CLOUD computing

Mehmet Sahinoglu[1]* and Luis Cueva-Parra[2]

CLOUD computing (Grid or utility computing, computing on-demand) which was the talk of the computing circles at the end of 1990s has become once again a relevant computational topic. CLOUD computing, also considered as a fifth utility after water, electric power, gas, and telephony, is on the basis of the hosting of services on clusters of computers housed in server farms. This article reviews CLOUD computing fundamentals in general, its operational modeling and quantitative (statistical) risk assessment of its much neglected service quality issues. As an example of a CLOUD, a set of distributed parallel computers is considered to be working independently or dependently, but additively to serve the cumulative needs of a large number of customers requiring service. Quantitative methods of statistical inference on the quality of service (QoS) or conversely, loss of service (LoS), as commonly used customer satisfaction metrics of system reliability and security performance are reviewed. The goal of those methods is to optimize what must be planned about how to improve the quality of a CLOUD operation and what countermeasures to take. Also, a discrete event simulation is reviewed to estimate the risk indices in a large CLOUD computing environment favorably compared to the intractable and lengthy theoretical Markov solutions. © 2010 John Wiley & Sons, Inc. *WIREs Comp Stat* 2011 3 47–68 DOI: 10.1002/wics.139

INTRODUCTION AND MOTIVATION

CLOUD computing, an emerging form of computing using services provided through the largest network (Internet or CLOUD) is becoming a promising alternative to the traditional in-house IT computing services. CLOUD computing is a form of computing in which providers offer computing resources (software and hardware) on-demand. All of these resources are connected to the Internet and are provided dynamically to the users. Figure 1 shows a schematic representation of CLOUD computing. Here, CLOUD computing providers are connected to the Internet and able to provide computing services to both enterprise and personal users. Some companies envision this form of computing as a single major type of service which will be demanded extensively in the next decade. In fact, companies like Google, IBM, Microsoft, HP, Amazon, and Yahoo among others have already made investments not only in CLOUD research but also in establishing CLOUD computing infrastructure services (see Figure 1).

CLOUD computing services fall into three major categories[1]: (1) infrastructure as a service (IaaS), (2) software as a service (SaaS), and (3) platform as a service (PaaS). In IaaS virtualized servers, storage and networks are provided to the clients. SaaS is focused on allowing clients to use software applications through web-based interfaces. A service targeted to developers who focus primarily on application development only, without dealing with platform administration (operating system maintenance, load balancing, scaling, etc.), is called PaaS. Advances in virtualization, distributed computing, and high-speed network technologies have given further impetus to CLOUD computing. The major advantages of CLOUD computing are scalability, flexibility, resilience, and affordability. However, as users (companies, organizations, and individual persons) turn to CLOUD computing services for their businesses and commercial operations, there is a growing concern from the security and reliability perspectives as to how those services actually rate. The serviceability measurement can be categorized into three areas: performance, reliability, and security. Performance and reliability are two characteristics related to the condition of the providers' infrastructure and

*Correspondence to: msahinog@aum.edu

[1] Informatics Institute, Auburn University Montgomery, Montgomery, AL, USA

[2] Department of Mathematics, CS Option, Auburn University Montgomery, Montgomery, AL, USA

DOI: 10.1002/wics.139

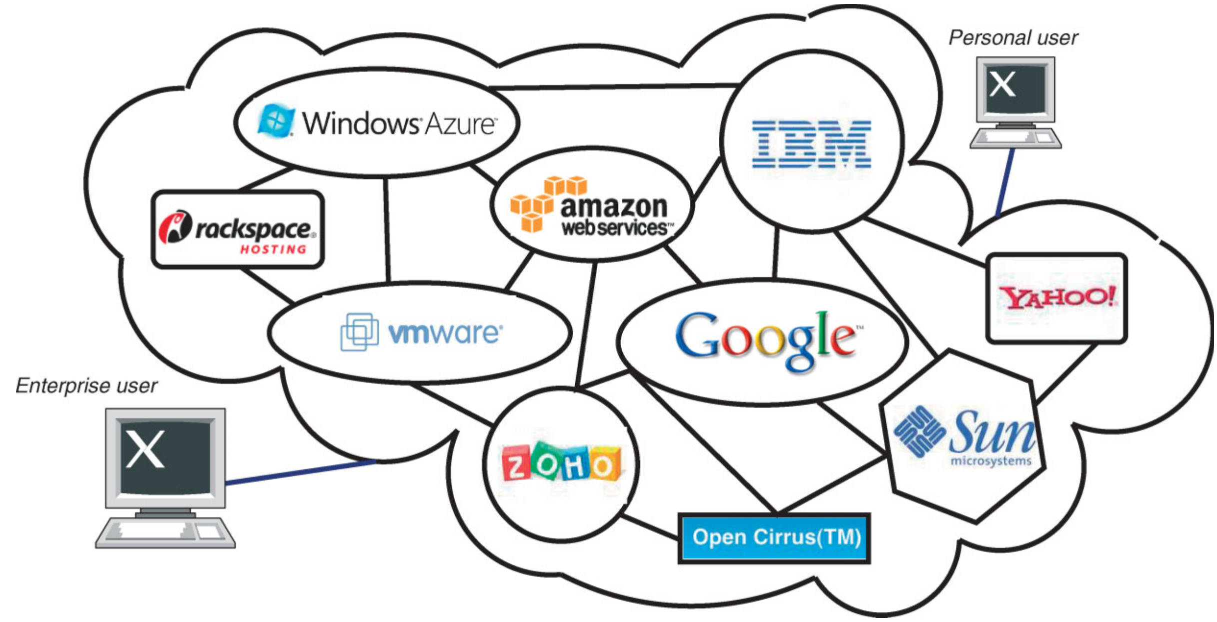


FIGURE 1 | Schematic representation of CLOUD computing.

the way they maintain and update them. Security (data protection and disaster recovery), on the other hand, is one aspect that is more difficult to measure. Both CLOUD computing providers and users need a way to measure the quality of this service, mainly in the area of reliability and security. This metric can provide the sending and receiving end-users with a better sense of what both parties are getting for their return of investment. Also, it gives providers a concrete numerical reference, rather than vague attributes, so they can improve the quality of the current service. However, despite evident benefits, CLOUD computing lacks rigorous analysis regarding its quality of service. Specifically, a quantitative assessment of the quality of service (QoS) in such enterprises leaves much to be desired. In general, the quality of CLOUD computing services is difficult to measure, not only qualitatively, but most importantly quantitatively. A qualitative indicator of security, for example, in terms of colors or any other arbitrary non-numerical classification such as 'high, medium, or low', or yet another scale with 'severe, high, elevated, guarded, and low' in Figure 2 helps if no others exist, but does not provide much other than a crude approximation.[2] If one needs to implement an adequate risk management program, it is imperative to have a numerical index to quantify the level of security against adversarial malware, and a degree of lack of reliability due to chance failures.[3,4]

We cannot improve something that we cannot measure or quantify.[5] But the actual task is first to define what and how to measure. There are various studies that analyzed different parameters and discussed their use for quantifying performance.[2–4,6–9] We review several studies published for evaluation of the QoS in CLOUD computing, as well as major trends in the overall CLOUD computing area including the Grid computing.

NOTATION AND HISTORY OF RISK ASSESSMENT METHODS

Notation

rv: Random variable
LSD: Logarithmic series distribution
pdf: Probability density function
cdf: Cumulative probability density function
GB: Gigabyte (hardware storage capacity equal to 1 GB = 1000 million bytes)
MW: Megawatt (electric power capacity measure of a generator equal to 1 million watts)

Severe
High
Elevated
Guarded
Low

FIGURE 2 | Qualitative or descriptive security metrics.

Poisson ^ LSD: Poisson compounded by logarithmic series distribution

Poisson ^ geometric: Poisson compounded by geometric distribution

x_i: LSD rv

θ: Correlation-like unknown parameter of the LSD rv, $0 < \theta < 1$

α: Parameter of the LSD, as a function of θ

X: Poisson ^ LSD rv, that is, sum of x_i's which are LSD. Negative binomial rv for a given assumption of k as a function of Poisson parameter

k: Parameter of the Poisson ^ LSD

q: Variance to mean ratio, a parameter of the Poisson ^ LSD

λ: Parameter of the Poisson

f: Frequency

d: Average duration time per outage

LoS: Loss of Service event, when capacity cannot meet load (service) demand plus equipment outages at a given hour or cycle.

LoSP: Loss of service probability.

LoSE: Loss of service expected = LoSP × NHRS (number of hours in a year, 8760 hours)

QoSE: quality of service expected = 1− LoSE

TOTCAP: Installed total capacity for an isolated computing system not interconnected

L_i: Service demand forecast rv at each discrete ith hour for the isolated network

O_i: Unplanned forced capacity outage rv at the ith hour

m_i: Capacity margin rv at the ith hour, where $m_i < 0$ signifies a loss of load hour; that is, $m_i = \text{TOTCAP} - O_i - L_i$, all variables in GB or MW

CLOURA: CLOUD Risk Assessor Software; www.areslimited.com and click Computer CLOUD.

Theoretical History of CLOUD Computing—Risk Assessment Methods

Several studies on CLOUD computing security metrics have their roots on the reliability analysis of one of the most familiar networks: the electrical power grid or simply, the power system. Reliability of power systems has been studied extensively for many years and numerous works have been published on methods to compute power system reliability indices, specifically LoSE.[10–18] Consider an electric power supply or any service-based system with operating (i.e., positive power margin) and nonoperating (i.e., negative power margin) states throughout a yearly long period of operational discrete hours.[10,17,18] The quantities $m_1, m_2, \ldots, m_n$, are the margin values at hourly steps where a positive margin denotes a nondeficient state and negative margin denotes a deficient state. Margin m_i, at a discrete hour is the difference between the total generating capacity (TOTCAP) and the service demand (hourly peak load forecast, L_i) plus unplanned forced capacity outages, O_i.

Hence m_i indicates the capacity (MW) margin or balance at each discrete step i, where the realization $\{m_i\}$ assumes either a positive margin or negative margin. Thus the count of negative margin hours within a year is the loss of load expected (LoSE), which when divided by total number of hours (NHRS) will yield the loss of load probability (LoLP), or loss of service probability (LoSP). Once the system is in a negative-margin state, it should usually have a period of several hours to form a clump of downs before overall system recuperation fully happens following a repair process.

MODELING RISK OF SERVICE: FREQUENCY AND DURATION METHOD

A few papers have treated the risk of service issue in a stochastic sense where the number of loss of load or service expected hours in a year (LoSE = LoSP × NHRS) was expressed in terms of a pdf. Among those works, the distribution of the sum of negative-margin (loss of load) hours in an isolated power system was approximated to be a limiting compound Poisson ^ geometric process as the sum of Markov nonidentical Bernoulli random variables were published in 1983 and 1990, respectively.[10,19] It considered variables changing from hour to hour in contrast to a previous study[20] which used a Binomial assumption with a homogenous rate of flow.

The said Poisson process is compounded by a discrete LSD.[21–24] Note that LSD recognizes the true positive contagion property of the occurrences in a clump; such as, a breakdown at a given epoch adversely affects and attracts more failures in the forthcoming hours until the current failures are repaired as is the case in these analysis.[2,10,19,22–25] The parameters of Poisson ^ LSD, the mean and q-ratio (variance/mean), are provided by utilizing the well-known frequency and duration method as in (an electric power generation) system reliability evaluation. These necessary parameters of compound Poisson distribution are obtained from the popular frequency and duration method pioneered by Hall et al. in 1968,[26] later to be followed by Ayoub and Patton in 1970 and 1976,[27] and documented by Billinton et al.[11,12,15,16]

The LoS events are assumed to occur in terms of clusters or multiple happenings.[23,24] The proposed closed-form analytical approach is facilitated by using a nonapproximate but exact closed-form probability distribution function. That is, Poisson ^ LSD model, generally called Negative-Binomial pdf for convenience.[2,21–25,28] Authors support that this new probabilistic model provides a better accuracy toward a more realistic modeling of the operation of a power generation system in utilizing the advantages of the well-recognized frequency-duration method.[13,14,17,26,27] The same probabilistic model can be applied to CLOUD computing systems, provided some explicit conditions, which we will briefly discuss later. Otherwise Refs 3,4,7–9,29–41 will present a good cross section of most recent papers on the CLOUD computing current advances in terms of fundamentals and practice.

Review of Compound Poisson as a Stochastic Model

Any pure Poisson process with no specific compounding distribution in mind has interarrival times (e.g., between customer arrivals in a shopping mall) as negative-exponentially distributed with a rate λ. That is, the pdf of interarrival times is independent of earlier arrival epochs with forgetfulness property. Suppose arrivals or incidents of power breakdown occur in a power generation system according to a Poisson counting process. Each arrival can demand or endure a positive integer amount 'x' deficient hours, which are independently and identically distributed as $\{f_x\}$. When we consider any fixed time interval t, the number of demands (in batches or clumps) in that time interval is said to have a compound Poisson distribution. If the mean breakdown arrival rate is given as λ, then the compound Poisson probability of $X = x_1 + x_2 + x_3 + \cdots$ demands within a time interval or period $= t$ over total arrivals is given by,

$$P(X) = \sum_{Y=0}^{X} (\lambda t)^{Y} \mathrm{e}^{-\lambda t} f^{Y^*}(X)/Y!;\ X = 0, 1, 2, \ldots;$$
$$Y = 0, 1, 2, \ldots, X;\ \lambda > 0. \tag{1}$$

where, $f^{Y^*}(X)$ is the Y-fold convolution of $\{f_x\}$, which denotes the probability that 'Y interruptions place a total of X failures'. Of course, in the case where each interruption places exactly one failure (hence, the orderliness property of Poisson), Eq. (1) reduces to a purely Poisson density function. It now remains to find the parameters of the compound Poisson process, 'mean' and 'q = variance/mean', where q is equal to the second moment divided by the first moment of the compounding distribution of x

$$q = E(x^2)/E(x). \tag{2}$$

Review of Negative Binomial as a Compound Poisson

Negative binomial distribution (NBD) has been used in many disciplines involving count data, such as accident statistics, biological sciences, ecology, market research, computer software, and psychology. NBD was originally formulated as the distribution of the number of tosses of a fair coin necessary to achieve a fixed number of heads.[2,21,23,24] Later on analyzing the effects of various departures from the conditions that lead to the Poisson distribution for the occurrence of individuals in divisions of time (and space), Gosset concluded that if different divisions have different chances of containing individuals, the NBD provides a better fit than does the Poisson.[28] This is why there is a strong similarity of this expression with the negative margin hours adversely affecting each other in a power utility operation. Hence, NBD can also be defined to be the Poisson sum of a logarithmic series distributed rv. Now let $X = x_1 + x_2 + \cdots + x_n$, where x_i are independent identically distributed (iid) logarithmic-series (LSD) rv with its pdf and corresponding moments as follows:

$$f(x) = p(x;\theta) = \frac{\alpha\theta^x}{x};\ x = 1, 2, 3, \ldots, \infty;$$
$$\alpha = -\frac{1}{\ln(1-\theta)};\ 0 < \theta < 1 \tag{3}$$
$$E(x) = \mu = \frac{\alpha\theta}{(1-\theta)} \tag{4}$$
$$\mathrm{Var}(x) = \mu\left[\frac{1}{1-\theta} - \mu\right]. \tag{5}$$

Then, randomly stopped sum of x_i, which are distributed with respect to a discrete LSD rv with parameter q, will possess a negative binomial distribution with parameters k and $1 - \theta = q^{-1}$ if the governing counting-process is a Poisson with its rate parameter equal to:

$$\lambda = -k\ln(1-\theta) = k\ln(1-\theta)^{-1} = k\ln q, k < 0. \tag{6}$$

Now let LSD probability density function in Eq. (3) be reorganized and reparameterized as follows:

$$\theta = (p/q), q = p + 1 = (1-\theta)^{-1},\ \text{where}$$
$$p = \theta(1-\theta)^{-1} \tag{7}$$
$$\alpha = -1/\ln(1-\theta) = 1/\ln(1-\theta)^{-1} = 1/\ln q \tag{8}$$
$$f(x) = \{1/(x\ln q)\}(p/q)^x;\ x = 1, 2, 3, \ldots\ \text{and}$$
$$q = p + 1 > 1 \tag{9}$$

$$E(x) = -\theta/[(1-\theta)\ln(1-\theta)]$$
$$= p/\ln q = (q-1)/\ln q \qquad (10)$$
$$\mathrm{Var}(x) = \{(q-1)/\ln q\}\{q - [(q-1)/\ln q]\} \qquad (11)$$
$$q = \text{variance/mean} = F''(0)/F'(0)$$
$$= E(x^2)/E(x) = 1/(1-\theta) \qquad (12)$$

where ln denotes natural logarithm. Then q can be estimated as a root of moment in either Eq. (10) or Eq. (12), where $E(x)$ is the empirical number of failures divided by the number of arrivals. The Poisson ^ logarithmic series distribution (LSD) probability distribution, which has a mean, kp, and variance to mean ratio, q (variance/mean), can be expressed as follows:

$$P(X) = [(k+X-1)!/((k-1)!X!)](p^X/q^{k+X}). \quad (13)$$

The negative binomial distribution is particularly convenient because of the simple recursion formula easily derived from Eq. (13) as follows[2,21,22,24]:

$$P(X+1) = \{(X+k)/(X+1)\}(p/q)P(X). \qquad (14)$$

Hence, for a given $q = $ variance/mean ratio and $k = \lambda t/\ln q$ or given the mean 'M' of the Poisson process, $k = \text{mean}/p = \text{mean}/(q-1)$, the discrete state probabilities can be computed through a simple coding using Eq. (14).[22,24] On the other hand, there are other noncompound Poisson probabilities that also yield negative-binomial state probabilities by other chance mechanisms. However, these mechanisms are out of scope and not suitable to be dealt with in this review. In the Poisson ^ LSD, the parameter λ of the Poisson is taken as the value of the frequency of loss of load within a year. The average outage duration is taken as $d = E(x)$ of the LSD where θ or q can be extracted from Eq. (10). Thus frequency (f) represents the number of times that the system is in negative margin within a year. Average duration (d) represents the average number of hours that the system stays in a negative-margin at each system breakdown.

