Advances in Computational Sciences and Technology (ACST).
ISSN 0973-6107 Volume 10, Number 6 (2017), pp. 1773–1784

http://www.ripublication.com/acst.htm

An Algorithmic Approach to Cloud Computing using Graph Theoretical Modelling

D. Pandiaraja

National Centre of Excellence,
Statistical and Mathematical Modelling on
Bio-Resources Management,
and Department of Mathematics,
Thiagarajar college,
Madurai-625 009, Tamilnadu, India.

R.P. Aditya

Department of Mathematics,
V.V. Vanniaperumal college for women,
Virudhunagar-626 001, Tamilnadu, India.

S. Abirami

Department of CA and IT,
Thiagarajar college,
Madurai-625 009, Tamilnadu, India.

Abstract

The notion of cloud computing has been widely examined by various researchers over the last few years. It has provoked major impact on global IT ecosystem. Users have attained numerous benefits because of the evolution of clouds. It indented us to define cloud-user graph and cloud graph. We proposed some centrality index to find the vital role played by clouds. We gave an algorithm for closeness centrality using reachability to detect the interrelationships between clouds.

AMS subject classification:
Keywords:

1. Introduction

Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. Cloud computing is attractive to business owners as it eliminates the requirement for users to plan ahead for provisioning, and allows enterprises to start from the small and increase resources only when there is a rise in service demand. However, despite the fact that cloud computing offers huge opportunities to the IT industry, the development of cloud computing technology is currently at its infancy, with many issues still to be addressed [1].

Computing is a virtual pool of computing resources. It provides computing resources in the pool for users through internet. Integrated cloud computing is a whole dynamic computing system. It provides a mandatory application program environment. It can deploy, allocate or reallocate computing resource dynamically and monitor the usage of resources at all times. Generally speaking, cloud computing has a distributed foundation establishment, and monitor the distributed system, to achieve the purpose of efficient use of the system. Cloud computing collects all the computing resources and manages them automatically through the software [2]. In the process of data analysis, it integrates the previous data and present data to make the collected information more accurate and provide more intelligent services for users and enterprises. The users need not care how to buy servers, softwares and so on. Users can buy the computing resources through internet according to their own needs. Cloud computing does not depend on special data center, but we can look it as the inevitable product of grid computing and efficiency computing. However, compared with general network service, cloud computing is easy to extend, and has an simple management style. Cloud is not only to simply collect the computer resource, but also provides a management mechanism and can provide services for millions of users simultaneously [2].

There are many definition of cloud computing, but the definition provided by The National Institute of Standards and Technology (NIST) seems to cover all essential aspects of cloud computing. žCloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

In recent years, there has been a surge of interest in the analysis of networks as models of complex systems. Graph theoretic tools have been widely used in analysing social and economic networks, information networks, technological networks, biological networks [3], [4]. It urge us to identify the cloud which can interact with many clouds and determine the closeness between the clouds. In this paper we have proposed a graph theoretical model [5] to stimulate cloud network and used centrality to assess the closeness using reachability in the cloud network.

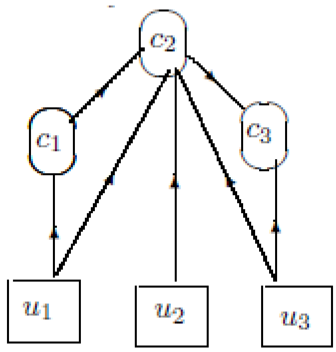

Figure 1. CU

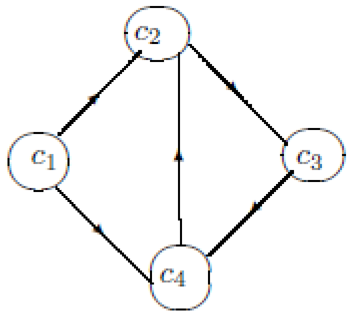

Figure 2. C

2. Model description

The cloud network has been illustrated as a directed graph by considering clouds and users as vertices and the interaction path as arcs in Figure 1. The digraph has been designated as cloud-user graph CU and cloud graph C. Since communication between clouds are more important, we use cloud graph C to analyze the reachability between clouds.

3. Preliminaries

In this paper, we follow the definitions and notations of Graph theory [6], [7]. In ***n*-dimensional butterfly graph**, vertices are pairs (w, i), where w is a binary string of length n and i is an integer in the range 0 to n and with directed edges from (w, i) to $(w^{'}, i+1)$ iff $w^{'}$ is identical to w in all bits with the possible exception of the $(i+1)^{th}$ bit counted from the left. The **n-dimensional butterfly graph** has $2^n(n+1)$ vertices and $2^{n+1}n$ edges.

