works well to maximize cooling effectiveness and lower cooling expenses in data centers. Data center operators can achieve considerable energy savings and operating expense reductions while maintaining optimal server performance and reliability by employing aisle containment and taking other cooling measures into consideration based on local climate conditions and workload.

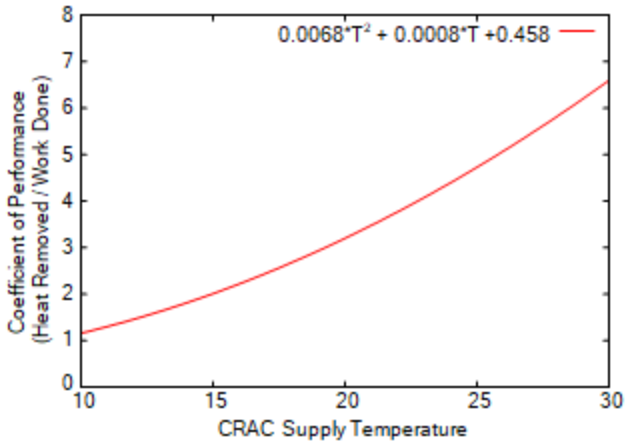

Fig. 5 Typical model of the Coefficient of Performance (COP) curve for chilled water CRAC Unit

Let's examine how the cooling expenses for the data center model were calculated and interpreted:

1. **Power Consumption of Servers: -** The HP Proliant DL360 G3s model uses 150W of power when it is not in use and 285W when it is. The data center uses 319.2 kW of power when it is fully utilized (285W/server * number of servers), and 168 kW when it is idle (150W/server * number of servers).
2. **Cooling Equipment: -** There are four CRAC units in the data center; they each have a 90 kW cooling capacity and a 10 kW power consumption. At a rate of 16,990 m^3, each CRAC unit forces chilled air into the plenum.
3. **Calculating the Cost of Cooling:** The following formula can be used to determine cooling costs (C): [C =frac{Q}{COP} (T_{text{sup}} + (T_{text{safe}} - T_{text{max}})) + P_{text{fan}}] where: - (Q) is the power consumed by the servers; - (T_{text{sup}}) is the temperature of the air supplied by CRAC units; - (T_{text{safe}}) is the maximum temperature that can be reached at the server inlets; - (T_{text{max}}) is the maximum temperature of the server inlets; - (P_{text{fan}}) is the power needed by the CRAC unit fans; and - (COP) is the CRAC unit's coefficient of performance. - The temperature of the air that the CRAC unit supplies affects the COP.
4. **Results and Interpretation: -** The cooling cost in kilowatts for the three systems—CAC-CRAC (cold aisle containment with CRAC cooling), FAC (free air cooling), and NCAC-CRAC (no cold aisle containment with CRAC cooling)—is displayed in the simulation results (Figure 9).

- The cooling cost in kilowatts is represented on the y-axis, and the x-axis shows the data center's % utilization.
- On the graph, every line corresponds to a distinct cooling system setup.
- The simulation makes it possible to compare cooling costs under various conditions, which helps choose the data center's most economical cooling plan.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

In summary, the cooling cost simulations offer significant insights into the energy consumption and financial consequences of various cooling approaches, assisting data center managers in making well-informed decisions to maximize energy efficiency and minimize operating costs.

.

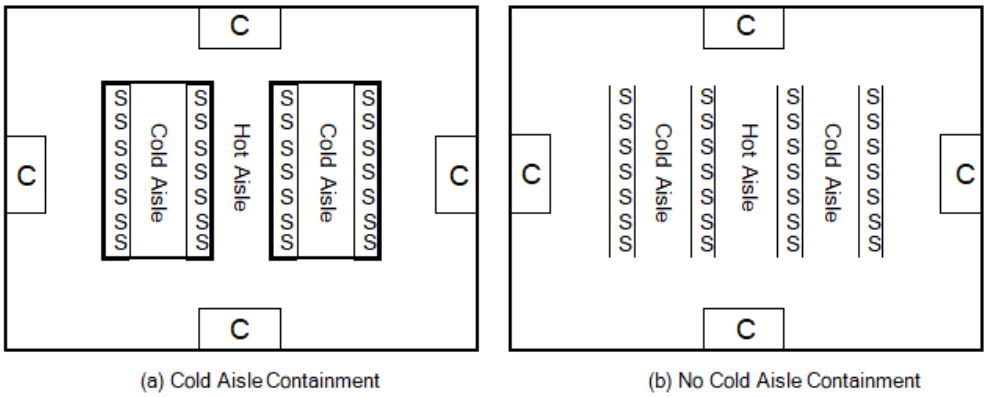

Fig. 6 Layout of cooling cost simulators with (a) cold aiste containment and (b) no cold aisle containment