Finally LoSE $= f \times d$. Using Eq. (10) where x is from LSD and letting $E(x) = d = (q-1)/\ln q$, one obtains the q value through a nonlinear solution algorithm using Newton–Raphson technique. Consequently, let the rate of occurrence $\lambda = f$, where the Poisson parameter is set equal to the frequency of loss of load. Again using Eq. (6) and Eq. (10):

$$f = k\ln q \qquad (15)$$
$$\text{LoLE} = f \times d = k(q-1). \qquad (16)$$

PROBABILISTIC MODELING ON CLOUD COMPUTING SYSTEMS

In order to calculate an index of security or lack thereof on CLOUD computing systems, recent works[3,4] used reliability concepts discussed in the previous sections. The proposed models are based on an analytical closed-form probability distribution function like those explored in Refs 2,19,22. Although the conventional reliability indices such as loss of load expected (LoLE) through deterministic methods such as frequency-duration technique provide some useful averaging information; it is very important that a statistical distribution function should be available.[2,13,22,25] The purpose is to see how to completely characterize the behavior of the loss of load hours in a year of operation so as to be able to conduct statistical tests of hypotheses concerning the unknown (population) value of these indices.[2,10,13,25] It is believed that a statistical approach in developing a closed form and exact (nonasymptotic) pdf for the random variable of interest, loss of service (LoS), is novel and more accurate. Additionally, the reviewed compound Poisson model (i.e., NBD) respects the autocorrelation of chance failures or security breaches in sequential occurrence of a clump, which adversely influences each other by attracting more failures, for example, once an interruption occurs in a CLOUD computing system.

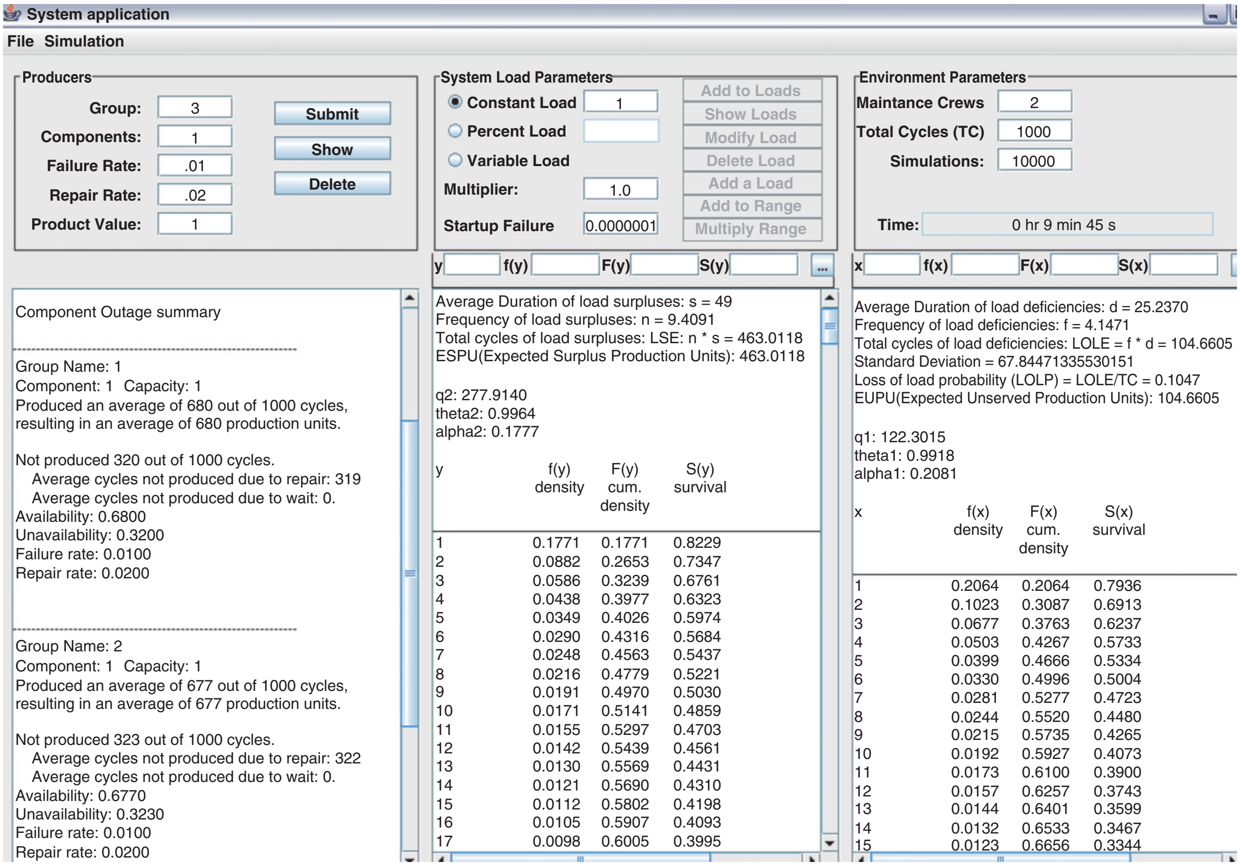
The authors believe that this probabilistic modeling approach offers a more realistic alternative to the traditional reliability evaluation, for instance, in a service-based electric power system, where the reliability index calculated is conventionally quoted as a single averaging number that bears no uncertainty generated by the variation of the input data. Also, the authors assume that the arrivals to a negative margin state are governed by a Poisson counting process, while inter-arrival times are exponentially distributed. However, the clump size of the negative-margin hours will be governed by the compounding logarithmic-series distribution (LSD) that assumes the interdependence effect. A more detailed review of these recent studies[2–4] will follow. We will review the implementation and validation of the model described.

DISCRETE EVENT SIMULATION (DES) ON A CLOUD COMPUTING SYSTEM

The implementation of the previously discussed model was performed by designing and developing a software assessment tool. This computational tool, called CLOUD Risk Assessor (CLOURA), will aid with the calculation of a security index for a given CLOUD computing system. Basically, one single security principle is considered: availability. That is, the total time in which a CLOUD computing system is on service without being disturbed or without loss of service (LoS) due an event of security breach. CLOURA is both a simulation tool and an expert system. As simulator, it will mimic real-life operation of the system in question and as expert system, it will help to quantify and manage the LoS index with the right responses to 'What If?' questions. It will assist taking care of the reliability or quality assessment so as to mitigate risk by quantifying the entire CLOUD operation with tangible and practical metrics.

Following the deterministic input data such as the number of servers or generators (or producers) with their failure and repair rates, and number of repair crews, and hourly (or cyclical) service demand which can be modified to reflect service maintenance or changes as implemented in Figures 3 and 5 all in gigabyte units as in Figure 10's hourly demand (load data); the operation starts at the first hour (or cycle) by randomly generating the negative exponentially distributed operating (up) and repair (down) times where the goal is to study the random evolution of a memoryless system.[2,6] Then the available total capacity at each cycle is contrasted against the outages to calculate the available capacity, which is also compared to the service demand at that hour to finally determine the reserve capacity.

If the reserve capacity (or margin) is less than a zero margin, then we have an undesired deficiency or loss of service. Once these hours (or cycles) of negative margin are added, it will constitute the expected number of hours of loss of service, LoSE. Divided by the total number of exposure units such as 8760 h (NHRS) for a year, it will give the LoSP = LoSE/NHRS. Once the LoSE is known, and its frequency (f = number of such occurrences of deficiencies per annum), then the average duration, $d = \text{LoSE}/f$, will indicate how long on the average a loss of service is expected to last. As in Eq. (10), $E(x) = d = (q-1)/\ln q$, one can best estimate the value of q using a Newton-Raphson nonlinear estimation technique, which later leads to the estimation of θ as in Eq. (12) and α as in Eq. (4). Then using Eq. (3), $f(x)$ for $x = 1, 2, 3, \ldots$ can be estimated to characterize the distribution of x. Further these probability values for d can be plotted; however, due to



Producers

Group: 3
Components: 1
Failure Rate: .01
Repair Rate: .02
Product Value: 1

System Load Parameters

● Constant Load 1
○ Percent Load
○ Variable Load
Multiplier: 1.0
Startup Failure 0.0000001

Environment Parameters

Maintance Crews 2
Total Cycles (TC) 1000
Simulations: 10000
Time: 0 hr 9 min 45 s

y f(y) F(y) S(y)

x f(x) F(x) S(x)

```
Component Outage summary

-----------------------------------------------------------
Group Name: 1
Component: 1  Capacity: 1
Produced an average of 680 out of 1000 cycles,
resulting in an average of 680 production units.

Not produced 320 out of 1000 cycles.
   Average cycles not produced due to repair: 319
   Average cycles not produced due to wait: 0.
Availability: 0.6800
Unavailability: 0.3200
Failure rate: 0.0100
Repair rate: 0.0200


-----------------------------------------------------------
Group Name: 2
Component: 1  Capacity: 1
Produced an average of 677 out of 1000 cycles,
resulting in an average of 677 production units.

Not produced 323 out of 1000 cycles.
   Average cycles not produced due to repair: 322
   Average cycles not produced due to wait: 0.
Availability: 0.6770
Unavailability: 0.3230
Failure rate: 0.0100
Repair rate: 0.0200
```

```
Average Duration of load surpluses: s = 49
Frequency of load surpluses: n = 9.4091
Total cycles of load surpluses: LSE: n * s = 463.0118
ESPU(Expected Surplus Production Units): 463.0118

q2: 277.9140
theta2: 0.9964
alpha2: 0.1777

y                 f(y)      F(y)      S(y)
                  density   cum.      survival
                            density

1                 0.1771    0.1771    0.8229
2                 0.0882    0.2653    0.7347
3                 0.0586    0.3239    0.6761
4                 0.0438    0.3977    0.6323
5                 0.0349    0.4026    0.5974
6                 0.0290    0.4316    0.5684
7                 0.0248    0.4563    0.5437
8                 0.0216    0.4779    0.5221
9                 0.0191    0.4970    0.5030
10                0.0171    0.5141    0.4859
11                0.0155    0.5297    0.4703
12                0.0142    0.5439    0.4561
13                0.0130    0.5569    0.4431
14                0.0121    0.5690    0.4310
15                0.0112    0.5802    0.4198
16                0.0105    0.5907    0.4093
17                0.0098    0.6005    0.3995
```

```
Average Duration of load deficiencies: d = 25.2370
Frequency of load deficiencies: f = 4.1471
Total cycles of load deficiencies: LOLE = f * d = 104.6605
Standard Deviation = 67.84471335530151
Loss of load probability (LOLP) = LOLE/TC = 0.1047
EUPU(Expected Unserved Production Units): 104.6605

q1: 122.3015
theta1: 0.9918
alpha1: 0.2081

x                 f(x)      F(x)      S(x)
                  density   cum.      survival
                            density

1                 0.2064    0.2064    0.7936
2                 0.1023    0.3087    0.6913
3                 0.0677    0.3763    0.6237
4                 0.0503    0.4267    0.5733
5                 0.0399    0.4666    0.5334
6                 0.0330    0.4996    0.5004
7                 0.0281    0.5277    0.4723
8                 0.0244    0.5520    0.4480
9                 0.0215    0.5735    0.4265
10                0.0192    0.5927    0.4073
11                0.0173    0.6100    0.3900
12                0.0157    0.6257    0.3743
13                0.0144    0.6401    0.3599
14                0.0132    0.6533    0.3467
15                0.0123    0.6656    0.3344
```

FIGURE 3 | Simulation results for a simple two component independent-additive system.

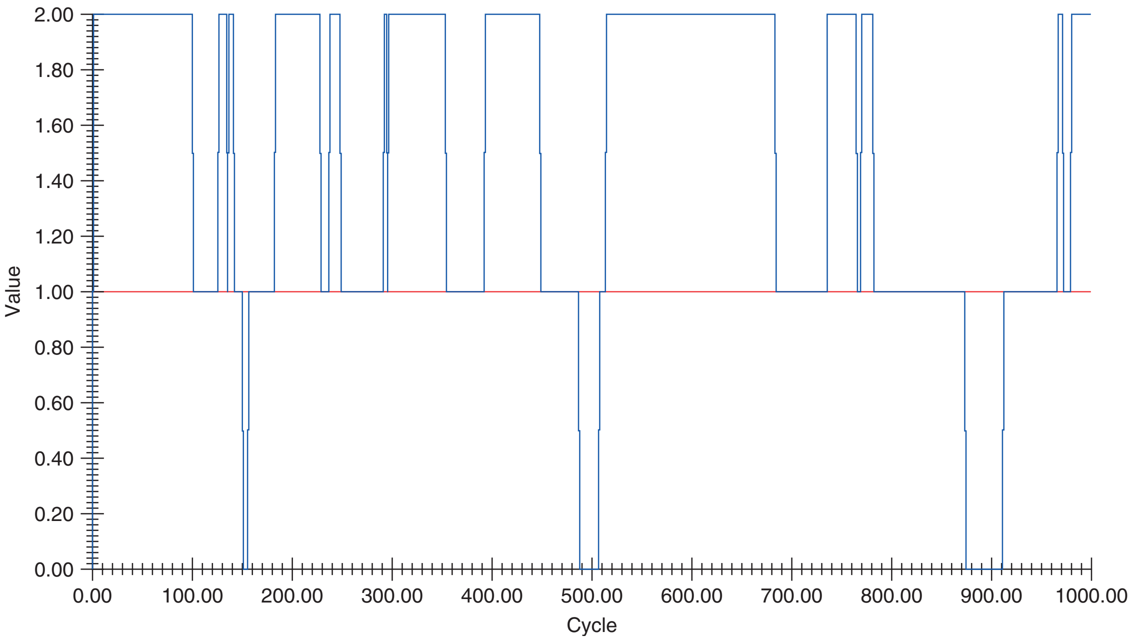
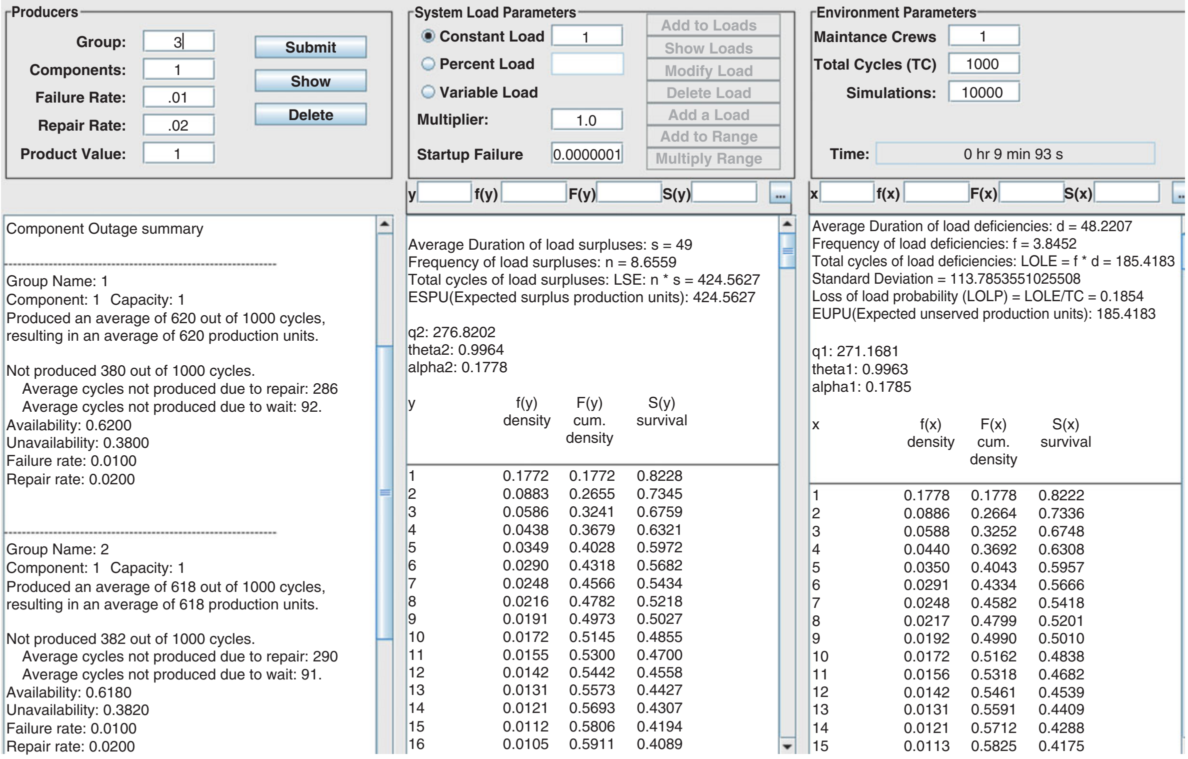


FIGURE 4 | Simulated outage history for a simple two component independent-additive system.