Theorem 3.1. A nontrivial tournament T is Hamiltonian if and only if T is strong.

Theorem 3.2. Every tournament contains a Hamiltonian path.

4. Definitions

Definition 4.1. A cloud-user graph CU is a digraph $CU = (V, A)$ with vertex set $V = V_1 \cup V_2$ and edge set $E = E_1 \cup E_2$, where $V_1 = \{c_i/c_i$ is a cloud$\}$, $V_2 = \{u_j/u_j$ is an user$\}$ and E_1 and E_2 are set of arcs such that $E_1 \subseteq V_1 \times V_1$ and $E_2 \subseteq V_2 \times V_1$.

Definition 4.2. A cloud graph C is a digraph $C = (V, A)$ with vertex set $V = \{c_i/c_i$ is a cloud $\}$ and edge set $E \subseteq V \times V$.

Throughout this paper we consider only cloud graph C.

5. Centrality

Centrality is used for finding paramount vertices/edges in graphs. In social networks the vertices refer to people/actors and the edges refer to relationships, where the relationship is dependent on the social network. In a communication network, the vertices may be servers and the edges might be physical connections between the servers. For email networks, the vertices will be senders/receives and the edges refer to emails sent between the sender and receiver [8]. Here, in a cloud network, the vertices are clouds and the arcs will be the interaction between them.

5.1. Reachability

Over the years, several measures of centrality have been proposed. Since centrality is a fundamental concept in the network analysis new centrality measures are arising daily based on application. There are many matrices of networks such as adjacency matrix, laplacian matrix, normalized matrix, modularity matrix which contain some useful information about the structure of the network [9]. Although there are number of existing measures for quantifying the structural properties of undirected graphs, there are relatively few corresponding measures for directed graphs. For directed networks, centrality measure usually based on indegree or outdegree. To determine the interactability of clouds in a cloud graph, we need a measure from which the communicability (reachability) of clouds can be found. So, a kind of measure called PR (point reachability) measure is introduced. It serves to quantify that in a cloud network some clouds are more interactable than other.

PR measure is the number of vertices reachable from a point x_k ,

$$C_{cr}(x_k) = \sum_{l=1}^{n} r(x_k, x_l), \text{ where } r(x_k, x_l) = \begin{cases} 1 \text{ iff } x_l \text{ is reachable from } x_k \\ 0 \text{ otherwise.} \end{cases}$$

$r(x_k, x_k) = 1\forall\, k$

$C_{cr}(x_k)$ is large if vertex x_k is reached to large number of vertices and small if x_k tends to be cut off from such reachability.

A given node, x_k can atmost be reached to n other vertices(including itself) in a digraph. The maximum of $C_{cr}(x_k)$, therefore is n. Then $C_{cr}{}'(x_k) = \dfrac{\sum_{l=1}^{n} r(x_k, x_l)}{n}$ is the proportion of other vertices that are reachable from x_k. The maximum value is achieved for every nodes only when the given digraph is strongly connected and it is 1.

Proposition 5.1. Let C be a cloud graph. $C_{cr}(x) = n\forall\, x \in V(C)$ if and only if C is strongly connected.

Proof. $C_{cr} = n,\ \forall x \in V(C)$ iff every vertex can reach all the remaining $n - 1$ vertices and itself iff for every pair of vertices x and y there is a (x, y) path and (y, x) path in C

iff C is strongly connected. ■

Proposition 5.2. C_{cr} of a node in a cloud graph C is 1 if and only if $d^+(x) = 0$.

Proof. Let $x \in V$. $C_{cr} = 1$ if and only if x can reach only itself not any other vertices if and only if no arc coming out of x if and only if $d^+(x) = 0$. ■

Proposition 5.3. Let C be a path cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$. $C_{cr}(x_1) = n$ and $C_{cr}(x_i) = C_{cr}(x_{i-1}) - 1 \; \forall i = 2, 3, \ldots n$.