Table 1

Average round trip time between data centres and sources of requests, carbon intensity of data centres and sources of requests and daily number of requests at source

Region	California (ms)	Ireland (ms)	Virginia (ms)	Carbon Intensity (g/kWhr)	Number of Requests (Millions)
Austria (AUS)	177.98	47.67	159.07	870	7.038
Belgium (BEL)	171.98	28.45	158.09	317	11.736
California (CAL)				384	
Colorado (COL)	42.77	155.36	101.27	903	6.76
Connecticut (CON)	88.05	117.65	75.73	392	4.293
Finland (FIN)	188.47	55.77	176.72	99	5.418
Florida (FLO)	54.26	171.87	98.21	762	26.365
France (FRA)	192.44	21.24	184.75	96	61.355
Georgia (GEO)	58.91	115.12	77.43	694	1.968
Germany (GER)	177.74	40.89	157.68	612	58.76

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Illinois (ILL)	63.81	142.78	102.08	544	18.049
Indiana (IND)	69.65	151.05	83	986	7.803
Ireland (IRE)				655	
Italy (ITA)	188.71	44.71	167.3	473	55.372
Kansas (KAN)	50.48	148.4	85.2	817	4.545
Kentucky (KEN)	71.98	146.85	87.29	968	5.169
Maryland (MAR)	99.71	140.46	88.05	641	6.69
Massachusetts (MAS)	89.33	98.02	72.1	603	9.602
Minnesota (MIN)	62.09	147.74	85.79	744	6.724
Netherlands (NET)	163.93	19.71	138.79	548	15.527
New York (NEW)	96.45	78.71	134.11	386	27.604
North Carolina (NCA)	72.32	72.45	31.12	604	11.817
Norway (NOR)	194.82	48.84	183.08	6	6.69
Ohio (OHI)	83.81	132.08	69.17	873	14.828
Oklahoma (OKL)	46.42	159.96	98.22	819	4.378
Ontario (ONT)	90.84	142.81	97.12	224	16.64
Oregon (ORE)	27.66	213.48	153.28	246	4.807
Pennsylvania (PEN)	71.99	118.53	52.78	597	16.097
Portugal (POR)	222.69	64.11	190.02	550	10.925
Spain (SPA)	194.55	35.83	172.47	487	40.633
Sweden (SWE)	186.01	48.79	170.28	19	11.887
Tennessee (TEN)	235.61	276.43	311.61	661	7.891
Texas (TEX)	37.61	151.93	98.96	763	31.015
UK (UK)	175.59	17.62	163.25	614	78.647

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Virginia (VIR)				559	
Washington (WAS)	29.57	192.11	126.53	938	10.119
Wisconsin (WIS)	67	146.49	94.03	834	7.025

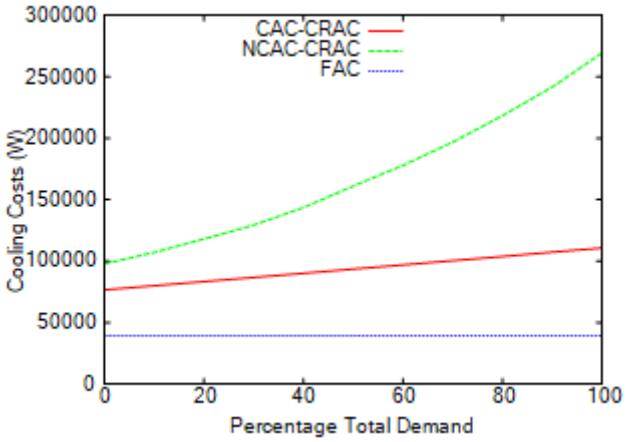

Fig. 7 Cooling cost of various data centre cooling system at various level of demand

If so, let's modify the interpretation as appropriate:-

Cooling Cost Calculation: -

The cooling cost for the system that uses CRAC cooling and cold aisle containment is computed using the previously given formula.

- On the other hand, because the "free air cooling" system uses ambient air from the surrounding environment and doesn't require additional mechanical cooling, the cooling cost simply comprises the power needed for the fans.

4 RESULTS AND INTERPRETATION: -

The cooling cost in kilowatts for the three systems—CAC-CRAC (cold aisle containment with CRAC cooling), FAC (free air cooling), and NCAC-CRAC (no cold aisle containment with CRAC cooling)—is displayed in the simulation results (Figure9).