Producers

Group: 3
Components: 1
Failure Rate: .01
Repair Rate: .02
Product Value: 1

System Load Parameters

● Constant Load 1
○ Percent Load
○ Variable Load
Multiplier: 1.0
Startup Failure 0.0000001

Environment Parameters

Maintance Crews 1
Total Cycles (TC) 1000
Simulations: 10000
Time: 0 hr 9 min 93 s

y f(y) F(y) S(y)

x f(x) F(x) S(x)

```
Component Outage summary

--------------------------------------------------------
Group Name: 1
Component: 1  Capacity: 1
Produced an average of 620 out of 1000 cycles,
resulting in an average of 620 production units.

Not produced 380 out of 1000 cycles.
   Average cycles not produced due to repair: 286
   Average cycles not produced due to wait: 92.
Availability: 0.6200
Unavailability: 0.3800
Failure rate: 0.0100
Repair rate: 0.0200


--------------------------------------------------------
Group Name: 2
Component: 1  Capacity: 1
Produced an average of 618 out of 1000 cycles,
resulting in an average of 618 production units.

Not produced 382 out of 1000 cycles.
   Average cycles not produced due to repair: 290
   Average cycles not produced due to wait: 91.
Availability: 0.6180
Unavailability: 0.3820
Failure rate: 0.0100
Repair rate: 0.0200
```

```
Average Duration of load surpluses: s = 49
Frequency of load surpluses: n = 8.6559
Total cycles of load surpluses: LSE: n * s = 424.5627
ESPU(Expected surplus production units): 424.5627

q2: 276.8202
theta2: 0.9964
alpha2: 0.1778

y            f(y)      F(y)      S(y)
             density   cum.      survival
                       density
-------------------------------------------
1            0.1772    0.1772    0.8228
2            0.0883    0.2655    0.7345
3            0.0586    0.3241    0.6759
4            0.0438    0.3679    0.6321
5            0.0349    0.4028    0.5972
6            0.0290    0.4318    0.5682
7            0.0248    0.4566    0.5434
8            0.0216    0.4782    0.5218
9            0.0191    0.4973    0.5027
10           0.0172    0.5145    0.4855
11           0.0155    0.5300    0.4700
12           0.0142    0.5442    0.4558
13           0.0131    0.5573    0.4427
14           0.0121    0.5693    0.4307
15           0.0112    0.5806    0.4194
16           0.0105    0.5911    0.4089
```

```
Average Duration of load deficiencies: d = 48.2207
Frequency of load deficiencies: f = 3.8452
Total cycles of load deficiencies: LOLE = f * d = 185.4183
Standard Deviation = 113.7853551025508
Loss of load probability (LOLP) = LOLE/TC = 0.1854
EUPU(Expected unserved production units): 185.4183

q1: 271.1681
theta1: 0.9963
alpha1: 0.1785

x            f(x)      F(x)      S(x)
             density   cum.      survival
                       density
-------------------------------------------
1            0.1778    0.1778    0.8222
2            0.0886    0.2664    0.7336
3            0.0588    0.3252    0.6748
4            0.0440    0.3692    0.6308
5            0.0350    0.4043    0.5957
6            0.0291    0.4334    0.5666
7            0.0248    0.4582    0.5418
8            0.0217    0.4799    0.5201
9            0.0192    0.4990    0.5010
10           0.0172    0.5162    0.4838
11           0.0156    0.5318    0.4682
12           0.0142    0.5461    0.4539
13           0.0131    0.5591    0.4409
14           0.0121    0.5712    0.4288
15           0.0113    0.5825    0.4175
```

FIGURE 5 | Small cyber CLOUD same as Figure 3 with only one repair crew instead of two crews.

space limitation, only the pdf for the LoSE, which is NBD, can be seen plotted in Figure 12 derived from results tabulated in Figure 11. Otherwise, the CLOURA software tool can accommodate Weibull distributed input data besides the default negative exponential assumption, and also incorporate dependence between production units as in Figure 9 by using λ_0 for failure rates, and μ_0 for repair rates valued other than quasizero.[42] The tool can accommodate start-up failure probability and start-up delay for the units as

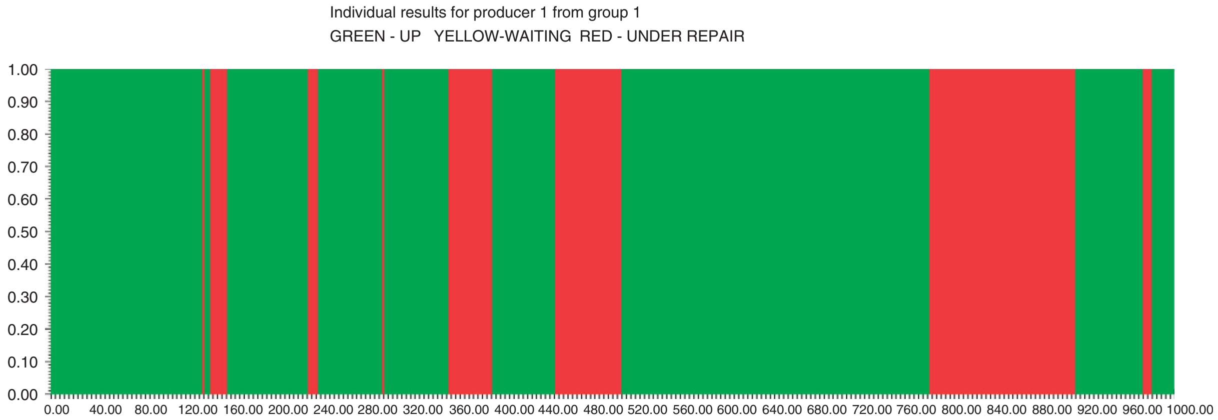


FIGURE 6 | For Component 1, up (green) and down (red) and no wait times (yellow) from Figure 3.

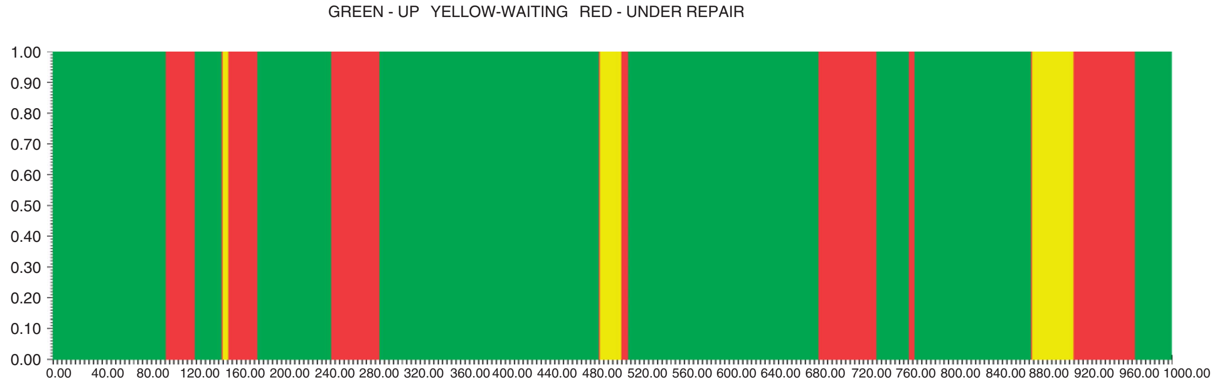


FIGURE 7 | For Component 2, up (green) and down (red) with wait times (yellow) from Figure 5.

observed in Figures 11, 17, and 18 where the default assumes quasi null values.[10,17,43,42,25]

VARIOUS APPLICATIONS TO SMALL AND LARGE CYBER CLOUDS

In terms of cybersystems scalability, consider two different scales. A small-scale is for example, one experimental cyber CLOUD system with only two groups and a total of two units (one unit per group). A large scale example can be three large cyber systems composed of (1) 103 groups with a total of 443 units, (2) 24 groups with a total of 348 units, (3) 61 groups with a total of 398 units. The small-scale experiment is presented first to clarify the theory behind the large systems.

Small Cyber CLOUD Systems with Experimental, Markov, and DES Solutions

Let us assume that there are two groups of components in a cyber system[3,4] each of which has 1 GB computing capacity. See Figures 3–7 with two scenarios:

1. One with both components having a repair crew each, as in Figures 3 and 6, or
2. Solely one repair crew for both, where there now will exist unfavorable 'wait' times in addition to repair times due to crew unavailability as in Figures 5 and 7.

The mathematical and statistical analyses are as follow to support the statistical simulations:

1. In Figure 3, we study 1000 cycles (or hours) simulating 10,000 times: Using Newton's nonlinear root finding solution algorithm, for the resulting average duration of service (load) $d = 25.24$ cycles with the solution $q = 122$. We can calculate the pdf values of average duration for 1, 2, 3, ..., n, at will and can have it plotted by the CLOURA software tool entering $M = 25.24$ and $q = 122.3$. However, the plot of d is not included due to space limitations.
2. In Figure 5, the same as in (1) is done when there is only one repair crew for two components, therefore this new decision leading to

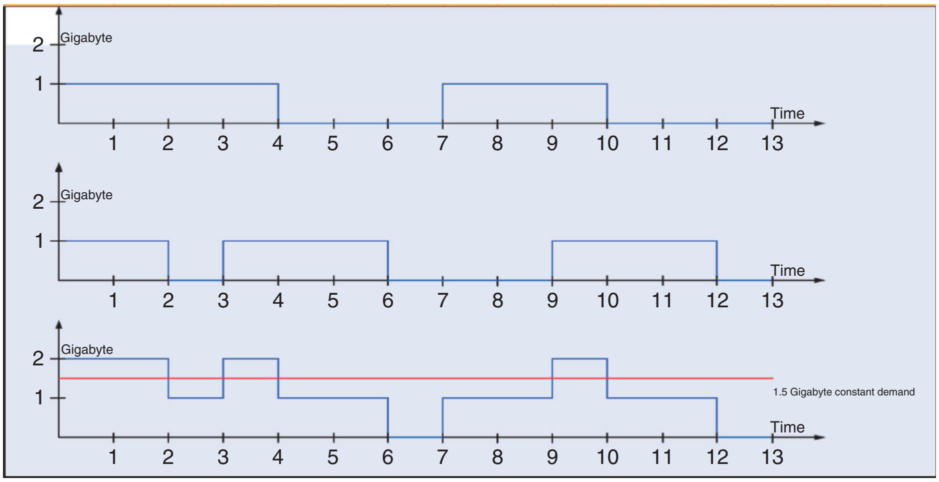


FIGURE 8 | Eqs. (17)–(19) plotted versus time for units 1, 2 and their sum with 1.5 GB = constant demand.

'wait times' for a less efficient system. Therefore, $d = 48.2$ cycles almost doubled and $q = 271$.

For our small experimental cyber system model (see Figure 8), we consider a basic operational case identical to which is shown in Eqs. (17)–(19). In this figure, we represent, the operation of each unit (Units 1 and 2 for Capacity versus Time), during the entire period of the study, which is 13 hours or cycles. We represent the operation of the entire system via the addition of the two 1 GB units, 1 and 2.

Note, $U_3(t) = U_1(t) + U_2(t), 0 \le t \le 13$ as follow in Eqs. (17)–(19).

$$U_1(t) = 1, 0 \le t < 4 \text{ and } 7 < t \le 10$$
$$= 0, \text{ elsewhere} \qquad (17)$$
$$U_2(t) = 1, 0 \le t < 2; 3 \le t < 6 \text{ and } 9 \le t < 12$$
$$= 0, \text{ elsewhere} \qquad (18)$$
$$U_3(t) = 2, 0 \le t < 2; 3 \le t < 4 \text{ and } 9 \le t < 10$$
$$= 1, 2 \le t < 3; 5 \le t < 6; 7 \le t < 9 \text{ and}$$
$$10 \le t < 12 = 0, \text{ elsewhere.} \qquad (19)$$

In this sample graph of Figure 8, unit 1 is available during the initial 4 h servicing at its entire capacity of 1 GB. However at the end of the fourth hour a disruption occurred, which caused unit 1 to be down. It takes 3 h for unit 1 to recover and it fails after 3 h. Unit 2 operates other than between second and third, and between sixth and ninth hours. So, we can see when each unit (U_1 and U_2), the entire cyber system (U_3) is up (available) or down (nonavailable) in an independent setting. This model is a form of oversimplified CLOUD environment with additive resources. It is a conglomerate of computers (units) distributed in different locations but working or servicing in parallel (simultaneously) to an external clientele of customers. This cluster of computers is arranged in groups, comprising sets of computers with similar specifications. The parameters involved in this simple experimental model are given as follow:

- Disruption rate caused by security breaches: γ
- Mean time to disruption: $m = 1/\gamma$
- Recovery rate: μ. The frequency with which a system recovers from a security breach.
- Mean recovery time: $r = 1/\mu$.
- Availability: $P = \mu/(\mu + \gamma)$
- Unit capacity: C. It refers to maximum unit generation or production capacity
- Demand of service: D. Total service demand for the entire system.

An initial study is designed for a constant service (demand), but a more realistic variable demand will be considered for larger-scale systems. Here unit capacity and service (demand) are measured in terms of computing cycles (e.g., flops), Gigabytes (GB) storage.

Unit 1 Availability:

- $P_1 = 7/13 = 0.538461538 = \mu_1/(\mu_1 + \gamma_1)$
- Mean time to disruption: average servicing hours before disruption, $m_1 = 7/2 = 3.5 = 1/\gamma_1$
- Disruption rate: $\gamma_1 = 1/3.5 = 0.285714286$
- Mean recovery time: $\gamma_1 = 6/2 = 3.0 = 1/\mu_1$
- Recovery rate: $\mu_1 = 1/3.0 = 0.333333333$.

Unit 2 Availability:

- $P_2 = 8/13 = 0.615384615 = \gamma_2/(\mu_2 + \gamma_2)$
- Mean time to disruption: average servicing hours before disruption, $m_2 = 8/3 = 2.66 = 1/\gamma_2$
- Disruption rate: $\gamma_2 = 1/2.66666667 = 0.375$
- Mean recovery time: $\gamma_2 = 5/3 = 1.66666667 = 1/\mu_2$
- Recovery rate: $\mu_2 = 1/1.66666667 = 0.6$.

The service demand is assumed to be constant and equal to 1.5 GB. The result for the small-scale experimental system availability from Figure 8 is that the probability of $U_3(\mathrm{t}) = 2$ (i.e., larger than 1.5 GB) is $4/13 = \mathbf{0.307}$. Markov Chain's exact result by substituting the input values above into Eq. (20) as follows in Figure 9 is $P_1 = (0.3333)(0.6)/[(0.2857 + 0.3333)(0.3750 + 0.6)] = \mathbf{0.331}$. DES's (Discrete Event Simulation) LoSE is **0.303** after 100,000 simulation runs (each simulation run = 13 hours or cycles) also as in Figure 9. With 1,000,000 simulation runs, the DES result converges to **0.305**. All three solutions (experimental, simulation, Markov) with different methods are satisfactorily close. The experimental case denotes only a single realization of the DES because one can realize the simulation input data in many more than a single scenario among which realization of Eqs. (17)–(19) is only one of them. This comparison duly demonstrates the validity of the discrete event simulation of additive units under a constant or varying service demand, a feature which will prove very useful. DES software will mimic and predict the CLOUD operations before actually experiencing what maybe the costly real life occurrences and their consequences.