Proof. Since C is a path cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$ and arc set $A(C) = \{(x_i, x_{i+1})/i = 1 \text{ to } n\}$. Vertex x_1 reach vertices $x_1, x_2, \ldots, x_n$. Vertex x_2 reach vertices $x_2, x_3, \ldots, x_n$. Likewise, vertex x_{n-1} reach vertices x_{n-1}, x_n. Vertex x_n reach x_n only. Reachability decreases by 1 in each case. So, $C_{cr}(x_1) = n$ and $C_{cr}(x_i) = C_{cr}(x_{i-1}) - 1 \; \forall i = 2, 3, \ldots n$. ■

Proposition 5.4. Let C be a cycle cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$. Then $C_{cr}(x_i) = n \; \forall i$.

Proof. Since cycle cloud graph is strongly connected, by Proposition 5.1 $C_{cr}(x_i) = n \; \forall i$. ■

Proposition 5.5. Let C be a oriented star cloud graph with vertex set $V(C) = \{y, x_1, x_2, \ldots, x_r\}$ where y is the center vertex. Then

$$C_{cr}(x) = \begin{cases} d^+(y) + 1 & \text{if } x \text{ is the center vertex} \\ 1 & \text{if } x \in N^+(y) \\ d^+(y) + 2 & \text{if } x \in N^-(y) \end{cases}$$

Proof. Center vertex y reach all the vertices in $N^+(y)$ and itself. $C_{cr}(y) = |N^+(y)| + 1 = d^+(y) + 1$. By Proposition 5.2, $C_{cr}(x) = 1 \; \forall x \in N^+(y)$. For $x \in N^-(y)$, x reach y, itself and all vertices in $N^+(y)$. $C_{cr}(x) = |N^+(y)| + 2 = d^+(y) + 2 \; \forall x \in N^-(y)$. ■

Proposition 5.6. Let C be a n-dimensional butterfly graph then $C_{cr}((w, n)) = 1$ and $C_{cr}((w, n - k)) = C_{cr}((w, n - k + 1)) \times 2 + 1$, where $k = 1, 2, \ldots, n$.

Proof. Since outdegree of every vertices at level n is 0, $C_{cr}((w, n)) = 1$. Since every vertices in i level reaches two vertices in level $i+1$ where $i = 0$ to $n-1$, $C_{cr}((w, n-k)) = C_{cr}((w, n - k + 1)) \times 2 + 1$, where $k = 1, 2, \ldots, n$. ■

5.2. Closeness centrality using Reachability

To find how a cloud A communicate to all others is to calculate the proportion of other clouds that A can reach in one step, two steps, three steps, …, ∞(or, alternatively the proportion of nodes that reach A in $n - 1$ steps. ∞ denote A cannot reach that cloud. Now calculate a single index for each node by summing up the proportion of other nodes reached(for the first time) at a given distance, appropriately weighted. (example, 0 for nodes at distance ∞, 1 for nodes at distance 1, $\frac{1}{2}$ for nodes at distance 2.

An important node is typically "close" to, and can communicate quickly with, the other nodes in the network.

Let $x \in V$. Define $Cccr(x) = \frac{1}{\infty}.\frac{l_1}{n-1} + \frac{1}{1}.\frac{l_2}{n-1} + \frac{1}{2}.\frac{l_3}{n-1} + \ldots + \frac{1}{n-1}.\frac{l_n}{n-1}$, where l_i, $i = 1, 2, \ldots, n$ are number of clouds at distance $\infty, 1, 2, \ldots, n-1$ respectively. Note that $0 \leq l_i \leq n-1, \forall i$ and $\sum_{i=2}^{n} l_i = C_{cr} - 1$. In [10] the problem of labeling the nodes of a graph in a way that will allow one to compute the distance between any two nodes directly from their labels were considered. Here the index C_{ccr} of each node is calculated to find out the nearest cloud.

Proposition 5.7. Let C be a cloud graph. Then $C_{ccr}(x) = 1 \forall\, x \in V(C)$ if and only if $d(x, y) = 1\ \forall y$.

Proof. $C_{ccr}(x) = 1$ if and only if $l_2 = n-1, l_1 = 0, l_3 = l_4 = \ldots = l_n = 0$ if and only if x reach all the $n-1$ vertices by distance 1 if and only if $d(x, y) = 1\ \forall y$. ∎

Proposition 5.8. For every cloud graph C, $C_{ccr}(x) = 0$ if and only if $d^+(x) = 0$.

Proof. Let $x \in V$. $C_{ccr}(x) = 0$ if and only if x cannot reach any vertices if and only if no arc coming out of x if and only if $d^+(x) = 0$. ∎

Proposition 5.9. Let C be a path cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$. Then $C_{ccr}(x_n) = 0$ and $C_{ccr}(x_j) = \frac{1}{n-1}\sum_{i=1}^{n-j}\frac{1}{i}\ \forall j = 1, 2, 3, \ldots n-1$.