- The "free air cooling" system's cooling cost is constant at all data center usage levels because it just uses fan power.
- On the other hand, because of the CRAC units' operation, the CAC-CRAC system has variable cooling costs based on the data center's use.
- In particular, in areas with favorable ambient conditions, the comparison of these systems sheds light on the potential cost and energy savings associated with switching to "free air cooling" from standard CRAC-based cooling solutions.

Overall, the simulation findings demonstrate the potential benefits of "free air cooling" over traditional CRAC-based cooling systems in terms of cost and energy efficiency, especially in areas with climates that allow ambient air to be used for cooling. policy balancing. The latency between the server and client is a useful statistic for service request time. A server was installed at each node location as part of an experiment on PlanetLab [39] to obtain the round trip time statistics. Then, for around two days, nodes in the same region as our three data center locations sent out

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

fifteen-minute interval pings to the other geographical regions. Table 1 displays the average latency determined by this experiment.

By adopting the lowest latency as a criterion for routing load, the average service request time might be decreased by routing load from a geographical area to the data center region. Table 1 illustrates that in the event that a load balancing scheme is implemented, a portion of the load will be sent towards each data center region. This means that since load is not routed using latency as the only criterion, any decrease in carbon emissions or electricity costs will result in an increase in average latency.

The measured latency for the Virginia, California, and Ireland data centers, respectively, between the data center and the other areas is shown in Figures 10, 11, and 12. Interestingly, latencies at the Virginia data center fluctuate a lot while they stay relatively consistent in the California and Ireland data centers over time. We speculate that the Virginia region's congestion is to blame for this. Furthermore, we observe that the latency between the California and Ireland data centers and the other areas varies to some extent. Consequently, we can say that latency changes over time, especially in areas where congestion occurs.

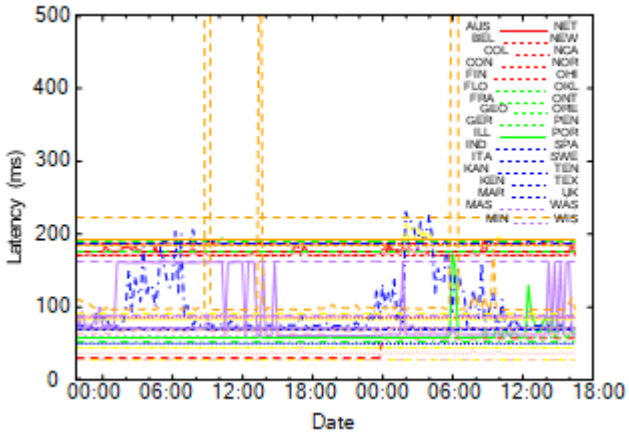

Fig. 8 Latency between US and different geographical region at 15 Min interval

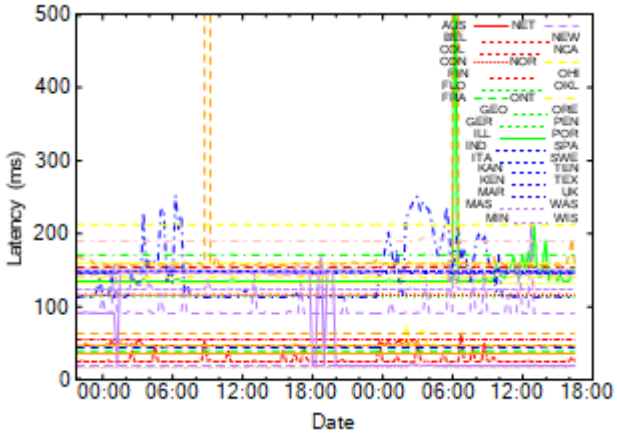

Fig. 9 Latency between India and different geographical region at 15 Min interval

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Without a doubt, keeping an eye on the rise in average service request time brought on by the decrease in carbon emissions or electricity costs is essential to precisely determining the effects of such optimizations. It's critical to understand that, while this may not always be the case, there are situations in which cutting electricity costs or carbon emissions doesn't result in longer average service request times. Changes in server loads, network congestion, and other variables might affect latency and overall service quality.

Through consistent observation and evaluation of these indicators, cloud providers can make knowledgeable choices regarding infrastructure optimization while upholding satisfactory standards of service quality. Furthermore, using dynamic load balancing algorithms that take into account a variety of variables, such as latency, carbon emissions, and electricity costs, can assist in achieving a balance between service efficiency and environmental sustainability.

Three data centers were established: one in Ireland and two in the United States (Virginia and California) to run the algorithm and determine the weights of the graph. These sites were picked to reflect the key data centers that Amazon's EC2 platform is located in these areas.

