Technical Disclosure Commons

Defensive Publications Series

August 2021

GREEN CLOUD COMPUTING: GREEN INDEXING OF CLOUD NETWORK RESOURCES

Navdeep Sood

Praveen Kumar

Sricharan Parakala

GREEN CLOUD COMPUTING: GREEN INDEXING OF CLOUD NETWORK RESOURCES

AUTHORS:
Navdeep Sood
Praveen Kumar
Sricharan Parakala

ABSTRACT

The present-day methods of green cloud computing are focused principally on data processing and data storage. However, the transmission and switching elements are often not considered in green cloud computing. Techniques are presented herein that support a mathematical model which focuses on the end-to-end cloud environment including, for example, transmission, switching, data storage, and data processing elements. In particular, energy efficiency, a minimal carbon footprint, optimal capital expenditures, and the fewest operating expenditures are the need of hour for both middle-scale and massive-scale cloud environments and data centers. The presented techniques encompass a mathematical model along with elaborating use cases to successfully meet all the objectives noted above.

DETAILED DESCRIPTION

Cloud networks employ an extensive equipment infrastructure that supports both cloud-based servers and an array of network equipment on Layers 0 (L0), 1 (L1), 2 (L2), and 3 (L3) for interconnecting the server infrastructure. The technology world is devoting significant efforts at the chip level to make the cloud and associated data centers energy efficient with minimal carbon footprints.

Thus, a need exists for a way to measure the carbon footprint of a given link and device or node which can serve several purposes starting from traffic engineering and carrying on to other areas. Such a mechanism should be equally applicable for private cloud, public cloud, hybrid cloud, and community cloud environments. Additionally, the method should be atomic in nature and most useful on scaled cloud networks.

To address the types of challenges that were described above, techniques are presented herein that support a base framework to quantify a green index for a given link and device. Aspects of the presented techniques will be described and illustrated in the narrative that is presented below, which will proceed in the following manner:

- Section 1. Background.
- Section 2. Sensors and green index calculation.
- Section 2.1. Absolute and aggregate value of the temperature of a line card (T_{ABS}).
- Section 2.2. Temperature deviation (T_{DEV}).
- Section 2.3. Linear conversion and a green index for temperature.
- Section 2.4. Green index – putting everything together.
- Section 3. Use of a green index.

1. Background

A cloud network is a complex mesh network involving, possibly among other things, servers of various sizes and scales, devices (such as routers, switches, optical nodes, etc.), and links interconnecting the devices in a complex mesh.

Aspects of the techniques presented herein support a mathematical model that defines a green index for a given node and a given link. The green index is a scale or rating of the form T_n-P_n-C_n where a lower value of a green index means greater energy efficiency with a minimal carbon footprint and a higher value of a green index means lesser energy efficiency with a worse carbon footprint.

The green index T_n-P_n-C_n, according to aspects of the techniques presented herein, comprises the following elements:

- T is a thermal-based green index derived from the thermal characteristics of a device or link;
- P is a power footprint-based green index derived from the power consumption or usage of a device or link; and
- C is a carbon footprint-based green index derived from the carbon footprint consumption or usage of a device or link.

The mathematical modeling for the techniques presented herein follow a down-top approach. For example:

- For computing a green index for a given link, a mathematical model or equation is used involving the thermal, power, and carbon footprint for the associated port, line module, device or chassis.
- For computing a green index for a given link, a mathematical model or equation is used involving the thermal, power, and carbon footprint for the chassis or common cards.

The T_n-P_n-C_n information may be presented as an advertisable metric by extending the type-length-value (TLV) encoding schemes of current routing protocols (such as, for example, the Open Shortest Path First (OSPF) protocol, the Intermediate System to Intermediate System (IS-IS) protocol, or the Border Gateway Protocol (BGP) protocol) or for that matter any other protocol or TLV extension.

Figure 1, below, depicts a very basic example within which aspects of the techniques presented herein may be applied.