Large Cyber CLOUD Systems with Markov and DES Solutions

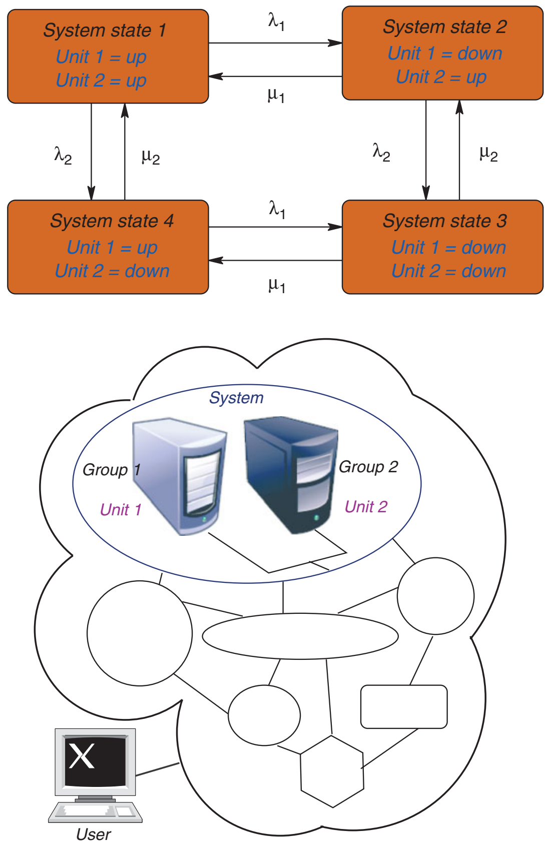
(1) Assume now that there is a very large system to examine like those in Refs 25,42–44 such as an interconnected cyber CLOUD in a server farm with 103 production groups, each of which has a given number of components, totaling to 443 servers and each with its own distinct repair crew. Total installed capacity is 26,237 GB. Java coded CLOURA simulates the cyber CLOUD for 10,000 runs covering 8760 cycles (or hours). The service demand (as a variable load) for an annual period of 8760 h is available, and given in Figure 10 in Megawatts (power) or Gigabytes (cyber).

As Figure 11 shows, at the end of 43 min of computation time for 10,000 simulation runs (one simulation run = one calendar year), CLOURA computes the average duration of loss of service (or load) as 3.12 cycles (or hours) with a frequency of $f = 384$ times. Hence, $f \times d = 3.12 \times 384 = 1202$ cycles of loss of service with a LoS probability of $1202/8760 = 0.137$. Expected Unserved Production, or EUP units (in terms of gigabyte cycles) was 2,060,459. In another what-if study, one can choose less than a perfect number (=443) of crews such as one half (=222) to see what an adverse effect it plays causing undesirable wait times. This leverages the CLOUD managers to judge the extent of crews so as to mitigate the risk to a tolerable value. Experimental model solution is infeasible and impossible for large systems.

The pdf (exact probability), cdf (cumulative probability), and sdf (survival probability) columns are given on the right-hand side in Figure 11 where any of the statistical plots can be extracted at will to facilitate statistical inference on the average duration. Since the total LoS duration is the average duration multiplied by the frequency. The LSD's pdf summed at every interruption of loss of service is given by Eq. (3). These individual probabilities add up to form the total LoSP (Loss of Service Probability) in deriving a closed form distribution for statistical inference. The total hours or cycles of expected LoS is a Compound Poisson, that is, Negative Binomial. Then LoSE is then distributed with respect to a Negative Binomial, NBD ($M = 1202$, $q = 7.16$) or Poisson ^ LSD whose distribution functions can be obtained from the NBD software shown in Figures 12 and 13. Note, for the pdf plot of average duration d, simply use the same $q = 7.16$ this time with $M = 3.12$ in Figure 11 filling the blanks for NB parameters on the upper right hand corner.

The expected surplus production (for units above the zero margin threshold) was calculated to be 35,343,953 gigabyte cycles (or hours). In Figure 14, those red (load) and blue (supply) combinations are shown. The deficiency occurs when the service demand (load) is higher than the production supply. Figure 15 shows 'Reserve Margin = Installed Capacity-Load-Outage' at any given cycle. Figure 16 illustrates any particular unit's fluctuations of up and down states in an entire year of operations, where lower than the marginal zero indicating red line indicates deficient hours of load to contribute to loss of load hours.

The feature in Figure 11 for Figure 10 has flexibilities to add a desired amount to all loads, or modify a certain selected load or delete a selected load or add single selected load as well as adding a desirable amount to a range of selected loads like from 8000th to 8760th or scaling a range like multiplying by a constant of 3/2 or 1/2. These are necessary for load (service) maintenance or shedding or load appending.

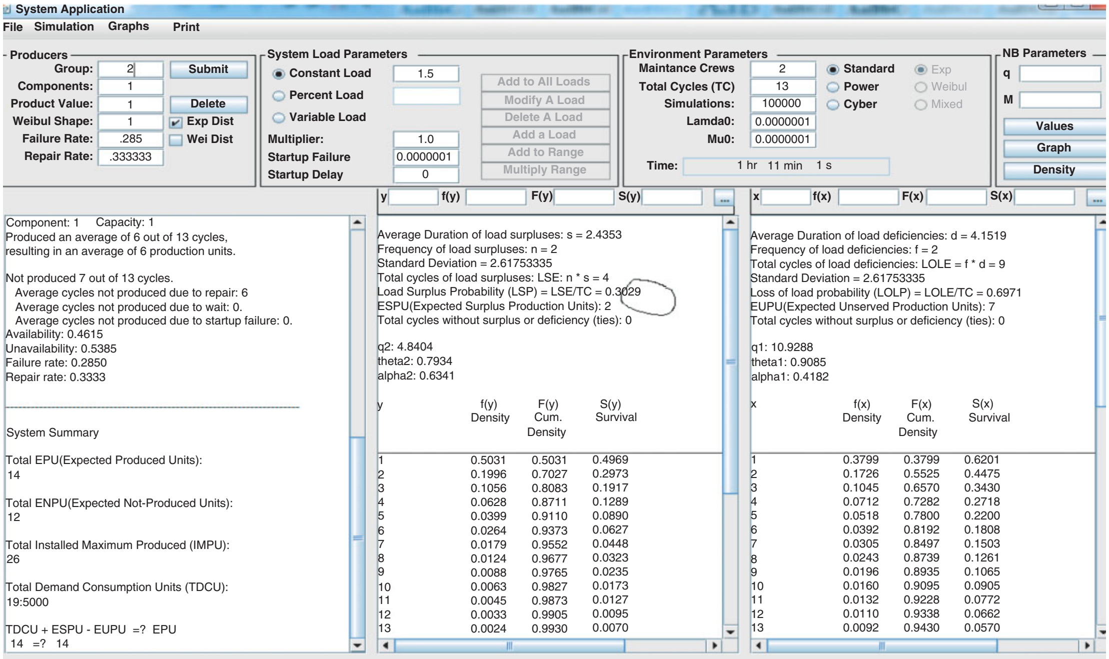


$$P_1 = \mu_1\,\mu_2/(\lambda_1 + \mu_1)(\lambda_2 + \mu_2) \tag{20}$$

$$P_2 = \lambda_1\,\mu_2/(\lambda_1 + \mu_1)(\lambda_2 + \mu_2) \tag{21}$$

$$P_3 = \lambda_2\,\mu_1/(\lambda_1 + \mu_1)(\lambda_2 + \mu_2) \tag{22}$$

$$P_4 = \lambda_1\,\lambda_2/(\lambda_1 + \mu_1)(\lambda_2 + \mu_2) \tag{23}$$



Producers

Group:	2
Components:	1
Product Value:	1
Weibul Shape:	1
Failure Rate:	.285
Repair Rate:	.333333

☑ Exp Dist
□ Wei Dist

System Load Parameters

● Constant Load 1.5
○ Percent Load
○ Variable Load

Multiplier:	1.0
Startup Failure	0.0000001
Startup Delay	0

Environment Parameters

Maintance Crews	2
Total Cycles (TC)	13
Simulations:	100000
Lamda0:	0.0000001
Mu0:	0.0000001

● Standard ○ Power ○ Cyber

● Exp ○ Weibul ○ Mixed

Time: 1 hr 11 min 1 s

NB Parameters

q

M

```
Component: 1     Capacity: 1
Produced an average of 6 out of 13 cycles,
resulting in an average of 6 production units.

Not produced 7 out of 13 cycles.
  Average cycles not produced due to repair: 6
  Average cycles not produced due to wait: 0.
  Average cycles not produced due to startup failure: 0.
Availability: 0.4615
Unavailability: 0.5385
Failure rate: 0.2850
Repair rate: 0.3333

-----------------------------------------------------------

System Summary

Total EPU(Expected Produced Units):
14

Total ENPU(Expected Not-Produced Units):
12

Total Installed Maximum Produced (IMPU):
26

Total Demand Consumption Units (TDCU):
19:5000

TDCU + ESPU - EUPU  =?  EPU
 14   =?   14
```

y f(y) F(y) S(y)

```
Average Duration of load surpluses: s = 2.4353
Frequency of load surpluses: n = 2
Standard Deviation = 2.61753335
Total cycles of load surpluses: LSE: n * s = 4
Load Surplus Probability (LSP) = LSE/TC = 0.3029
ESPU(Expected Surplus Production Units): 2
Total cycles without surplus or deficiency (ties): 0

q2: 4.8404
theta2: 0.7934
alpha2: 0.6341
```

y	f(y) Density	F(y) Cum. Density	S(y) Survival
1	0.5031	0.5031	0.4969
2	0.1996	0.7027	0.2973
3	0.1056	0.8083	0.1917
4	0.0628	0.8711	0.1289
5	0.0399	0.9110	0.0890
6	0.0264	0.9373	0.0627
7	0.0179	0.9552	0.0448
8	0.0124	0.9677	0.0323
9	0.0088	0.9765	0.0235
10	0.0063	0.9827	0.0173
11	0.0045	0.9873	0.0127
12	0.0033	0.9905	0.0095
13	0.0024	0.9930	0.0070

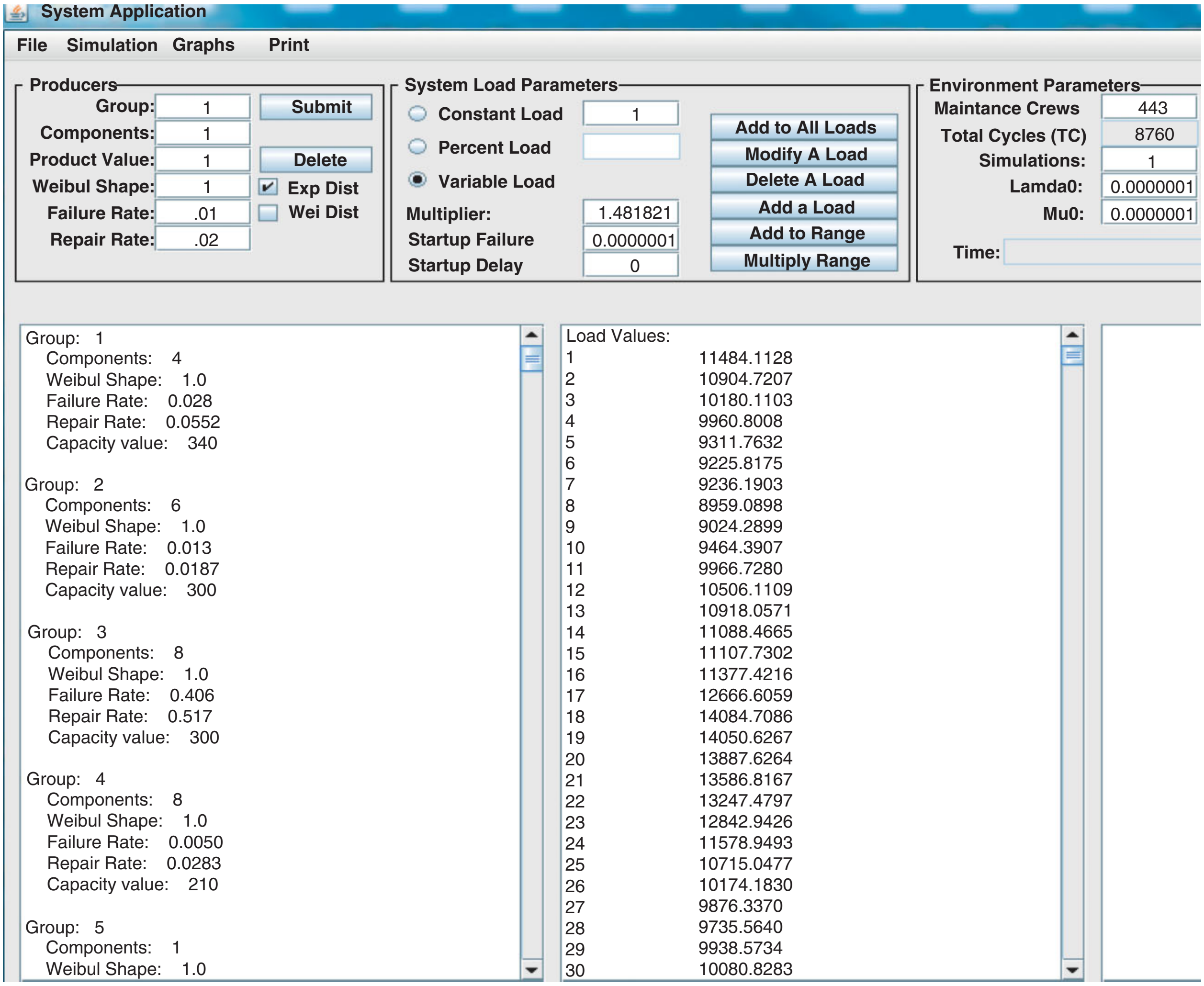
x f(x) F(x) S(x)

```
Average Duration of load deficiencies: d = 4.1519
Frequency of load deficiencies: f = 2
Total cycles of load deficiencies: LOLE = f * d = 9
Standard Deviation = 2.61753335
Loss of load probability (LOLP) = LOLE/TC = 0.6971
EUPU(Expected Unserved Production Units): 7
Total cycles without surplus or deficiency (ties): 0

q1: 10.9288
theta1: 0.9085
alpha1: 0.4182
```

x	f(x) Density	F(x) Cum. Density	S(x) Survival
1	0.3799	0.3799	0.6201
2	0.1726	0.5525	0.4475
3	0.1045	0.6570	0.3430
4	0.0712	0.7282	0.2718
5	0.0518	0.7800	0.2200
6	0.0392	0.8192	0.1808
7	0.0305	0.8497	0.1503
8	0.0243	0.8739	0.1261
9	0.0196	0.8935	0.1065
10	0.0160	0.9095	0.0905
11	0.0132	0.9228	0.0772
12	0.0110	0.9338	0.0662
13	0.0092	0.9430	0.0570

FIGURE 9 | Markov states, small cyber CLOUD, markov state space equations, DES solutions.