We model 34 sources of requests in our simulation, representing states in the US, provinces in Canada, and several European countries. As shown in Figure 13, every source is linked to every data center by a single edge.

We must compute the time, carbon emissions, and electricity cost involved with fulfilling the networking and computational components of a request in order to establish the weights of the edges in the graph. This includes:

1. **Time:** Determining the amount of time needed to fulfill a request while accounting for computational processing time and network latency.
2. **Carbon emissions:** Calculating the carbon emissions related to fulfilling the request while accounting for the electricity's carbon intensity and any other pertinent variables.
3. **Electricity cost:** Calculating the cost of electricity required to fulfill the request while taking into account variables like regional electricity rates and data center power usage.

After identifying these variables, we may use functions that aggregate these metrics to give weights to the graph's edges, as explained in Section 3.4. The algorithm will then use these weights to decide how to route requests within the cloud infrastructure.

Fig. 10 Simulation Setup

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

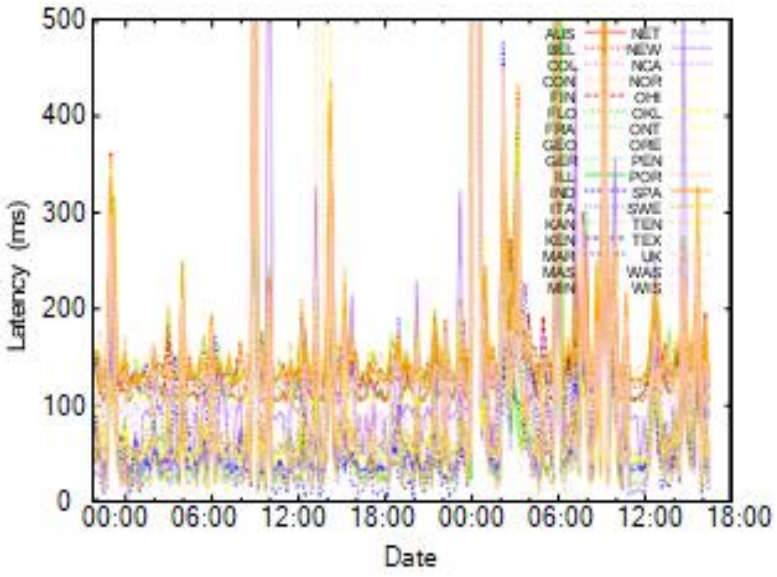

Fig. 11 Latency between Virginia and different geographical region at 15 Min interval

On the other hand, under the best effort electricity scenario, the algorithm gives priority to lowering electricity costs over carbon emissions, resulting in an equal distribution of requests throughout the data centers.

We modify the relative importance of carbon emissions and power cost by varying the parameters α and β in the second set of simulations, where the algorithm balances the elements of time, carbon emissions, and electricity cost. Table 3 provides a summary of these simulations' outcomes. These findings shed light on the trade-offs that arise when attempting to balance several goals in cloud resource management.

All things considered, the simulations show how well the suggested algorithm works to optimize the distribution of cloud resources according to a number of parameters, including time, carbon emissions, and electricity costs. Cloud operators can make well-informed judgments to optimize their operations while taking cost and environmental effect into account by looking at the performance under various scenarios.

To investigate more trade-offs and enhance the algorithm's performance in actual cloud systems, more research and optimization may be necessary. Furthermore, continuous observation and adjustment in response to evolving circumstances and demands will be necessary to guarantee the optimization and sustainability of cloud resource management moving forward.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Table 2

Average Service Request time, Daily Carbon Emission and Number of Requests Serviced at Each DC for Various Scenarios

Scenario	Average Service Request Time (ms)	Carbon Emis-sions (kg)	Electricity Cost ($)	Number of Requests Serviced by California (million)	Number of Requests Serviced By Ireland (million)	Number of Requests Serviced By Virginia (million)
Best Effort Time	90	1378	240	241	822	302
Best Effort Carbon	132	1200	195	0	1365	0
Best Effort Electricity	177	1567	100	1276	0	89
RoundRobin	170	1522	257	455	455	455

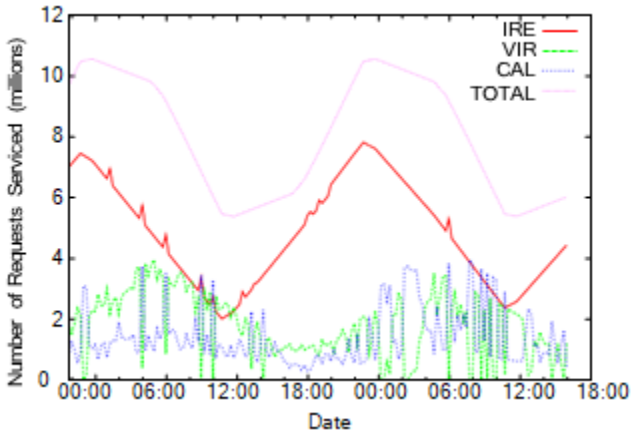

Fig. 12 Number of request serviced at each data centre when best effort time scenario is used