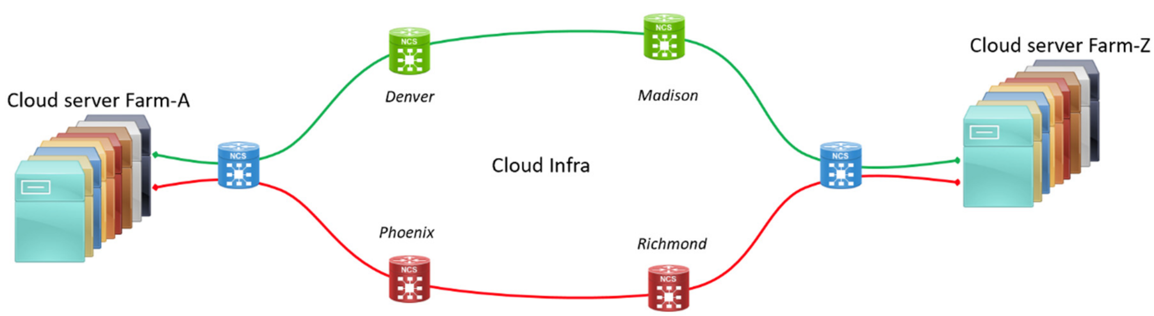

Figure 1: Illustrative Example

For simplicity, the example that is illustrated in Figure 1, above, only includes two cloud server farms as end points A and Z.

In an exemplary experiment, aspects of the techniques presented herein deployed a node-level mathematic model and, as depicted in Figure 1, above, colored the Denver - Madison path as a green T_n-P_n-C_n and colored the Phoenix - Richmond path as a red T_n-P_n-C_n. Accordingly, a server farm shall choose the green path as the preferred path in the green cloud computing paradigm. A similar approach may also be applied to network links.

2. Sensors and Green Index Calculation

Typically, multiple sensors are installed on the critical hardware components of a card. For example, software may monitor the temperature, voltage, current sensors, etc. that are supported by the hardware. The sensor values are periodically polled and reported. Under aspects of the techniques presented herein, temperature indexing is performed in great detail. Similar modeling can be provided for other parameters like power and a carbon footprint.

If any temperature sensor crosses a minor threshold, the software can increase the speed for all the associated fans in the system. Figure 2, below, presents the temperature sensors that are available on one example card. It is important to note that the specific sensor types and thresholds depend on the particular line card that is installed. Under aspects of the techniques presented herein, the mathematical model for the green index computation is quite generic and will work on any kind of line card or chassis. As depicted in Figure 2, below, there are nine temperature sensors that are available for the line card component. Each of the sensors displays a current temperature reading. Although the range of the operating temperature is very wide, the lower a temperature then the lower the carbon footprint. Temperature conditions depend on several factors such as, for example, a data center environment, a cooling design, filler card placement, etc.

Location	TEMPERATURE Sensor	Value (deg C)	Crit (Lo)	Major (Lo)	Minor (Lo)	Minor (Hi)	Major (Hi)	Crit (Hi)
0/1								
	Inlet Air Temp	36	-10	-5	-2	60	70	80
	Outlet Air Temp	51	0	5	10	75	85	95
	HotSpot	50	-10	-5	0	85	95	105
	DIGI0 Temp	43	-10	-5	0	100	105	110
	DIGI1 Temp	56	-10	-5	0	100	105	110
	Jericho Temp	67	-10	-5	0	100	105	110
	LOCAL TEMP DB	39	-10	-5	0	85	95	105
	MELKOR FPGA TEMP DB	47	-10	-5	0	90	100	125
	DENALI TEMP DB	31	-10	-5	0	103	105	110

Figure 2: Exemplary Card Temperature Sensors

Under aspects of the techniques presented herein, two critical aspects of the temperature are considered in the green index calculation. The first critical aspect of temperature considered in the green index calculation is an absolute or aggregate value of the temperature of the line card. This may be indicated as TG_{abs} where T denotes a temperature, G denotes green, and abs denotes an absolute value. The second critical aspect of temperature considered in the green index calculation is a temperature deviation, which may be indicated as TG_{DEV}.

Section 2.1: Absolute/Aggregate Value of the Temperature of a Line Card

There is no absolute or total temperature of a line card. Rather, the temperature varies across the different hardware components of the card. Typically, an inlet temperature is taken into consideration for shutting down the card or chassis. A fan speed algorithm depends upon the individual sensors of the card. To approximate the overall temperature of the line card there are multiple options. The simplest option comprises checking the percentage increase in the fan speed compared to the normal operating revolutions per minute (rpm). Another option is to average out the temperature of the various components. The averaging calculation presents some challenges. For example, one outlier can impact the calculation of the average. Additionally, an outlier (e.g., a high temperature on a small hardware component) can increase the average value of the temperature. A median formula may be used to offset the outlier impact. If outliers can be detected using standard interquartile range (IQR) calculations then the arithmetic mean can also be used to represent the average or absolute value of the line card temperature.