Producers
Group: 1
Components: 1
Product Value: 1
Weibul Shape: 1
Failure Rate: .01
Repair Rate: .02
☑ Exp Dist
□ Wei Dist

System Load Parameters
○ Constant Load 1
○ Percent Load
● Variable Load
Multiplier: 1.481821
Startup Failure 0.0000001
Startup Delay 0

Environment Parameters
Maintance Crews 443
Total Cycles (TC) 8760
Simulations: 1
Lamda0: 0.0000001
Mu0: 0.0000001
Time:

```
Group:  1
   Components:   4
   Weibul Shape:   1.0
   Failure Rate:   0.028
   Repair Rate:   0.0552
   Capacity value:   340

Group:  2
   Components:   6
   Weibul Shape:   1.0
   Failure Rate:   0.013
   Repair Rate:   0.0187
   Capacity value:   300

Group:  3
   Components:   8
   Weibul Shape:   1.0
   Failure Rate:   0.406
   Repair Rate:   0.517
   Capacity value:   300

Group:  4
   Components:   8
   Weibul Shape:   1.0
   Failure Rate:   0.0050
   Repair Rate:   0.0283
   Capacity value:   210

Group:  5
   Components:   1
   Weibul Shape:   1.0
```

```
Load Values:
1          11484.1128
2          10904.7207
3          10180.1103
4          9960.8008
5          9311.7632
6          9225.8175
7          9236.1903
8          8959.0898
9          9024.2899
10         9464.3907
11         9966.7280
12         10506.1109
13         10918.0571
14         11088.4665
15         11107.7302
16         11377.4216
17         12666.6059
18         14084.7086
19         14050.6267
20         13887.6264
21         13586.8167
22         13247.4797
23         12842.9426
24         11578.9493
25         10715.0477
26         10174.1830
27         9876.3370
28         9735.5640
29         9938.5734
30         10080.8283
```

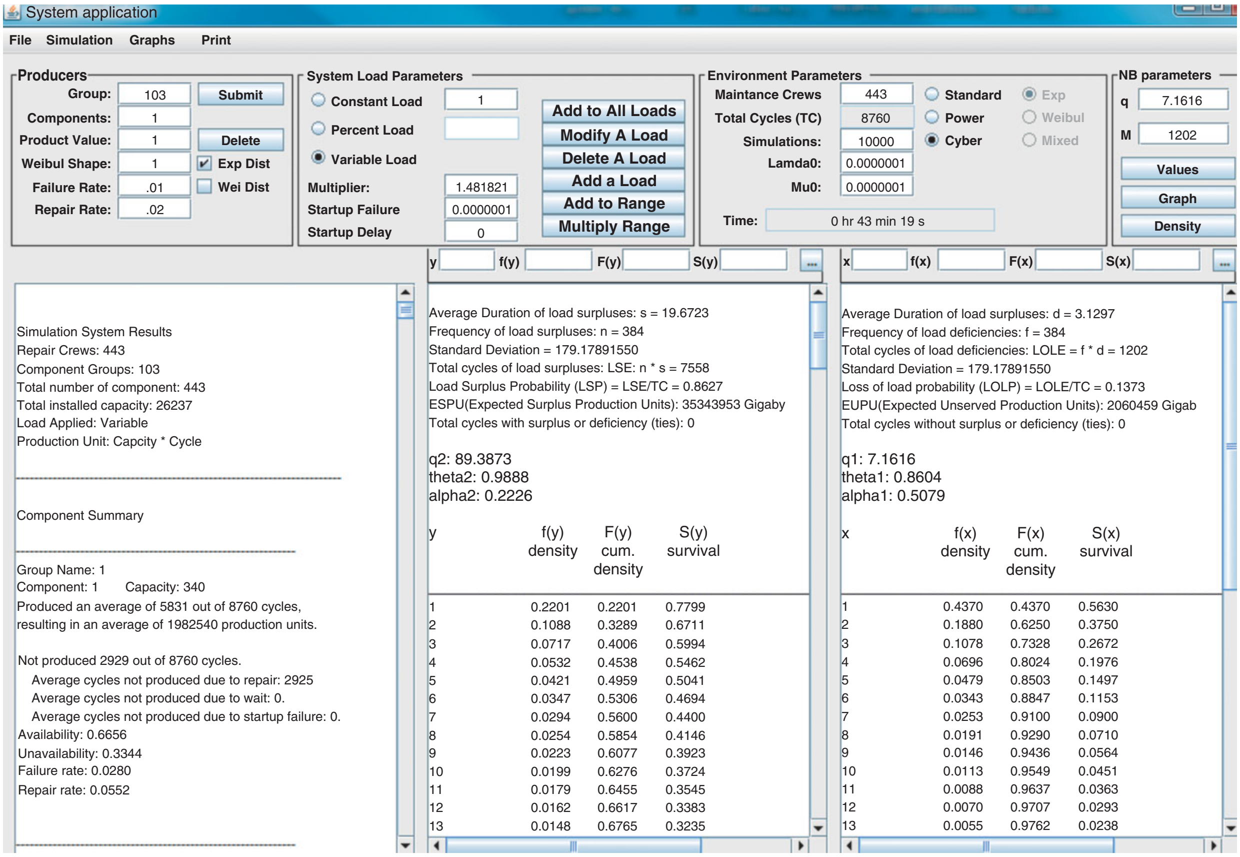
FIGURE 10 | An example of hourly load (values) demand from 1st to 8760th hour in a calendar year.

Figure 10 shows for the above cited example (1) data values from 1st to 8760th hour for 103 production groups, each of which has a given number of components, totaling to 443 servers.

Surplus and deficiency productions (gigabyte cycles) are not additive, and not related other than the fact that if one increases, the other decreases. Figure 16 depicts the individual behavior of a producing unit among many to constitute the entire CLOUD. In the middle column of Figure 11, this time, surpluses instead of deficiencies are studied to estimate availability instead of unavailability. That is the average of surpluses, $s = 20$ cycles recurring 384 times totaling to 7558 cycles = 8760 (annual total)—1202(deficiencies). Figures 12 and 13 illustrate the plotting of NBD ($M = 1202$, $q = 7.16$) using the earlier research by the author[21,22,25,2] on NBD.

(2) The next large cyber system example to be examined comprises 24 groups of units totaling to 348 units in Figure 17. Therefore, the total number of Markov system's up and down states are $2^{348} = 5.73374654 \times 10^{104}$, which is indescribably enormous. Respectively, the failure (disruption) rate λ_k, repair (recovery) rate μ_k, and production capacity (storage or generation) C_k for each of kth unit are supplied by the analyst. The service demand (load) cycle is varying from hour to hour for the entire year (8760 h) of operations. Experimentally it is very tedious and not practical, if not infeasible, to add the existing unit availabilities hour by hour for 348 units, a process which will take years. This is why discrete event simulation (DES) techniques have to be used to obtain large-scale solutions, which are intractably lengthy using theoretical Markov solutions.

The result by using the Markov chains similar to the small-scale system (Figure 9) is the steady



Producers
Group: 103
Components: 1
Product Value: 1
Weibul Shape: 1
Failure Rate: .01
Repair Rate: .02
☑ Exp Dist
□ Wei Dist

System Load Parameters
○ Constant Load 1
○ Percent Load
● Variable Load
Multiplier: 1.481821
Startup Failure 0.0000001
Startup Delay 0

Environment Parameters
Maintance Crews 443
Total Cycles (TC) 8760
Simulations: 10000
Lamda0: 0.0000001
Mu0: 0.0000001
Time: 0 hr 43 min 19 s
○ Standard
○ Power
● Cyber
● Exp
○ Weibul
○ Mixed

NB parameters
q 7.1616
M 1202

y f(y) F(y) S(y)
x f(x) F(x) S(x)

Simulation System Results
Repair Crews: 443
Component Groups: 103
Total number of component: 443
Total installed capacity: 26237
Load Applied: Variable
Production Unit: Capcity * Cycle

Component Summary

Group Name: 1
Component: 1 Capacity: 340
Produced an average of 5831 out of 8760 cycles,
resulting in an average of 1982540 production units.

Not produced 2929 out of 8760 cycles.
Average cycles not produced due to repair: 2925
Average cycles not produced due to wait: 0.
Average cycles not produced due to startup failure: 0.
Availability: 0.6656
Unavailability: 0.3344
Failure rate: 0.0280
Repair rate: 0.0552

Average Duration of load surpluses: s = 19.6723
Frequency of load surpluses: n = 384
Standard Deviation = 179.17891550
Total cycles of load surpluses: LSE: n * s = 7558
Load Surplus Probability (LSP) = LSE/TC = 0.8627
ESPU(Expected Surplus Production Units): 35343953 Gigaby
Total cycles with surplus or deficiency (ties): 0

q2: 89.3873
theta2: 0.9888
alpha2: 0.2226

y	f(y) density	F(y) cum. density	S(y) survival
1	0.2201	0.2201	0.7799
2	0.1088	0.3289	0.6711
3	0.0717	0.4006	0.5994
4	0.0532	0.4538	0.5462
5	0.0421	0.4959	0.5041
6	0.0347	0.5306	0.4694
7	0.0294	0.5600	0.4400
8	0.0254	0.5854	0.4146
9	0.0223	0.6077	0.3923
10	0.0199	0.6276	0.3724
11	0.0179	0.6455	0.3545
12	0.0162	0.6617	0.3383
13	0.0148	0.6765	0.3235

Average Duration of load surpluses: d = 3.1297
Frequency of load deficiencies: f = 384
Total cycles of load deficiencies: LOLE = f * d = 1202
Standard Deviation = 179.17891550
Loss of load probability (LOLP) = LOLE/TC = 0.1373
EUPU(Expected Unserved Production Units): 2060459 Gigab
Total cycles without surplus or deficiency (ties): 0

q1: 7.1616
theta1: 0.8604
alpha1: 0.5079

x	f(x) density	F(x) cum. density	S(x) survival
1	0.4370	0.4370	0.5630
2	0.1880	0.6250	0.3750
3	0.1078	0.7328	0.2672
4	0.0696	0.8024	0.1976
5	0.0479	0.8503	0.1497
6	0.0343	0.8847	0.1153
7	0.0253	0.9100	0.0900
8	0.0191	0.9290	0.0710
9	0.0146	0.9436	0.0564
10	0.0113	0.9549	0.0451
11	0.0088	0.9637	0.0363
12	0.0070	0.9707	0.0293
13	0.0055	0.9762	0.0238

FIGURE 11 | Simulation results for a large cyber CLOUD with 443 components for large-scale example (1).

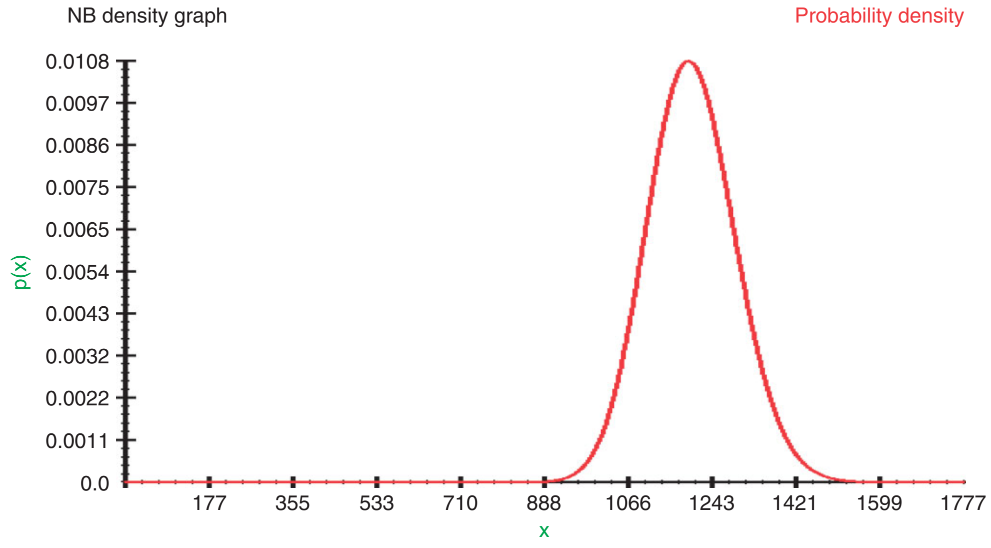
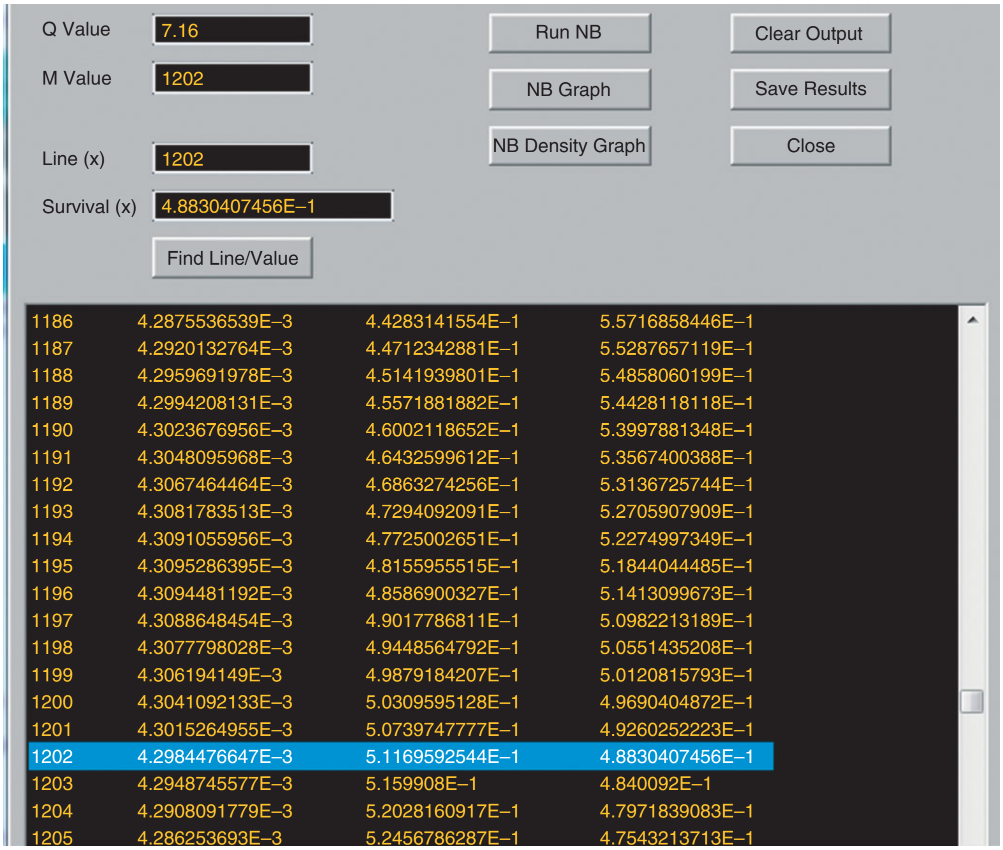


FIGURE 12 | The frequency distribution of the LoSE ($M = 1202$, $q = 7.16$) from Figures 11 and 13.

state probability of the operational units satisfying the demand. The experimental or manual solution for 348 units is impossible because it is infeasible and intractable. That is exactly where a DES is necessary to mimic a quasi-life scenario of CLOUD operations.

The Markov Chain study using an Alabama supercomputer gave a solution of QoSE = 1 − LoSE = 1 − 0.0191 = **0.9809** for the entire annual operation, whereas the discrete event simulation resulted in QoSE = 1 − 0.0546 = **0.9454** (see LSP in



Q Value 7.16

M Value 1202

Line (x) 1202

Survival (x) 4.8830407456E–1

```
1186   4.2875536539E–3   4.4283141554E–1   5.5716858446E–1
1187   4.2920132764E–3   4.4712342881E–1   5.5287657119E–1
1188   4.2959691978E–3   4.5141939801E–1   5.4858060199E–1
1189   4.2994208131E–3   4.5571881882E–1   5.4428118118E–1
1190   4.3023676956E–3   4.6002118652E–1   5.3997881348E–1
1191   4.3048095968E–3   4.6432599612E–1   5.3567400388E–1
1192   4.3067464464E–3   4.6863274256E–1   5.3136725744E–1
1193   4.3081783513E–3   4.7294092091E–1   5.2705907909E–1
1194   4.3091055956E–3   4.7725002651E–1   5.2274997349E–1
1195   4.3095286395E–3   4.8155955515E–1   5.1844044485E–1
1196   4.3094481192E–3   4.8586900327E–1   5.1413099673E–1
1197   4.3088648454E–3   4.9017786811E–1   5.0982213189E–1
1198   4.3077798028E–3   4.9448564792E–1   5.0551435208E–1
1199   4.306194149E–3    4.9879184207E–1   5.0120815793E–1
1200   4.3041092133E–3   5.0309595128E–1   4.9690404872E–1
1201   4.3015264955E–3   5.0739747777E–1   4.9260252223E–1
1202   4.2984476647E–3   5.1169592544E–1   4.8830407456E–1
1203   4.2948745577E–3   5.159908E–1       4.840092E–1
1204   4.2908091779E–3   5.2028160917E–1   4.7971839083E–1
1205   4.286253693E–3    5.2456786287E–1   4.7543213713E–1
```

FIGURE 13 | The print out shows $P(\text{LoSE} > 1202\ \text{h}) = 0.4883$ (slight right-skewed) as in Figure 13.