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

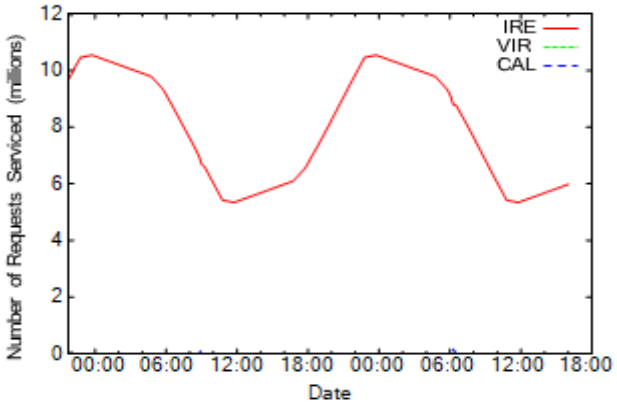

Fig. 13 Number of request serviced at each data centre when best effort carbon scenario is used

This graphic shows how requests are split up among the data centers dynamically over time for various scenarios. It makes it possible to see patterns and trends in the distribution of requests, demonstrating how well the algorithm adjusts to shifting priorities and situations.

Furthermore, cloud operators can learn more about the effectiveness and efficiency of their resource allocation policies by examining the distribution of requests over time. To maximize efficiency and resource use, they can recognize times of high demand or congestion and modify their allocation methods accordingly.

In general, the integration of the quantitative analysis presented in Table 2 with the visual aid of Figure 14 provides a thorough comprehension of the ways in which various scenarios affect the distribution of requests among data centers, as well as the related expenses and service durations. By adopting a holistic viewpoint, cloud operators can balance multiple elements like cost, environmental effect, and service quality while making well-informed decisions to accomplish their goals.

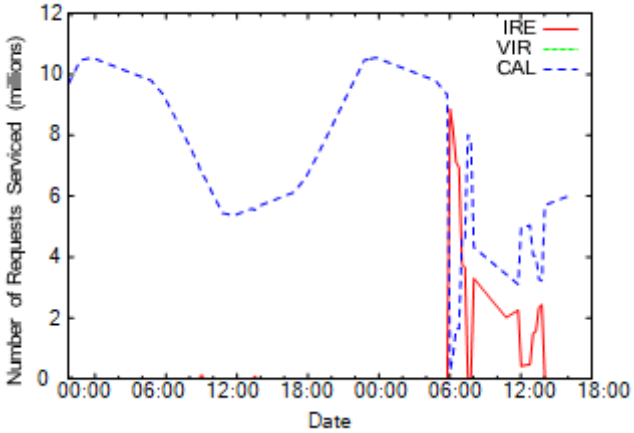

Fig. 14 Number of request serviced at each data centre when best effort Electricity scenario is used

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Figure 17 shows how many requests were handled at each data center over a period of time for the round robin scenario. According to the round-robin load balancing approach, Figure 17 shows that the requests are evenly split among the three data centers during the course of the observation period.

These visuals offer insightful information on how various events affect the service requests distribution across data centers over time. They show how variances in parameters like latency, power expenses, and carbon emissions affect how requests are distributed, resulting in various patterns of use for every data center. Cloud operators can improve their understanding of the load balancing algorithms' performance and make well-informed decisions to maximize resource utilization, save expenses, and improve service quality by examining these patterns.