Proof. $A(C) = \{(x_i, x_{i+1})/i = 1 \text{ to } n-1\}$. Since $d^+(x_n) = 0, C_{ccr}(x_n) = 0$. Let $x_i \in V(C) - \{x_n\}$. Then x_i reach vertices $x_{i+1}, x_{i+2}, \ldots, x_n$ by distances $1, 2, \ldots, n-i$.

$$Cccr(x_i) = \frac{1}{\infty}.\frac{l_1}{n-1} + \frac{1}{1}.\frac{l_2}{n-1} + \frac{1}{2}.\frac{l_3}{n-1} + \ldots + \frac{1}{n-i}.\frac{l_{n-i+1}}{n-1}.$$

$l_1 = i-1, l_2 = l_3 = \ldots = l_{n-i+1} = 1$

$$Cccr(x_i) = 1.\frac{1}{n-1} + \frac{1}{2}.\frac{1}{n-1} + \ldots + \frac{1}{n-i}.\frac{1}{n-1} = \frac{1}{n-1}\left[1 + \frac{1}{2} + \ldots + \frac{1}{n-i}\right]$$

$$= \frac{1}{n-1}\sum_{j=1}^{n-i}\frac{1}{j}\ \forall i = 1, 2, 3, \ldots n-1.$$ ∎

Proposition 5.10. Let C be a cycle cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$. Then $C_{ccr}(x_i) = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{1}{i}\ \forall i$.

Proof. The index C_{ccr} is same for all clouds in cycle cloud graph. For $x_i \in V$, x_i reach $x_{i+1}, x_{i+2}, \ldots, x_{i+n-1}$ by distances $1, 2, \ldots, n-1$.

So, $l_1 = 0, l_2 = l_3 = \ldots l_n = 1$.

$$Cccr(x_i) = 1.\frac{1}{n-1} + \frac{1}{2}.\frac{1}{n-1} + \ldots + \frac{1}{n-1}.\frac{1}{n-1} = \frac{1}{n-1}\left[1 + \frac{1}{2} + \ldots + \frac{1}{n-1}\right]$$
$$= \frac{1}{n-1}\sum_{j=1}^{n-1}\frac{1}{j}. \qquad \blacksquare$$

Proposition 5.11. Let C be a oriented star cloud graph with vertex set $V(C) = \{y, x_1, x_2, \ldots, x_r\}$, where y is the center vertex. Then

$$C_{ccr}(x) = \begin{cases} \dfrac{d^+(y)}{r} & \text{if } x \text{ is the center vertex} \\ 0 & \text{if } x \in N^+(y) \\ 1.\dfrac{1}{r} + \dfrac{1}{2}\dfrac{d^+(x)}{r} & \text{if } x \in N^-(y) \end{cases}$$

Proof. Center vertex y reach all the vertices in $N^+(y)$ by distance 1. $C_{ccr}(y) = 1.\frac{l_2}{r} = \frac{|N^+(y)|}{r} = \frac{d^+(y)}{r}$. For $x \in N^+(y)$, $d^+(x) = 0$, $C_{ccr}(x) = 0\ \forall x \in N^+(y)$. For $x \in N^-(y)$, x reach y by distance 1and reach all vertices in $N^+(y)$ by distance 2. So, $l_2 = 1, l_3 = d^+(y)$. $C_{ccr}(x) = \frac{1}{r} + \frac{1}{2}\frac{d^+(y)}{r}\ \forall x \in N^-(y)$. $\blacksquare$

Proposition 5.12. Let C be a n- dimensional butterfly graph with $|V| = N$ then $C_{ccr}((w, n)) = 0$ and $C_{ccr}((w, l)) = \frac{1}{N-1}\sum_{i=1}^{N-l}\frac{1}{i}2^i$.

Proof. $Cccr(w, n) = 0$, since $d^+((w, n)) = 0,\ \forall w$. For $i = 0$ to n, every vertices in i level reaches 2 vertices in level $i+1$ by distance 1, reaches 2^2 vertices in level $i+2$ by distance 2, reaches 2^3 vertices in level $i+3$ by distance 3, ..., reaches 2^{n-i} vertices in level n by distance $n-i$ respectively.