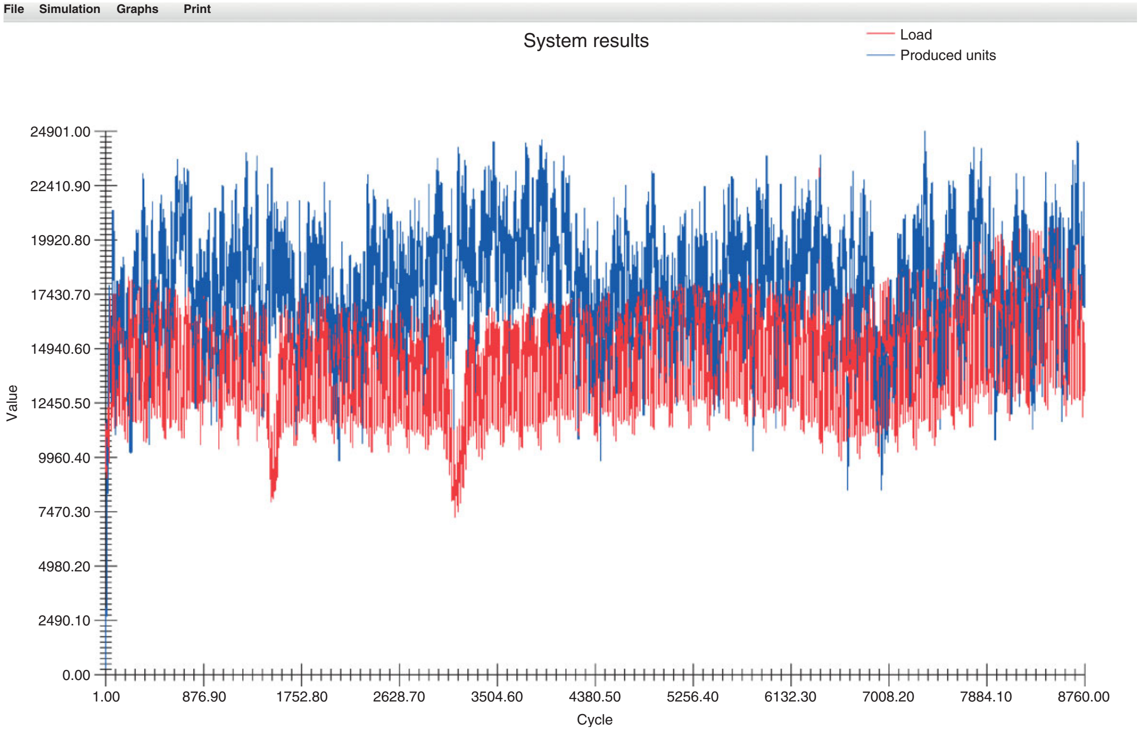


FIGURE 14 | Hourly available (blue colored available = Installed 26,237 GB − Outage Capacity due to hourly failures) and load (red colored customer demand) cycles over a year.

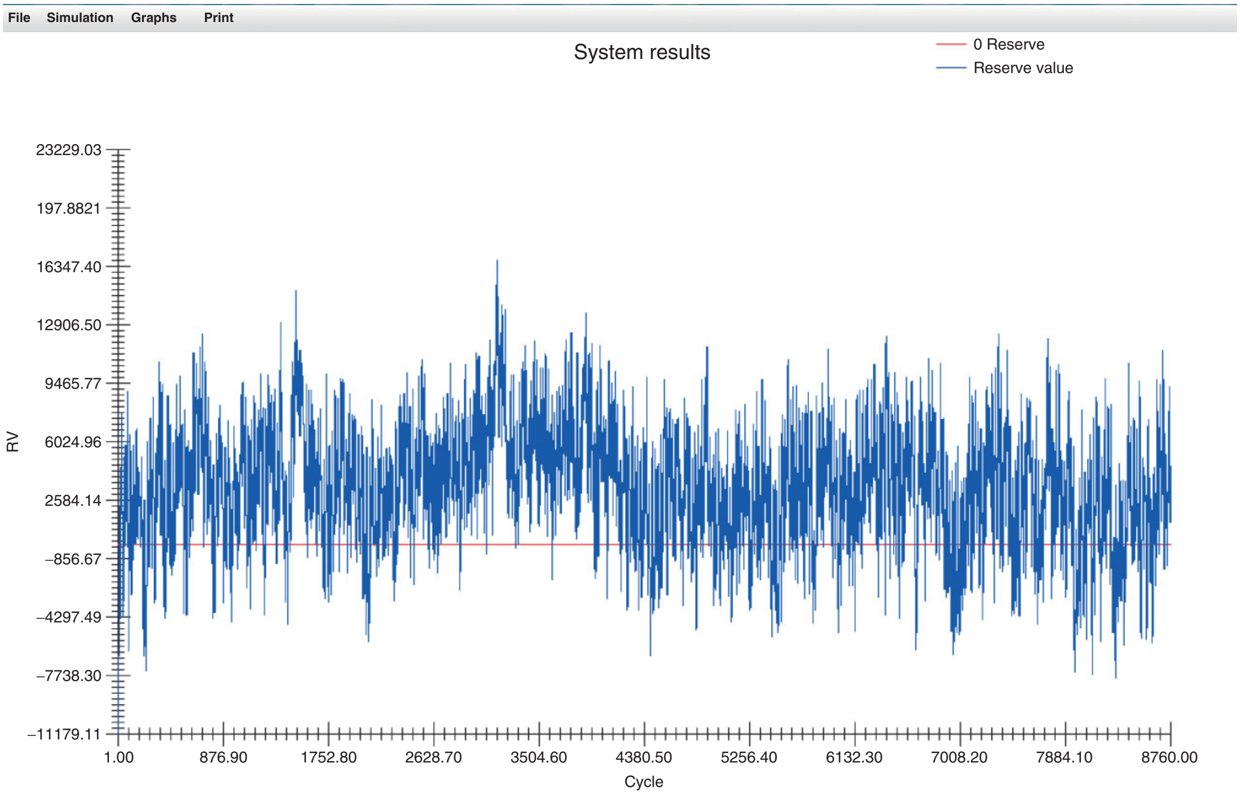


FIGURE 15 | Reserve Margin = Installed Capacity − Outages − Service Demand (load). Below zero implies Loss of Service (LoSE). Above zero (red) implies surplus service.

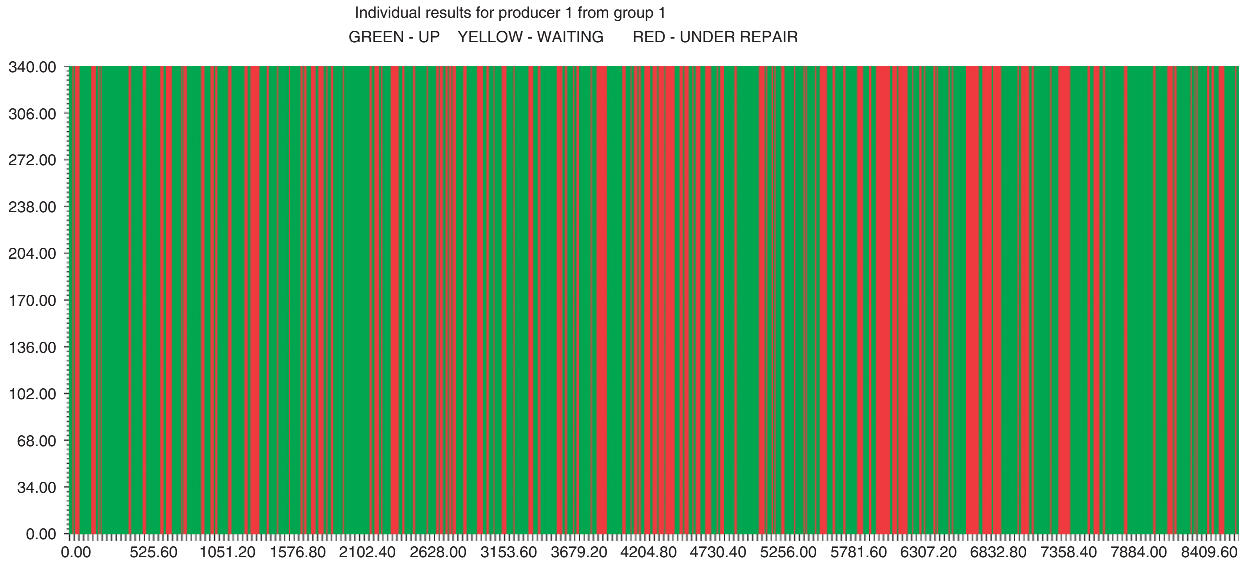


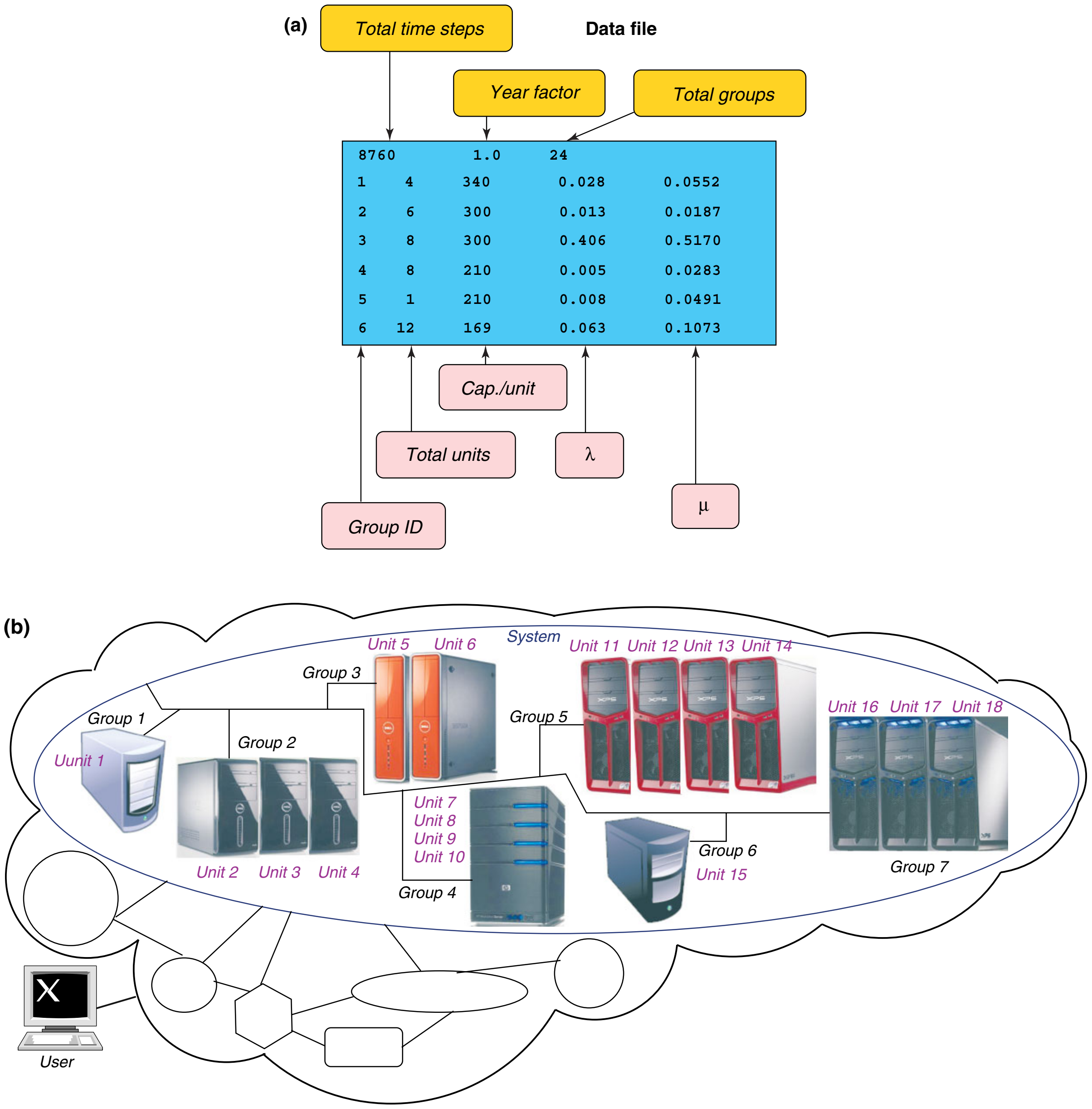
FIGURE 16 | Group 1's Component 1 up-down fluctuations in a year (8760 cycles) of operations.

Figure 17) which is satisfactorily comparable with a probabilistic difference of 0.0355 to the analytical Markov state space solution for 10,000 simulation runs (=10,000 calendar years) in 30 min of run time.

(3) For the Markov analysis of another large cyber system, as studied in Figure 18 in addition to the first large-scale example of Figure 11, was composed of 62 groups with a total 398 production units. For the Markov state solution, thus the total number of system states happens to be $2^{398} = 6.4556247 \times 10^{119}$ (very far away quadrillions), which is again a colossal value. The computing effort took about

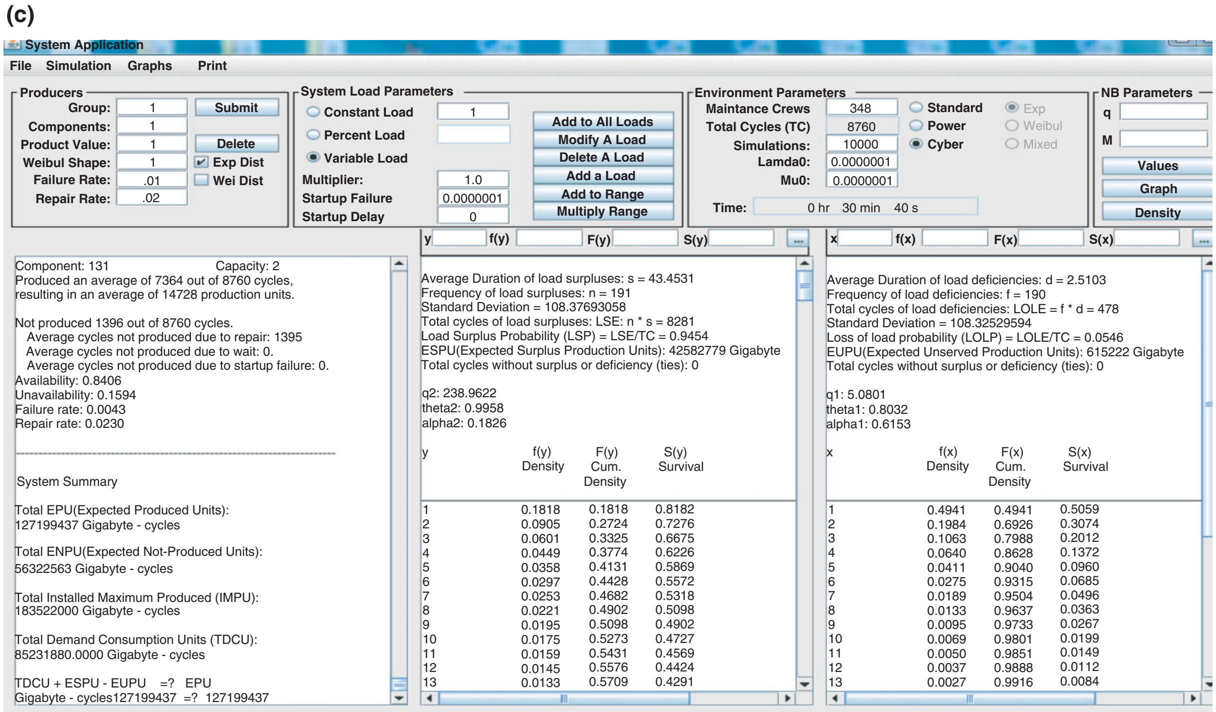
36 h with the Alabama Supercomputer Center; see Figure 19. Earlier when another CLOUD simulation was attempted with 443 production units of Figure 11, the supercomputing process crashed due to number of states exceeding 10^{133}. The goal was to calculate the CLOUD's steady state probability of being reliable at any given time under the data assumptions. The probability of likelihood is calculated for each steady state. Then, for each incremental time step, the probability of matching or exceeding the variable service demand is computed such as the one listed in Figure 10. Finally, the reliability is averaged by calculating the steady state probability over the entire period in dividing the number of favorable hours by the total number of exposure hours (=8760). The system service demand was a varying load cycle hour by hour for the entire 8760 cycles or hours. The result of the lengthy Markov Process solution is the steady state probability of the entirety of production units satisfying the system service demand.

Quality of Service Expected (QoSE) due to Markov Chains for the very large-scale CLOUD of 398 units resulted in 1 – LoSE = 1 – 0.061602 = **0.938398** (Figure 19) whereas due to Discrete Event



```
8760          1.0     24
1    4      340       0.028       0.0552
2    6      300       0.013       0.0187
3    8      300       0.406       0.5170
4    8      210       0.005       0.0283
5    1      210       0.008       0.0491
6   12      169       0.063       0.1073
```

FIGURE 17 | (a) Large cyber CLOUD data file for Markov state solution. (b) Large cyber CLOUD schema. (c) Digital event simulation results for large cyber CLOUD in large-scale example (2).