For purposes of illustration, temperature sensor data from four exemplary line cards is presented below, where each list represents the temperature sensor values.

```
LC1_temp = [35, 51, 50, 43, 56, 66, 39, 47, 31]
LC2_temp = [30, 41, 36, 32, 62, 53, 40, 42, 99]
LC3_temp = [30, 46, 45, 35, 50, 58, 35, 42, 25]
LC4_temp = [36, 60, 58, 58, 65, 65, 44, 51, 61]
```

Figure 3, below, illustrates visually the above temperature sensor values.

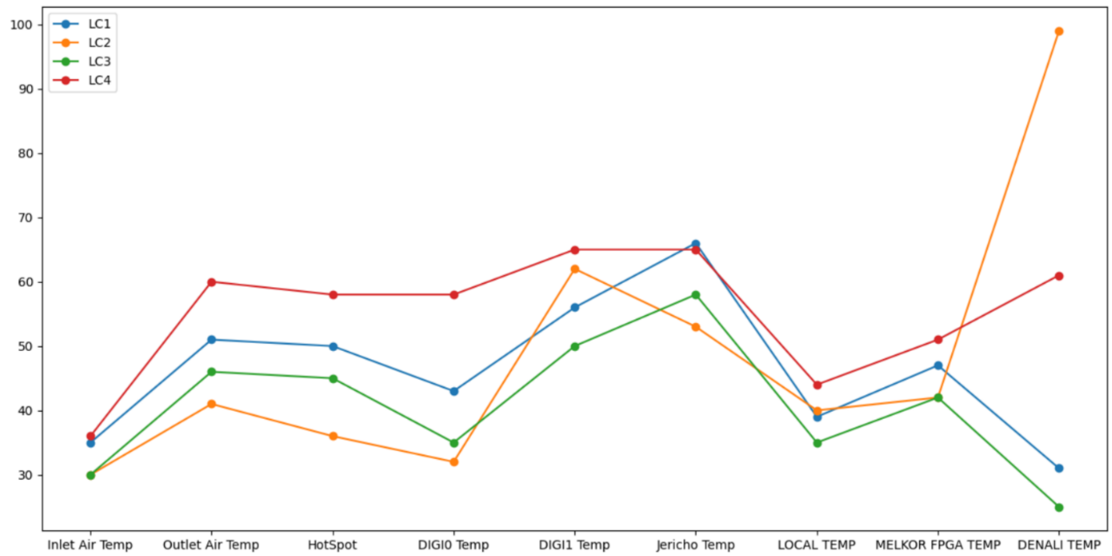

Figure 3: Illustrative Temperature Sensor Values

The arithmetic mean of a temperature data set may be calculated using the following formula:

$$T_{mean} = \frac{1}{N_{sensor}} \sum_{i=1}^{N_{sensor}} T_i$$

Applying the above formula to the four exemplary temperature data sets that were presented earlier yields:

```
# py - statistics.mean
mean-LC1 46.44
mean-LC2 48.33 >> no offset
mean-LC3 40.67
mean-LC4 55.33
Set of Tmean = [46.44, 48.33, 40.67, 55.33]
```

As seen in Figure 3, above, LC2 (i.e., line card number 2 or the slot 2 card) card's hardware component DENALI has a very high temperature compared to the other components. This can be considered as an outlier and it impacts the average or absolute value of the temperature. To offset any outliers, the median may be used to represent the average or absolute value of the card.

A data set 'T' contains an ordered list of values in the temperature data set. If the number of sensors (i.e., the value of N_{SENSOR}) is even then the median value may be calculated as:

$$T_{med} = T[n/2]$$

Alternatively, if the number of sensors (i.e., the value of N_{SENSOR}) is odd then the median value may be calculated as:

$$Tmed = \frac{\left(T[\frac{Nsensor-1}{2}] + T[\frac{Nsensor+1}{2}]\right)}{2}$$

Applying the above formulas to the four exemplary temperature data sets that were presented earlier yields:

```
# py - statistics.median
median-LC1 47
median-LC2 41 >> offsets outlier
median-LC3 42
median-LC4 58
```

In summary, an arithmetic mean (T_{MEAN}) or a median (T_{MED}) may be used to represent the absolute value (TG_{ABS}) of a temperature, where a higher TG_{ABS} value implies a larger carbon footprint.