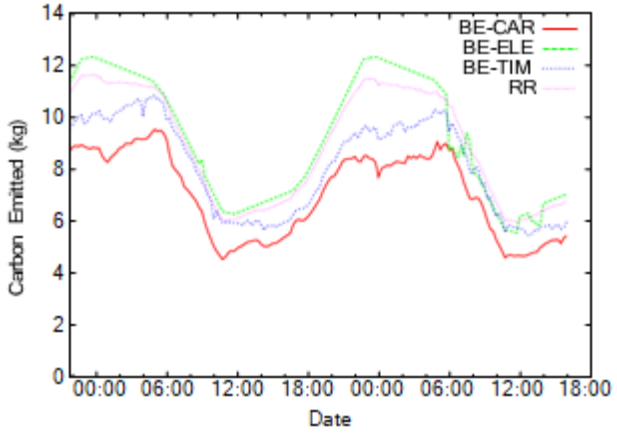

Fig. 15 Carbon Emitted at each time interval under a variety of scene

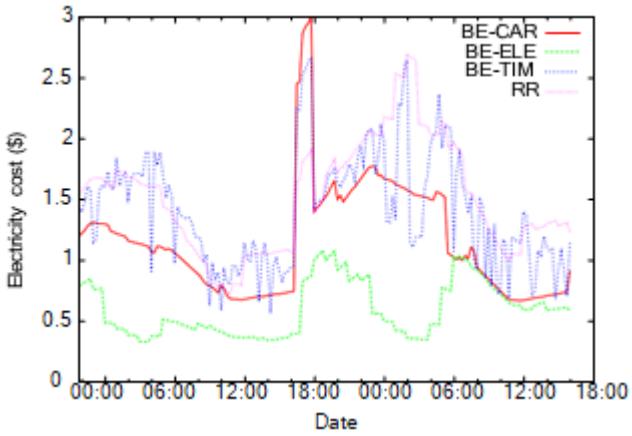

Fig. 16 Electricity Cost at each time Interval

observe that the cost of electricity fluctuates greatly based on the circumstances. Compared to the other scenarios, the electricity cost is comparatively high in the "Best Effort Time" scenario, where the objective is to minimize the time taken for service requests. This is due to the fact that requests are sent to data centers without taking

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

electricity costs into account, only considering delay. On the other hand, since requests are sent to the data center with the lowest carbon emissions—in this case, the Ireland data center—the "Best Effort Carbon" scenario yields the lowest electricity cost. In comparison to the other scenarios, the "Best Effort Electricity" scenario results in intermediate electricity costs since it strikes a balance between decreasing electricity costs and service request times. In conclusion, the "RoundRobin" scenario leads to greater electricity costs than best effort scenarios but lower costs than "Best Effort Time" scenario since it fairly distributes requests around data centers.

These findings demonstrate the trade-offs associated with maximizing load balancing tactics in cloud computing settings. In order to manage their resources more sustainably and efficiently, cloud operators might take into account variables like latency, electricity costs, and carbon emissions.

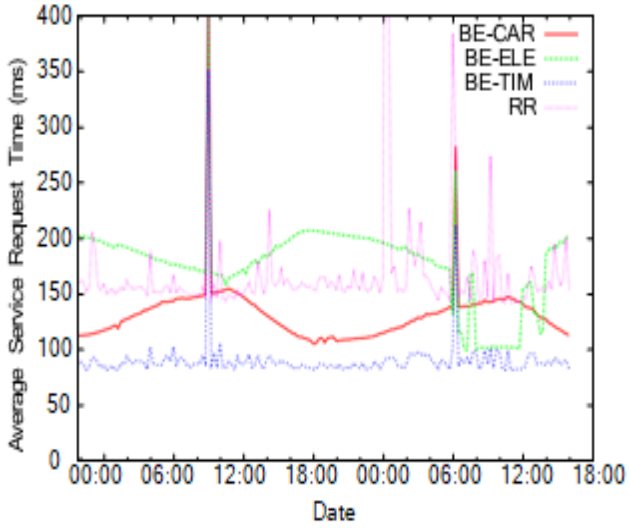

Fig. 17 Average Service request time at each time interval under a variety of Scenarios

The "Best Effort Time" scenario, on the other hand, has the most variability along with the lowest average service request time. This is because it places a higher priority on lowering service request time. This is due to the fact that requests are only sent in response to latency concerns, which causes response times to vary depending on the level of congestion at various data centers.

All things considered, these findings highlight how crucial it is to take into account a variety of elements when developing load balancing algorithms for cloud computing environments, such as latency, electricity costs, and carbon emissions. Cloud operators can maximize resource usage while guaranteeing the effective and long-lasting operation of their infrastructure by finding a balance between these factors. Further studies could examine more intricate request processing models, such the Partition/Aggregate design pattern, to enhance load balancing techniques and boost system performance in general.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