Producers
Group: 1
Components: 1
Product Value: 1
Weibul Shape: 1
Failure Rate: .01
Repair Rate: .02
☑ Exp Dist
□ Wei Dist

System Load Parameters
○ Constant Load 1
○ Percent Load
● Variable Load
Multiplier: 1.0
Startup Failure 0.0000001
Startup Delay 0

Environment Parameters
Maintance Crews 348
Total Cycles (TC) 8760
Simulations: 10000
Lamda0: 0.0000001
Mu0: 0.0000001
○ Standard
○ Power
● Cyber
● Exp
○ Weibul
○ Mixed
Time: 0 hr 30 min 40 s

NB Parameters
q
M

y f(y) F(y) S(y)

x f(x) F(x) S(x)

```
Component: 131                         Capacity: 2
Produced an average of 7364 out of 8760 cycles,
resulting in an average of 14728 production units.

Not produced 1396 out of 8760 cycles.
   Average cycles not produced due to repair: 1395
   Average cycles not produced due to wait: 0.
   Average cycles not produced due to startup failure: 0.
Availability: 0.8406
Unavailability: 0.1594
Failure rate: 0.0043
Repair rate: 0.0230

-----------------------------------------------------------------

 System Summary

Total EPU(Expected Produced Units):
127199437 Gigabyte - cycles

Total ENPU(Expected Not-Produced Units):
56322563 Gigabyte - cycles

Total Installed Maximum Produced (IMPU):
183522000 Gigabyte - cycles

Total Demand Consumption Units (TDCU):
85231880.0000 Gigabyte - cycles

TDCU + ESPU - EUPU    =?   EPU
Gigabyte - cycles127199437  =?  127199437
```

```
Average Duration of load surpluses: s = 43.4531
Frequency of load surpluses: n = 191
Standard Deviation = 108.37693058
Total cycles of load surpluses: LSE: n * s = 8281
Load Surplus Probability (LSP) = LSE/TC = 0.9454
ESPU(Expected Surplus Production Units): 42582779 Gigabyte
Total cycles without surplus or deficiency (ties): 0

q2: 238.9622
theta2: 0.9958
alpha2: 0.1826

y                f(y)       F(y)        S(y)
               Density      Cum.      Survival
                          Density

1              0.1818     0.1818     0.8182
2              0.0905     0.2724     0.7276
3              0.0601     0.3325     0.6675
4              0.0449     0.3774     0.6226
5              0.0358     0.4131     0.5869
6              0.0297     0.4428     0.5572
7              0.0253     0.4682     0.5318
8              0.0221     0.4902     0.5098
9              0.0195     0.5098     0.4902
10             0.0175     0.5273     0.4727
11             0.0159     0.5431     0.4569
12             0.0145     0.5576     0.4424
13             0.0133     0.5709     0.4291
```

```
Average Duration of load deficiencies: d = 2.5103
Frequency of load deficiencies: f = 190
Total cycles of load deficiencies: LOLE = f * d = 478
Standard Deviation = 108.32529594
Loss of load probability (LOLP) = LOLE/TC = 0.0546
EUPU(Expected Unserved Production Units): 615222 Gigabyte
Total cycles without surplus or deficiency (ties): 0

q1: 5.0801
theta1: 0.8032
alpha1: 0.6153

x                f(x)       F(x)        S(x)
               Density      Cum.      Survival
                          Density

1              0.4941     0.4941     0.5059
2              0.1984     0.6926     0.3074
3              0.1063     0.7988     0.2012
4              0.0640     0.8628     0.1372
5              0.0411     0.9040     0.0960
6              0.0275     0.9315     0.0685
7              0.0189     0.9504     0.0496
8              0.0133     0.9637     0.0363
9              0.0095     0.9733     0.0267
10             0.0069     0.9801     0.0199
11             0.0050     0.9851     0.0149
12             0.0037     0.9888     0.0112
13             0.0027     0.9916     0.0084
```

FIGURE 17 | Continued.

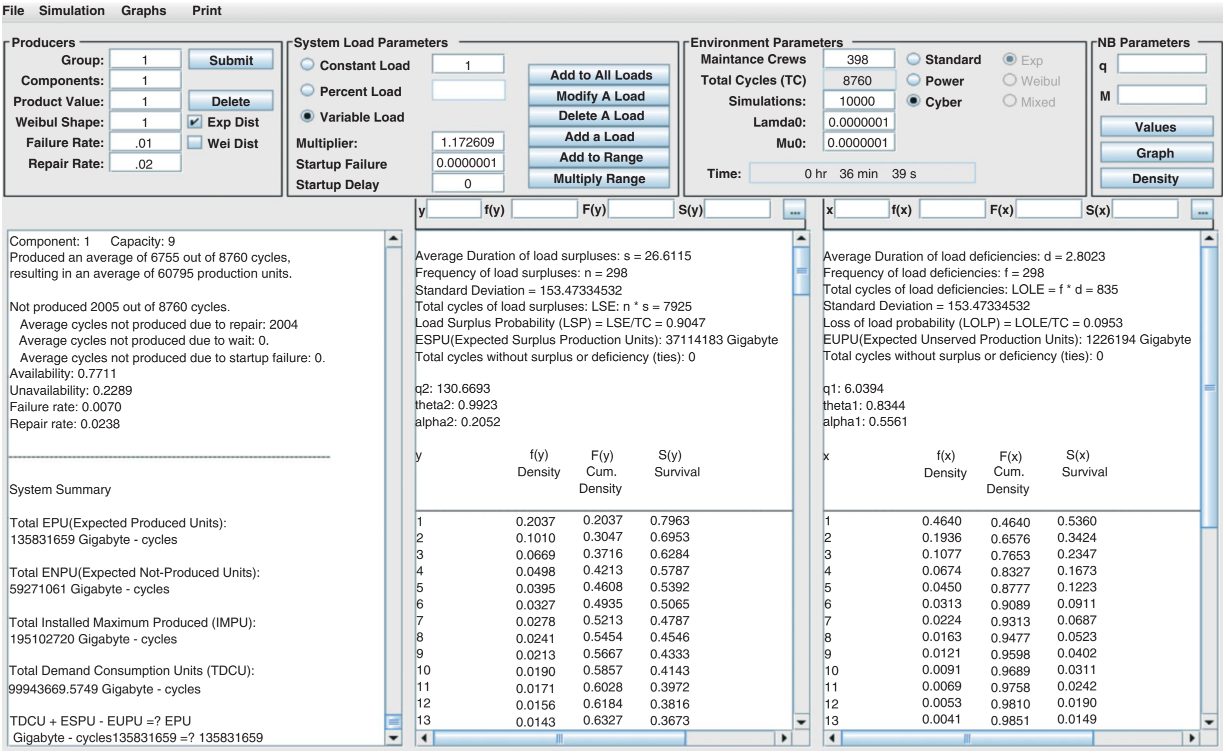
Simulation (DES) as in Figure 18, QoSE = 1 − LoSE = 1 − 0.0953 = **0.9047**. A difference of 0.0337 or roughly 3% is justifiable since it is virtually impossible to process such colossal grids even with supercomputers. See Figure 19 using Alabama Supercomputing Center (www.asc.edu). This printout shows that the digital event simulation and Markov solutions yield comparable results.

DISCUSSIONS ON CLOUD COMPUTING AND FIGHTING CYBERCRIME

CLOUD computing, as so called 5th utility, is becoming such a powerful entity for cyber-users as well as large commercial companies that someday users will not need to buy other than a computer terminal such as in the case we only provide for electric bulbs and wiring to light up our homes from the electric power grid. Meanwhile Web platforms, which supply service-based access to infrastructure services, information, applications, and business processes through Web based CLOUD computing environments are on the rise.[45] However, these cyber CLOUDS at the turn of the 21st century, just like the electric power or gas or water supply companies launching on commercial business in the western hemisphere at the turn of the 20th century, will need to sell highly reliable (not just any) and relatively secure (free of hacking and virus malware) service to their new-breed demanding clients. This is only possible if the service-based system managers can quantify and then duly manage their risks of not meeting the service demand as planned.

In many ways, electric power, or gas, or water, or telephony utilities' challenges will be revisited as they once were eminent in the latter half of the 20th century.[14–17] CLOUDS are clearly related to Grids, both in goals and implementation, but there are differences.[41] CLOUDS as systems are not orthogonal to Grids, nor are they necessarily complementary to Grids. CLOUDS are the evolution of Grids and they can both in turn be viewed as evolution of Clusters. CLOUDS can be viewed as a logical and next higher-level abstraction from Grids which in their current form of deployment and implementation have not been as successful as hoped in engendering distributed applications for higher level applications. It is important to note, we are not suggesting that Grids are not useful, but pointing out, how it is generally agreed that the vision of pervasiveness, ubiquity and interoperability of computing resources and infrastructure in the early days of Grid computing have not come to fruition.[30]

In addition to an overview of CLOUD computing practices, we have also reviewed that the statistical estimation of the sum of loss of service (LoS) events in hours or cycles is available by modeling through a compound Poisson process (NBD: Negative Binomial



Producers
Group: 1
Components: 1
Product Value: 1
Weibul Shape: 1
Failure Rate: .01
Repair Rate: .02
☑ Exp Dist
□ Wei Dist

System Load Parameters
○ Constant Load 1
○ Percent Load
● Variable Load
Multiplier: 1.172609
Startup Failure 0.0000001
Startup Delay 0

Environment Parameters
Maintance Crews 398
Total Cycles (TC) 8760
Simulations: 10000
Lamda0: 0.0000001
Mu0: 0.0000001
○ Standard
○ Power
● Cyber
● Exp
○ Weibul
○ Mixed
Time: 0 hr 36 min 39 s

NB Parameters
q
M

```
Component: 1      Capacity: 9
Produced an average of 6755 out of 8760 cycles,
resulting in an average of 60795 production units.

Not produced 2005 out of 8760 cycles.
   Average cycles not produced due to repair: 2004
   Average cycles not produced due to wait: 0.
   Average cycles not produced due to startup failure: 0.
Availability: 0.7711
Unavailability: 0.2289
Failure rate: 0.0070
Repair rate: 0.0238

-----------------------------------------------------------

System Summary

Total EPU(Expected Produced Units):
135831659 Gigabyte - cycles

Total ENPU(Expected Not-Produced Units):
59271061 Gigabyte - cycles

Total Installed Maximum Produced (IMPU):
195102720 Gigabyte - cycles

Total Demand Consumption Units (TDCU):
99943669.5749 Gigabyte - cycles

TDCU + ESPU - EUPU =? EPU
 Gigabyte - cycles135831659 =? 135831659
```

```
y         f(y)        F(y)        S(y)

Average Duration of load surpluses: s = 26.6115
Frequency of load surpluses: n = 298
Standard Deviation = 153.47334532
Total cycles of load surpluses: LSE: n * s = 7925
Load Surplus Probability (LSP) = LSE/TC = 0.9047
ESPU(Expected Surplus Production Units): 37114183 Gigabyte
Total cycles without surplus or deficiency (ties): 0

q2: 130.6693
theta2: 0.9923
alpha2: 0.2052

y                f(y)      F(y)       S(y)
                 Density   Cum.       Survival
                           Density
1                0.2037    0.2037     0.7963
2                0.1010    0.3047     0.6953
3                0.0669    0.3716     0.6284
4                0.0498    0.4213     0.5787
5                0.0395    0.4608     0.5392
6                0.0327    0.4935     0.5065
7                0.0278    0.5213     0.4787
8                0.0241    0.5454     0.4546
9                0.0213    0.5667     0.4333
10               0.0190    0.5857     0.4143
11               0.0171    0.6028     0.3972
12               0.0156    0.6184     0.3816
13               0.0143    0.6327     0.3673
```

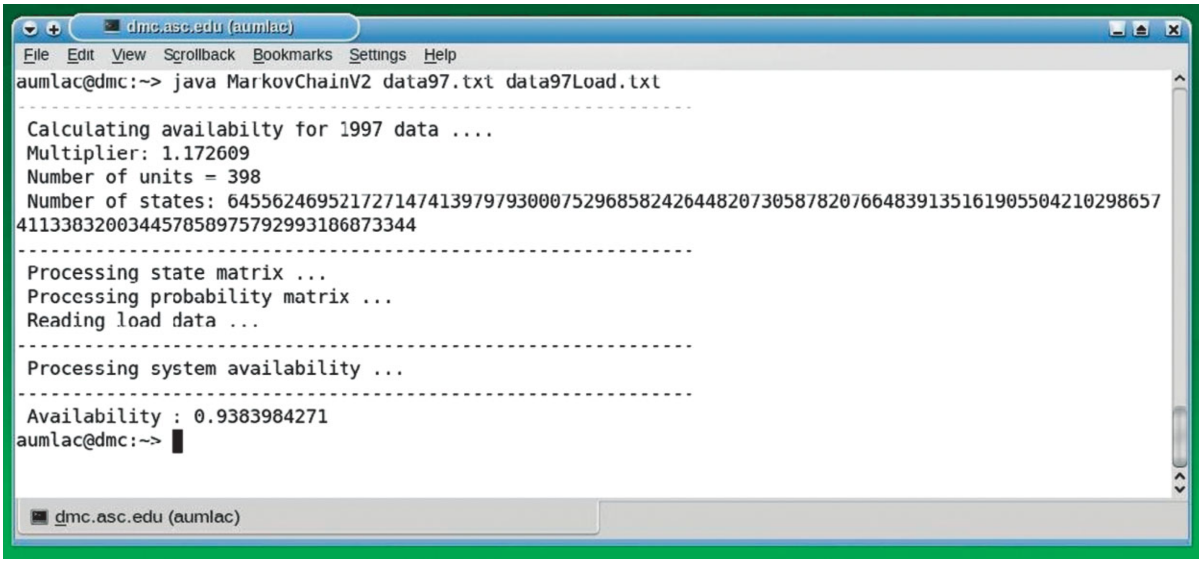
```
x         f(x)        F(x)        S(x)

Average Duration of load deficiencies: d = 2.8023
Frequency of load deficiencies: f = 298
Total cycles of load deficiencies: LOLE = f * d = 835
Standard Deviation = 153.47334532
Loss of load probability (LOLP) = LOLE/TC = 0.0953
EUPU(Expected Unserved Production Units): 1226194 Gigabyte
Total cycles without surplus or deficiency (ties): 0

q1: 6.0394
theta1: 0.8344
alpha1: 0.5561

x                f(x)      F(x)       S(x)
                 Density   Cum.       Survival
                           Density
1                0.4640    0.4640     0.5360
2                0.1936    0.6576     0.3424
3                0.1077    0.7653     0.2347
4                0.0674    0.8327     0.1673
5                0.0450    0.8777     0.1223
6                0.0313    0.9089     0.0911
7                0.0224    0.9313     0.0687
8                0.0163    0.9477     0.0523
9                0.0121    0.9598     0.0402
10               0.0091    0.9689     0.0311
11               0.0069    0.9758     0.0242
12               0.0053    0.9810     0.0190
13               0.0041    0.9851     0.0149
```

FIGURE 18 | Simulation results for a large cyber CLOUD with 398 components in large-scale example (3).



```
aumlac@dmc:~> java MarkovChainV2 data97.txt data97Load.txt
-------------------------------------------------------------
 Calculating availabilty for 1997 data ....
 Multiplier: 1.172609
 Number of units = 398
 Number of states: 645562469521727147413979793000752968582426448207305878207664839135161905504210298657
411338320034457858975792993186873344
-------------------------------------------------------------
 Processing state matrix ...
 Processing probability matrix ...
 Reading load data ...
-------------------------------------------------------------
 Processing system availability ...
-------------------------------------------------------------
 Availability : 0.9383984271
aumlac@dmc:~> █
```

FIGURE 19 | Large-scale cyber CLOUD example Markov Solution for 398 Units using Alabama Supercomputer for large-scale example (3).

Distribution). One assumes the compounding distribution of the NBD to be a logarithmic series distribution. This model is math-statistically a practical approximation for two essential reasons. The first one is the taking into account of the phenomenon of interdependence or true contagious property of the system failures that constitute a cluster at each breakdown in concert with the electric power or cyber system engineering practice. Secondly, the underlying distribution to estimate the probability distribution function for the number of power or cyber system failures in this review paper is no more an asymptotical or an approximate expression as conventionally has been treated,[26,27,11,12] whereas it is established as a closed-form and non-asymptotic exact probability distribution function.[22,25,2]

When the system is in a negative margin at any hour, the contagious condition affects the forthcoming hours until full system recuperation. Due to this true contagious property, the usage of a compounded Poisson with LSD gives mathematically sound results where the failures can be interdependent. The parameters of the Poisson ^ LSD can be obtained in the CLOUD system application by the well-known method, namely frequency and duration method, as earlier exercised in a power generation system reliability evaluation. Contrary to limiting Poisson ^ geometric process, the proposed Poisson ^ LSD gives a closed-form exact distribution for loss of load hours in a power generating or cyber CLOUD computing setting. In the event of a cyber or power system, the process is simulated hourly using CLOURA software based on the additive (cumulative) property of the distributed server farms with their adequate (or inadequate according to budgeting resources) number of repair crews to handle the service outages, therefore meeting the demand of service (load).