Section 2.2: Temperature Deviation (T_{DEV})

Temperature deviation is also important for the green index calculation. Two kinds of variations are possible. The first kind considers too high a level of disparity across various hardware components. The second kind considers variation compared to the high temperature thresholds. Variation (i.e., a delta with respect to a high temperature threshold) matters because if a higher number of components are close to the high threshold, then the chance of a card heating or burning up will be higher and thus result in a larger carbon footprint. For simplicity of exposition, the narrative below includes just the calculations for the variation across hardware components. It is important to note that the same methods may be applied for the calculation of variation with respect to the predefined high-temperature thresholds.

Multiple methods are available to measure a spread. A few of the popular methods are variance, standard deviation, IQR, and mean absolute deviation (MAD). Variance and standard deviation are related. Outliers impact the variance and standard deviation because of the square term that is used in the calculation. IQR and MAD are not impacted by outliers.

Variance may be calculated according to the following formula:

$$VAR(T) = \frac{1}{N_{sensor}} \sum_{i=1}^{N_{sensor}} (T_i - \mu)^2$$

$$\mu = \frac{1}{N_{sensor}} \sum_{i=1}^{N_{sensor}} T_i$$

Applying the above formula to the exemplary temperature data sets that were presented earlier yields:

```
# py - statistics.variance
variance-LC1 118.03
variance-LC2 461.75
variance-LC3 107.5
variance-LC4 97.0
```

The standard deviation may be calculated according to the following formula:

$$\sigma = \sqrt{\frac{1}{N_{sensor}} \sum_{i=1}^{N_{sensor}} (x_i - \mu)^2}$$

Applying the above formula to the exemplary temperature data sets that were presented earlier yields:

```
# py - statistics.stdev
stddev-LC1 10.86
stddev-LC2 21.49
stddev-LC3 10.37
stddev-LC4 9.85
```

The IQR may also be used to monitor the spread of temperatures according to the following formula:

$$IQR = Q3 - Q1$$

The following code fragment highlights the development of the terms Q1 and Q3, as employed in the above formula, and the following results show the outcome of applying the above formula to the exemplary temperature data sets that were presented earlier:

```
#import numpy as np
#Q1 = np.percentile(sorted(LC1_temp), 25, interpolation = 'midpoint')
#Q3 = np.percentile(sorted(LC1_temp), 75, interpolation = 'midpoint')
#IQR = Q3 - Q1
IQR-LC1 12.0
IQR-LC2 17.0
IQR-LC3 11.0
IQR-LC4 10.0
```

MAD is also a useful metric to measure the spread. Such a metric may be calculated according to the following formula:

$$T_{MAD} = \frac{1}{N_{sensor}} \sum_{i=1}^{N_{sensor}} |T_i - m(T)|$$

As indicated in the above formula, the distance from the mean is calculated and then the average is taken. Here mean can be either the mean of the temperatures of the sensor values of a line card or the high temperature threshold. It is important to note that the distance from a high temperature threshold makes more sense with respect to green cloud computing. When the distance is calculated from the median, this term is referred to as median absolute deviation.

Applying the above formula to the exemplary temperature data sets that were presented earlier yields:

```
#import pandas as pd
#series = pd.Series(LC1_temp)
#series.mad()
MAD-LC1 8.4
MAD-LC2 15.33
MAD-LC3 8.37
MAD-LC4 7.78
Set of TMAD = [8.4,15.33,8.37,7.78]
```

In summary, the variance, the standard deviation, an IQR, or the MAD may be used to represent the deviation value TG_{DEV} of the temperature.

Section 2.3: Linear Conversion and a Green Index for Temperature

Under aspects of the techniques presented herein a green index resides in the range from one (1) to 100. The lower the value, the better for the environment (i.e., a smaller carbon footprint).

As noted previously, different options are available for the calculation of the absolute or aggregate temperature measure (T_{ABS}) and the temperature deviation (T_{DEV}). A Tabs measure may be developed using either an arithmetic mean or a median. A T_{DEV} measure may be developed using either a variance, a standard deviation, an IQR, or a MAD.