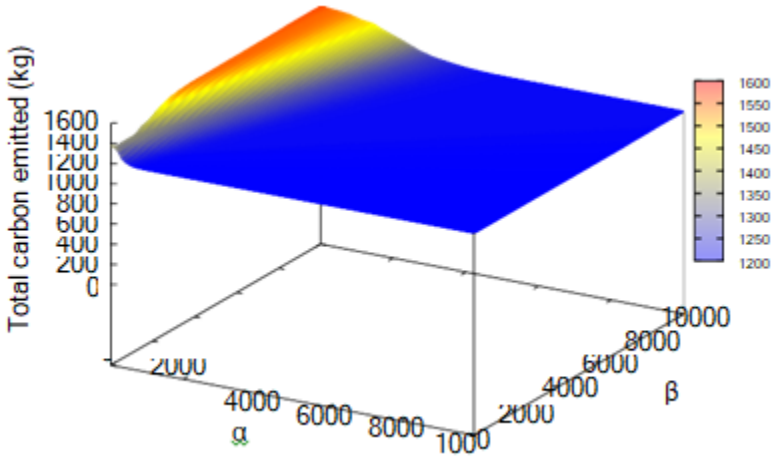

Fig. 18 Total Carbon Output with verying relative price function

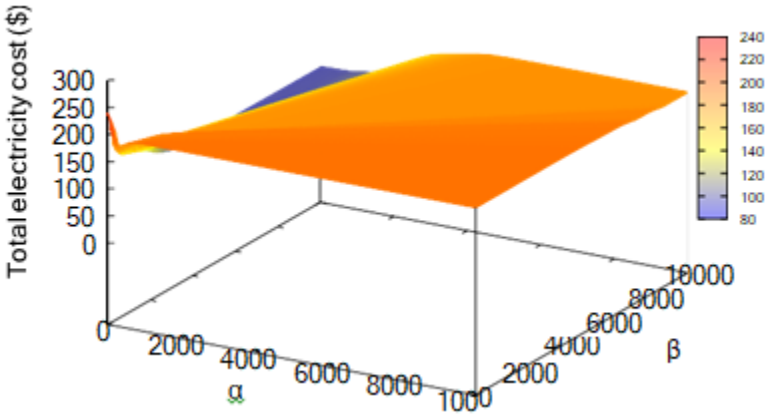

Fig. 19 Total Electricity Cost with Varing Relative Price Function

Apple begins to shift queries from Dublin to California, resulting in an increase in latency and electricity costs. These findings highlight the compromises made in order to balance average service request time, electricity cost, and carbon emissions. Through the manipulation of α and β, cloud operators can customize their load balancing tactics to achieve particular sustainability and performance goals.

In conclusion, the simulations offer insightful information on the relationships between many elements influencing the sustainability and performance of cloud services. Through meticulous evaluation of these variables and the implementation of suitable load balancing algorithms, cloud operators may maximize resource efficiency, reduce ecological footprints, and guarantee compliance with service level agreements.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

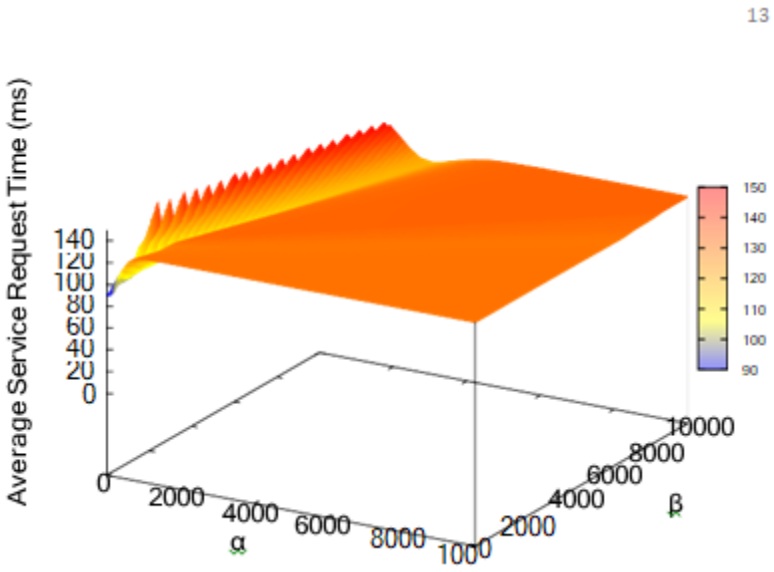

Fig. 20 Average Service Request Time with Relative Price Function

Therefore, choosing the ideal values of α and β requires thorough analysis of the trade-offs between environmental goals and performance indicators. Furthermore, as you pointed out, the load balancing algorithm's adaptability and efficacy might be further increased by dynamically adjusting α and β in response to real-time conditions.

All things considered, the simulations offer insightful information about the intricate relationships between different elements in cloud computing infrastructures. Cloud operators can create more sustainable and effective strategies for resource management and operational goal achievement by integrating these insights into their decision-making processes.
Our research concludes by showing that cloud operations can be adjusted to lower operating expenses and carbon emissions. But the average service request time goes up as a result of this optimization. We have examined the trade-offs between service performance, carbon emissions, and electricity costs under various scenarios through thorough simulations.