The ratio of unsuccessful interception of the load (service demand) cycles or hours, Loss of Service Expected = LoSE = $f \times d$, by the supply side is calculated in addition to the frequency of loss of service (f) and average duration of loss of service (d). Further, parameters of the probability density functions of LoSE and d as random variables are estimated for statistical inference. Load management is also available and desired maintenance planning can be incorporated. In the cyber-world, a CLOUD administrator should benefit by first assessing the reliability and equally, security of CLOUD computing systems rising in popularity. There are other cyber-networks where s–t (source–target) is sought where the topology is directional,[46–48] not regional or national as in this study. Then, the CLOUD managers can prematurely use the said computational tool (CLOURA) to avoid bottlenecks by preliminary planning through emulating the operations, and checking where one can improve. The bottom-line impact is that a CLOUD planning department will know in advance through examining and responding to what–if questions by planning how to weigh its physical resources before actually taking remedies or buying the equipment, or hiring new personnel to fortify repair facilities.

The physical glitches and monetary losses (excluding the erosion of customer trust) could have been many times avoided if the CLOUD operations centers knew the actual reliability status of the supply/demand system and what to do to raise the quality of service (QoS) by taking the necessary precautions so as to plan ahead. Of course not to forget, input data to be entered carries a very important role and responsibility for the success of this software, that is, CLOURA for the job of quantitative risk assessment, and consequently *ad hoc* risk management. It can apply dependence between the production units (components) by using the necessary features as in Figures 11, 17, and 18 for three large examples through rightful evaluations of λ_0 and μ_0.

The much neglected reliability or security assessment amid rampant commercialism will continue to be a liability; if otherwise, with media coverage of glitches becoming everyday news.[49] This is why the CLOUD reliability and security metrics have to be computed and daily operations ought to be simulated to circumvent any potential problems in advance before undesirable glitches happen, and consequently commercial customers lose trust. However, if the growth of Grid technology is to continue, it will be important that Grid systems also provide high reliability. In particular, it will be critical to ensure that Grid systems are reliable as they continue to grow in scale, exhibit greater dynamism, and become more heterogeneous in composition.[29] In brief, ensuring Grid system reliability in turn requires that the specifications used to build these systems fully support reliable Grid services[30–32]. In Ref 33, whereas, the authors describe a CLOUD-based infrastructure that they have developed that is optimized for wide area, and high performance networks, designed to support data mining applications. The said infrastructure consists of a storage CLOUD called Sector and a compute CLOUD called Sphere. They describe two applications that they have built using the CLOUD and some experimental studies. Therefore, with the significant advances in Information and Communications Technology (ICT) over the last half century, there is an increasingly perceived vision that computing will one day be the 5th utility (after water, electricity, gas, and telephony). This computing utility, like all other four existing utilities, will provide the basic level of computing service that is considered essential to meet the everyday needs of the general community.[34] To deliver this vision, a number of computing paradigms have been proposed, of which the latest one is known as CLOUD computing. In the ref.[34], they define CLOUD computing and provide the architecture for creating CLOUDS with market-oriented resource allocation by leveraging technologies such as Virtual Machines (VMs).

Grid computing has been the subject of many large national and international IT projects.[41] However, not all goals of these projects have been achieved. In particular, the number of users lags behind the initial forecasts laid out by proponents of Grid technologies. This particular underachievement may have led to claims that the Grid concept as a whole is on its way to being replaced by CLOUD computing and various X-as-a-Service approaches. In another paper, the authors try to analyze the current situation, and to identify promising directions for future Grid development. Although there are shortcomings in current Grid systems, we are convinced that the concept as a whole remains valid and can benefit from new developments, including CLOUD computing. Furthermore, we strongly believe that some future applications will require the Grid approach and that, as a result, further research is required in order to turn this concept into reliable, efficient and user-friendly computing platforms.[35]

To summarize, the future of computing lies in CLOUD computing, whose major goal is to reduce the IT services' costs while increasing processing throughput, and decreasing processing time, increasing reliability, availability and flexibility.[36] CLOUD computing is a new paradigm where computing resources (from data storage to complete configurations of distributed systems) are made available and offered over the Internet as scalable, on-demand (Web) services. In CLOUD computing, the resources hosted within CLOUDS can be anything: they could be database services, virtual servers (virtual machines), complete service workflows or complex configurations of distributed computing systems such as clusters. Regardless of their nature, all resources are provided via services to clients (users or software processes) by computers rented from the CLOUD, such as those offered by, for example, Amazon, Google, Microsoft, rather than by private systems. The services are provided on demand and clients only pay for the quantity of resources (data storage, computation, etc.) they use. Moreover, CLOUD computing is becoming an adoptable technology for many of the organizations with its dynamic scalability and usage of virtualized resources as a service through the Internet. CLOUD computing will have a significant impact on the educational environment in the future.

Additionally, CLOUD computing is an excellent alternative for educational or state institutions which are especially under budget shortage in order to operate their information systems effectively without spending any more capital for the computers and network devices. Universities take advantage of available CLOUD-based applications offered by service providers and enable their own users/students to perform business and academic tasks.[37]

Authors of this review have surveyed techniques to assess reliability or security of such CLOUD computing systems assisted by statistical methodology. The multifarious risks posed by this new IT delivery model, that is, CLOUD computing, ranging from a potential lack of awareness of where data is held to possible insider threats and vendor lock-in, have been well documented.[38] One of the key promises of CLOUD is the speed and ease with which organizations can temporarily access additional compute resources if required, so-called 'CLOUD-bursting'. However, there is a tension between this, and the need for due diligence via mechanisms such as auditing (versus security breaches), which inevitably take time.[38] In his keynote at RSA 2010 (an encryption and network security company) President Art Coviello spoke of the industry's latest and greatest challenge: Securing CLOUD computing. 'A new wave of computing is struggling to take hold', he announced. 'It is called CLOUD computing. We must play an essential role in making CLOUD computing a secure reality', Coviello continued by referring to the members of the information security industry in the audience. 'The information security industry needs to leverage technology to enable secure CLOUD computing'.[39]

Fighting cybercrime in the CLOUD is crucial because cybercrime is now a roughly $100 billion market, surpassing the illegal drug trade. The openness of CLOUD computing denotes that cybercriminals are a busy bunch these days. They steal identities, grab credit and debit card account numbers, and wage a myriad of other attacks on unwitting users, businesses, and vulnerable websites. Their weapon of choice is the malware injection. Among the most vulnerable—and the most lucrative for cybercriminals due to the sites' enormous reach—are trusted, popular sites with unpatched vulnerabilities.[8] The risk of hackers getting at the data is only the smallest part of the risk involved in storing data with third parties. Yet not all CLOUD security requires remote storage of private data, and in many cases the addition of a CLOUD security component to an overall network security strategy is very beneficial.

Like it or not, CLOUD security elements are being integrated into everything from antivirus programs to firewalls.[41] The trick is to know which technologies have the best benefit-to-risk ratio. There are many benefits that can come from using CLOUD security and whether you know it or not, you are probably using it already. Nearly all of the major web browsers now have built-in features that check URL blacklists, which are regularly updated. When doing a search on Yahoo, for example, sites that have failed Google's antivirus checks receive a 'This site may harm your computer' notice. If attempting to go on to a site like this from within Firefox, the browser will display a message warning that the site has been blacklisted. This is one excellent use of CLOUD security.[7] Antivirus scanning is an expensive operation and using multiple antivirus scanners requires a lot of CPU and memory. By doing this ahead of time and doing it once for the benefit of many, the resource burden has been reduced overall. Plus, through the power of Grid and CLOUD computing, more antivirus engines can be employed than an average network gateway or workstation.[9] After all, one should remember what J. Viega from McAffee says,[1] 'The CLOUD offers several advantages but until some of its risks are better understood many major players might hold back'. This review has done that, that is, while reviewing a range of CLOUD computing practices, it has also examined risk modeling schemes.

ACKNOWLEDGEMENTS

The author duly thanks Mehmet Gokmen from DEU's Statistics Department at Izmir-Turkey for some of his experimental contributions to this earlier analysis during his M.S. thesis studies (1994–1996) under the author's supervision. Thanks also go to David Tyson for programming of the java code in the discrete event simulations (2007–2009).

REFERENCES

1. John V. Cloud computing and the common man. *Computer* 2009:106–108.
2. Sahinoglu M. *Trustworthy Computing: Analytical and Quantitative Engineering Evaluation*. New York: John Wiley & Sons Inc.; 2007.
3. Cueva-Parra Luis A, Sahinoglu M., Security metrics on Cloud computing using statistical simulation and Markov process. *12th SDPS Transdisciplinary Conference Proceedings on Integrated Systems, Design and Process Science*, Montgomery, Alabama; November 2009.
4. Cueva-Parra Luis A, Sahinoglu M, Tyson D, Das S. Statistical inference on security metrics in Cloud computing for large cyber-systems. *Joint Statistical Meeting (JSM), Invited session: Quantitative Security and Cyber-Systems*. Washington DC; August 2009.
5. Lindstrom P. Yes, you can quantify risk, For Pete's Sake. *ISSA J* (*Inform Syst Sec Assoc*); www.issa.org (Accessed April 2008).
6. Chew E, Swanson M, Stine K, Bartol N, Brown A, Robinson W., *Performance Measurement Guide for Information Security*, vol. 1. National Institute of Standards and Technology (NIST); 2007, 800–855.
7. Hinson G. Seven myths about information security metrics. *Inform Syst Sec Assoc (ISSA) J*. July 2006 (on-line).
8. Jansen W. *Directions in Security Metrics Research*. NISTIR 7564; 2009.
9. Jaquit A. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley; 2007.
10. Sahinoglu M, Longnecker M, Ringer L, Singh C, Ayoub AK. Probability distribution function for generation reliability indices—analytical approach. Paper 82JPGC 603-9 *IEEE/PES/ASME/ASCE Joint Power Generation Conference*, 1982, and *IEEE Trans Power App Syst* 1983, 102:1486–1493.
11. Billinton R. *Power System Reliability Evaluation*. Gordon & Breach Science Publishers; 1974.
12. Singh C, Billinton R. *System Reliability Modeling and Evaluation*. London: Hutchinson Educational; 1977.
13. Sahinoglu M. *Statistical Inference of Reliability Performance Index for Electric Power Generation Systems*, Ph.D. Dissertation, 77843. College Station, TX, Institute of Statistics, Texas A&M University; 1981.
14. Ayoub AK, Guy JD, Patton AD. Evaluation and comparison of some methods for calculating generation system reliability. *IEEE Trans Power App Syst* 1970 PAS-89:514–521.
15. Billinton R. Bibliography on the application of probability methods on power system reliability evaluation. *IEEE Winter Power Meeting*. Paper No. 93-PWR, New York; 1971.
16. Allan RN, Billinton R, Lee SH. Bibliography on the application of probability methods in power system reliability. *IEEE Trans Power App Syst* 1984, 103: 275–282.
17. Patton AD, Singh C, Sahinoglu M. Operating considerations in generation reliability modeling—analytical approach. *IEEE Winter Power Meeting Power App Syst* 1980, A80-082-8:2656–2663.
18. Sahinoglu M, Gebizlioglu O. Exact pmf estimation of system reliability indices in a boundary crossing problem. *First World Congress of the Bernoulli Society, Tashkent, Uzbekistan*, and *Commun Fac Sci Univ Ank*, Series A1 V.36 Number; 1986, 115–121.
19. Sahinoglu M. The limit of sum of Markov Bernoulli variables in system reliability evaluation. *IEEE Trans Reliab* 1990, 39:46–50.
20. Serfozo RF. Compound Poisson approximations for sums of random variables. *Ann Prob* 1986, 14: 1391–1398.
21. Sahinoglu M. Negative Binomial (Poisson^Logarithmic) density estimation of the software failure count. *Proceedings of the Fifth International Symposium On*

Computer And Information Sciences (ISCIS), Goreme (Cappadocia), Turkey, vol. 1; 1990, 231–240.

22. Sahinoglu M. Compound Poisson software reliability model. *IEEE Trans Softw Eng* 1992, 18:624–630.
23. Brown B. *Some Tables of the Negative Binomial Distribution and Their Use*, Memorandum RM-4577-PR. Santa Monica, CA: The Rand Corporation; 1965.
24. Sherbrooke CC. *Discrete Compound Poisson Processes and Tables of the Geometric Poisson Distribution*, Memorandum RM-4831-PR. Santa Monica, CA: The Rand Corporation; 1966.
25. Gokmen M. *An Exact Compound Poisson (Poisson^LSD) Probability Density Function for Loss of Load in Electric Power Generation Reliability Evaluation*, M.S. Thesis (Supervised by: Dr. M. Sahinoglu), Department of Statistics, Dokuz Eylul University, Izmir; May 1996.
26. Hall JD, Ringlee RJ, Wood AJ. Frequency and duration methods for power system reliability calculations—part I: generation system model. *IEEE PAS-87* 1968, 9: 1787–1796.
27. Ayoub AK, Patton AD. A frequency and duration method for generating system reliability evaluation. *IEEE Trans Power App Syst* 1976, 95: 1929–1933.
28. Student. *Biometrika* 1919, 12:211–215.
29. Dabrowski C. Reliability in grid computing systems. *US Government Work. Concurrency and Computation: Practice and Experience*, vol. 21. Wiley & Sons; 2009, 927–959.
30. Jha S, Merzky A, Fox G. Using Clouds to provide grids with higher levels of abstraction and explicit support for usage modes. *Concurr Comput: Pract Exp* 2009, http://grids.ucs.indiana.edu/ptliupages/publications/cloud-grid-saga_rev.pdf.
31. Liu H, Gridbatch DO. Cloud computing for large-scale data-intensive batch applications. *8th IEEE International Symposium on Cluster Computing and the Grid*; 2008.
32. Buyya R, Yeo CS, Venugopal S., Market-oriented Cloud computing: vision, hype, and reality for delivering services as computing utilities. *Proceedings of the 10th IEEE International Conference on Cluster Computing and the Grid;* 2008.
33. Grossman Robert L, Gu Y, Sabala M, Zhang W. Compute and storage clouds using wide area high performance networks. *Future Gener Comput Syst* 2009, 25:179–183. ISSN 0167-739X, http://www.sciencedirect.com/science/article/B6V06-4T3DCND-4/2/56835e46b2fcd855b1d818940da3a814. doi: 10.1016/j.future.2008.07.009
34. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener Comput Syst* 2009, 25:599–616. ISSN 0167-739X, http://www.sciencedirect.com/science/article/B6V06-4V47C7R-1/2/d339f420c2691994442c9198e00ac87e. doi: 10.1016/j.future.2008.12.001.
35. Pallis G. "The New Frontier of Internet Computing", IEEE Internet Computing, September/October 2010, 70–73.
36. Goscinski A, Brock M. Toward dynamic and attribute based publication, discovery and selection for Cloud computing *Future Gener Comput Syst* 2010, 26: 947–970.
37. Ercan T. Effective use of Cloud computing in educational institutions. *Proced–Social Behav Sci* 2010, 2: 938–942.
38. Everett C. A question of thrust. *Comput Fraud Sec* 2009:5–7.
39. Securing Cloud computing is industry's responsibility. *Infosecurity* 2001, 7:11.
40. Hawthorn N. Finding security in the Cloud. *Comput Fraud Sec* 2009:19–20.
41. Alnoshan AA, Rotenstreich S. Distributed grid-type architectures. *Wiley Interdiscip Rev: Comput Stat* 2010, 2:287–302. doi: 10.1002/wics.94.
42. Sahinoglu M, Selcuk AS. Application of Monte Carlo simulation method for the estimation of reliability indices in electric power generation systems. *Doga-Tr. J Eng Environ Sci* 1993, 17:157–163.
43. Sahinoglu M. Global benefits of interconnection among Balkan power systems (1990). Final Report, *Coordinating Committee of Development of Interconnection of the Electric Power Systems of Balkan Countries*, Geneva-Switzerland, Project No:06.3.1.3.-EP/GE.2/R.70.3; 1988.
44. Sahinoglu M, Libby D, Das SR. Measuring availability indices with small samples for component and network reliability using the Sahinoglu-Libby probability model. *IEEE Trans Instrum Measure* 54: 1283–1295.
45. Press Releases. *Web platform and WOA: Gartner identifies the top 10 strategic technologies for 2008*, October 2007, http://www.gartner.com/it/page.jsp?id=530109, Orlando.
46. Sahinoglu M, Ramamoorthy CV., RBD tools using compression and hybrid techniques to code, decode and compute s-t reliability in simple and complex networks. *IEEE Trans Instrum Measure, Special Guest Edition on Testing* 2005, 54:1789–1799.
47. Sahinoglu M, Rice B. Network reliability evaluation. *Wiley Interdiscip Rev: Comput Stat* 2010, 2:189–211. doi:10.1002/wics.81.
48. Sahinoglu M, Rice B, Tyson D. An analytical exact RBD method to calculate s-t reliability in complex networks. *IJCITAE—Int J Comput Inform Technol Eng*, 2008, 2:95–104. ISSN: 0973-743X.
49. Worthen G, Vascellaro J. E mail glitch shows pitfalls of online software—photo: services like Gmail run on vast computer farms. A Google center in Lenoir, N.C. *Media Market Wall Street J*, 2009:4–5.