In the narrative that follows the Tabs measure will be based on an arithmetic mean (T_{MEAN}) and the T_{DEV} measure will be based on a MAD (T_{MAD}). Based on the different derivations that were presented above, the values of T_{MEAN} and T_{MAD} for the exemplary temperature data sets that were presented earlier are:

```
TMEAN set = [46.44, 48.33, 40.67, 55.33]
TMAD set = [8.4, 15.33, 8.37, 7.78]
```

The values of the absolute temperature and the deviation may not reside exactly within the range of 1 to 100. Accordingly, it may be necessary to convert the current range to the desired range of 1 to 100 without disturbing the ratios. The following code fragment illustrates such a linear conversion:

```
#def scale(l, old_min, old_max, new_min, new_max):
#for old_value in l:
#new_value = ( (old_value - old_min) / (old_max - old_min) ) * (new_max - new_min) + new_min
#print(old_value, round(new_value,2))
```

Following such a linear conversion a T_{abs} measure may be renamed to TG_{ABS} (where, as noted previously, T identifies temperature and G indicated green):

$$T_{ABS} \xrightarrow{LinearConversion<1-100>} TG_{ABS}$$

Applying the above approach to the exemplary temperature data sets that were presented earlier yields:

```
TABS = TMEAN
TMEAN set = [46.44, 48.33, 40.67, 55.33]
Tabs set after linear conversion = [42.39, 43.78, 38.16, 48.91]
```

Following a similar linear conversion that is applied to a T_{DEV} measure, a T_{DEV} measure may be renamed to TG_{DEV} (where, as noted previously, T identifies temperature and G indicated green):

$$T_{DEV} \xrightarrow{LinearConversion<1-100>} TG_{DEV}$$

Applying the above approach to the exemplary temperature data sets that were presented earlier yields:

```
TDEV = TMAD
TMAD set = [8.4, 15.33, 8.37, 7.78]
TDEV after linear conversion = [14.49, 19.58, 14.47, 14.04]
```

Once the linear conversions are completed (i.e., after TGabs and TGdev both reside in the range of 1 to 100), a green index for the temperature (i.e., GTindex) may be derived from the mean of TGabs and TGdev. A GTindex may be calculated according to the following formula:

$$GT_{index} = \frac{TG_{ABS} + TG_{DEV}}{2}$$

Applying the above formula to the exemplary temperature data sets that were presented earlier yields:

```
GTindex = Mean of [42.39, 43.78, 38.16, 48.91] and
                  [14.49, 19.58, 14.47, 14.04]
        = [28.44, 31.68, 26.31, 31.47]
```

Table 1, below, presents the results of the above calculations in tabular form.

LC Slot	GTindex (where a smaller value indicates a better green compliance and a smaller carbon footprint)
LC1	28.44
LC2	31.68
LC3	26.31
LC4	31.47

Table 1: Exemplary GTindex Measures

Section 2.4: Green Index – Putting Everything Together

As modeling was performed for the temperature (as described above), similar modeling may be done for a green power index GP_{index} that considers elements such as power, voltage, and current along with other factors such as, for example, fans. Similarly, other factors may be part of a green carbon footprint index GC_{index}.

The green index of a line card (GL_{index}) may be calculated according to the following formula:

$$GL_{index} = \frac{GT_{index} + GP_{index} + GC_{index}}{3}$$

The same GL_{index} may be applied to all of the interfaces of a line card. The green index of a node or chassis (GN_{index}) may be calculated from the mean of all of the line card green index values. Additionally, node- or chassis-related factors may also be considered in the node-level green index calculation. A GN_{index} measure may be calculated according to the following formula:

$$GN_{index} = \frac{1}{N_{slots}} \sum_{i=1}^{N_{slots}} GL_{index_i}$$

Section 3. Use of a Green Index

As described in the previous sections, green indexing on the links may be used for traffic engineering in cloud, datacenter, or service provider networks. Green links, nodes, or domains should be preferred over non-green resources. The overall purpose is to reduce the carbon footprint. Once the green indexing is completed, the index or metric values may be advertised as traffic engineering (TE) extensions of the link state of interior gateway protocol (IGP) protocols (such as IS-IS or OSPF) or BGP. The Constrained Shortest Path First (CSPF) approach with respect to the green index will work exactly the same as other metrics (such as, for example, cost, latency, etc.).

Aspects of the techniques presented herein may be applied to many different use cases. Several of those use cases will be described below.

A first, most relevant, use case for a green index comprises avoiding the use of resources in the network (such as, for example, a link, a node, or a datacenter) that have a low green index. Services (e.g., L2, L3, storage, etc.) should try to take green resources, links, or nodes first. The CSPF approach can calculate the shortest green path based on a topology database. In the case of protected services, an active service should come up on the green path and the backup (e.g., idle) path should take a less green or worst green (e.g., red) path. Services will switch over to a less green path following a working path failure but this can be again re-optimized (through, for example, a make-before-break approach) on the better green path. A green index value will be very similar to other metrics like cost, latency, hop count, etc. A green index is not only meant for the tunnel kind of services. It can be used for the forwarding of control plane protocols as well. Although control plane protocols consume a smaller amount of bandwidth each bit counts and leaves a larger carbon footprint if a less green path is followed.