In the end, a number of factors, including service level agreements (SLAs), legal constraints, and the financial implications of carbon trading programs, will determine how these aspects are balanced. Operators may design their cloud services in the most advantageous way by using the findings from our study and taking into account the unique features of their cloud infrastructure.

It is significant to remember that the nature of the service and the limitations imposed by SLAs may have an impact on how feasible it is to execute such optimization techniques. However, our research offers helpful advice to cloud operators that want to improve the efficiency and sustainability of their business in a quickly changing technological environment.

REFERENCES

[1] V. Mathew, R. K. Sitaraman, and P. Shenoy, “Energy-aware load balancing in content delivery networks,” in Proceedings of IEEE INFOCOM, Orlando, 25 - 30 March 2012, pp. 954–962.

[2] P. Mahadevan, S. Banerjee, and P. Sharma, “Energy propor- tionality of an enterprise network,” in Proceedings of ACM GreenNet, New Delhi, 30 August 2010, pp. 53–60.

[3] F. F. Moghaddam, M. Cheriet, and K. K. Nguyen, “Low Carbon Virtual Private Clouds,” in Proceedings of IEEE International Conference on

Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(6), 3691-3711 | ISSN:2251-6727

Cloud Computing, Washington DC, 4 - 9 July 2011, pp. 259–266.

[4] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in Proceedings of SIGCOMM, Helsinki, 13 - 17 August 2012, pp. 221–222.

[5] F. Aurenhammer, "Voronoi Diagrams-a survey of a fundamen- tal geometric data structure," ACM Computing Surveys, vol. 23, no. 3, pp. 345–405, September 1991.

[6] R. J. Fowler, M. S. PAterson, and S. L. Tanimoto, "Optimal packing and covering in the plane are NP-complete," Informa- tion processing letters, vol. 12, pp. 133–137, 1981.

[7] R. Z. Hwang, R. C. T. Lee, and R. C. Chang, "The slab dividing approach to solve the Euclidean p-center problem," Algorithmica.

[8] T. F. Gonzalez, "Clustering to minimize the maxium interclus- ter distance," Theoretical Computer Science, vol. 38, pp. 293–306, 1985.

[9] Amazon, "Elastic Compute Cloud," http://aws.amazon.com/ec2.

[10] "Carbon Monitoring for Action," http://carma.org/.

[11] United States Environmental Protection Agency, "eGRID," http://www.epa.gov/cleanenergy/energy-resources/egrid/index.html.Eirgrid, http://www.eirgrid.com.

[12] Mikko Pervila¨ and Jussi Kangasharju, "Cold air containment," in Proceedings of ACM SIGCOMM Workshop on Green Network- ing, Toronto, 19 August 2011, pp. 7–12.

[13] L. A. Barroso and U. Ho¨ lzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," Synthesis Lectures on Computer Architecture, 2009.

[14] D. Atwood and J. G. Miner, "Reducing data center cost with an air economizer," August 2008, http://www.intel.com/content/www/us/en/data-center- efficiency/data-center-efficiency-xeon-reducing-data-center- cost-with-air-economizer-brief.html.

[15] A Qureshi, J. Guttag, R. Weber, B. Maggs, and H. Balakrish- nan, "Cutting the electric bill for internet-scale systems," in Proceedings of ACM SIGCOMM, Barcelona, 17-21 August 2009, pp. 123–134.

[16] R. Stanojevic´ and R. Shorten, "Distributed dynamic speed scaling," in Proceedings of IEEE INFOCOM, San Diego, 14-19 March 2010, pp. 1–5.

[17] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in Proceeding of SIG- METRICS, San Jose, 7 June 2011, pp. 233–244.

[18] M. Conti, E. Gregori, and F. Panzieri, "Load distribution among replicated Web servers: a QoS-based approach," SIG- METRICS Performance Evaluation Review, vol. 27, no. 4, pp. 12– 19, 2000.

[19] Z. M. Mao, C. D. Cranor, F. Bouglis, M. Rabinovich, O. Spatscheck, and J. Wang, "A precise and Efficient Evalu- ation of the Proximity between Web Clients and their Local DNS Servers," in Proceedings of USENIX, Monterey, 10 - 15 June 2002, pp. 229–242.

[20] M. Pathan, C. Vecchiola, and R. Buyya, "Load and Proximity Aware Request-Redirection for Dynamic Load Distribution in Peering CDNs," in On the Move to Meaningful Internet Systems: OTM, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5331, pp. 62–81.

[21] Greenpeace, "Make IT green cloud computing and its contribution to climate change," Retrieved February 2011, http://www.greenpeace.org/international/Global/international/planet-2/report/2010/3/make-it-green-cloud- computing.pdf.