A second use case considers optional re-optimization that may be done on tunnels or services if any better green path becomes available due to, for example, network changes. Re-optimization is usually done in a make-before-break (MBB) fashion that doesn't impact the running traffic. The purpose of re-optimization is to keep the services running on a greener path irrespective of the dynamic changes happening in the network.

A third use case considers the rerouting on a particular segment (e.g., a fast reroute) that may be done if the green index increases suddenly (due to, for example, temperature, power, or carbon footprint changes). Local link or segment restoration will ensure that end-to-end signaling doesn't happen and the end goal of the minimum green index along the path is also achieved at the same time.

A fourth use case highlights the fact that a demarcation of green vs. non-green or less-green network resources is missing at present. Resources such as a node, card, link, etc. may be arranged in the sequence of carbon footprint. Based on such a categorization certain business decisions may be made like phasing out or migrating the less-green network resources first.

A fifth use case highlights the fact that network-wide categorization or grouping may be achieved. Green indexing starts with a link and the calculations may be done for a domain (e.g., a group of nodes) as well. For example, a progression something like link --> card --> node or chassis --> IGP area or geographic area --> autonomous system (AS). Two domains can be compared in terms of the overall green index. Internet traffic forwarding decisions may be made based on checking the aggregate green index of two AS. In summary, the green index is not limited to a link, domain-wise calculations are possible, and certain traffic engineering rules can be imposed based on a domain green index.

A sixth use case considers the opportunity that a vendor has in leading the effort in standardizing the green index within the Internet Engineering Task Force (IETF). Once done, others will follow. Once the indexing method is standardized, a comparison of any product or live network's carbon footprint may be easily done. Any governing body that is responsible for green cloud computing, carbon footprints, or sustainable development (see, for example, https://sdgs.un.org/goals) can monitor the green data.

A seventh use case consider the Flex Algo mechanism that allows IGPs themselves to compute constraint-based paths over a network. This is used to create separate planes based on some criteria like "create two planes, one with green affinity and another one with red affinity." Manual affinity assignments on the links are error-prone. A green index can be used as a criterion for defining the green versus non-green plane. Once the plane is defined, traffic engineering can be easily done. Such an approach will simplify network operations. A sample configuration is presented below:

```
router igp
flex-algo 130 # plane 1
advertise-definition
metric-type green_index < 50
!
flex-algo 131 #plane 2
advertise-definition
metric-type green_index >= 50
!
```

An eighth use case considers multi-control plane structures like datacenter switches going over data center interconnect (DCI) optical boxes where green_index recording (e.g., IETF Request for Comments (RFC) 8001 for a shared risk resource group (SRLG)) across a control plan may be implemented to achieve the carbon footprint access across control planes.

A ninth use case considers a reduction or optimization that is possible in capital expenditures (CapEx) – e.g., the purchase of more green equipment – and operating expenditures (OpEx) – e.g., air conditioning, power consumption, etc. costs – in a data center based on the green index results and data analysis. The green cost per bit can also be analyzed with the help of a green index.

A tenth use case considers the use by routing and switching algorithms of a green index (e.g., green-aware routing by default).

An eleventh use case highlights the fact that a green index algorithm calculates the node or chassis level green index as well which can also be part of the TLVs of the protocols and hence traffic forwarding decisions can include a node green index as one of the criteria.

A twelfth use case notes that a green index, as described and illustrated above, is mostly related to data processing, data storage, transmission, and switching. However, similar algorithms may be implemented at the application level as well. Applications using more CPU, memory, software, hardware, etc. resources will be less green and hence have a higher green index.

The use cases that were described and illustrated in the above narrative are exemplary only. It is important to note that aspects of the techniques presented herein can benefit many other use cases as well.

In summary, techniques have been presented that support a mathematical model which focuses on the end-to-end cloud environment including, for example, transmission, switching, data storage, and data processing elements. In particular, energy efficiency, a minimal carbon footprint, optimal capital expenditures, and the fewest operating expenditures are the need of hour for both middle-scale and massive-scale cloud environments and data centers. The presented techniques encompass a mathematical model along with elaborating use cases to successfully meet all the objectives mentioned above.