$Cccr((w, l)) = \frac{1}{\infty}.\frac{l_1}{N-1} + \frac{1}{1}.\frac{l_2}{N-1} + \frac{1}{2}.\frac{l_3}{N-1} + \ldots + \frac{1}{n-l}.\frac{l_{n-l+1}}{N-1}$.

$l_1 = l-1, l_2 = 2, l_3 = 2^2, l_4 = 2^3 \ldots l_{n-l+1} = 2^{n-l}$

$Cccr((w, l)) = 1.\frac{2}{N-1} + \frac{1}{2}.\frac{2^2}{N-1} + \ldots + \frac{1}{n-l}.\frac{2^{n-l}}{N-1} = \frac{1}{N-1}[\frac{2}{1} + \frac{2^2}{2} + \ldots + \frac{2^{n-l}}{n-l}]$

$$= \frac{1}{N-1}\sum_{j=1}^{n-l}\frac{2^j}{j}\ \forall l = 0, 1, 2, 3, \ldots n. \qquad \blacksquare$$

6. Interactability of Cloud graph

Definition 6.1. Let C be a weighted cloud graph where the weight $w(i, j)$ of each arc (i, j) denote the number of clouds reached through that edge from i [10]. That is, $w(i, j) = |A_{ij}|$ where $A_{ij} = \{z/\exists$ a directed walk starting from the arc (i, j) and reaches $z\}$. Choose a path(open) of maximum length in C.

The interactability of a cloud graph is defined by $I(C) = \sum_{i=1}^{n-1} w(i, i+1)$ where $123\ldots n$ is a path of maximum length in C.

Note 6.2. $A_{ij} \neq \emptyset$, since $j \in A_{ij}$.

Proposition 6.3. For any arc (x, y) in C, $w(x, y) = 1$ if and only if $d^+(y) = 0$.

Proof. $w(x, y) = 1$ if and only if $|A_{xy}| = 1$ if and only if $A_{xy} = \{y\}$ if and only if $d^+(y) = 0$. ■

Proposition 6.4. Let C be a path cloud graph. Then $I(C) = \frac{n(n-1)}{2}$.

Proof. Let C be a path cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$. $w(x_i, x_j) = \lfloor\frac{i+j}{2}\rfloor$. $I(C) = \sum_{i=1}^{n-1} w(x_i, x_{i+1}) = 1 + 2 + \ldots + n - 1 = \frac{n(n-1)}{2}$. ■

Proposition 6.5. Let C be a cycle cloud graph. Then $I(C) = n(n-1)$.

Proof. Let C be a cycle cloud graph with vertex set $V(C) = \{x_1, x_2, \ldots, x_n\}$. $w(x_i, x_j) = n\ \forall (x_i, x_j) \in A(C)$. $I(C) = \sum_{i=1}^{n-1} w(x_i, x_{i+1}) = \underbrace{n + n + \ldots + n}_{n-1 times} = n(n-1)$. ■

Proposition 6.6. Let C be a oriented star cloud graph and x be the center vertex. Then

1. if $d^+(x) = 0$ or $d^-(x) = 0$ then $I(C) = 1$.
2. if $d^+(x) = l$ and $d^-(x) = m$ then $I(C) = l + 2$.

Proof. If $d^+(x) = 0$ or $d^-(x) = 0$ then $w(x, y) = 1$ or $w(y, x) = 1$, $\forall y \neq x$ and path maximum length is 1. Hence $I(C) = 1$. If $d^+(x) = l$ and $d^-(x) = m$ then $w(z, x) = l + 1$ and $w(x, y) = 1$, $\forall y, z \neq x$ and path $z - y$ maximum length is 2. Hence $I(C) = l + 2$. ■

Proposition 6.7. Let C be a n-dimensional butterfly cloud graph. Then $I(C) = 2(2^n - 1) - n$.

Proof. Arcs from level i to $i + 1$ level where $i = 0$ to n, receives weight $2^{n-i} - 1$. So the maximum length of path in a n-dimensional butterfly cloud graph is n. Let

$P = x_0 x_1 \ldots x_n$ be the path. Then $x_i \in$level i where $i = 0$ to n. Hence $I(C) = (2^1 - 1) + (2^2 - 1) + \ldots + (2^n - 1) = (2 + 2^2 + \ldots + 2^n) - n = 2(2^n - 1)$. ■

Proposition 6.8. For every cloud graph C, $1 \leq I(C) \leq n(n-1)$.

Proof. For $(i, j) \in A(C)$, $w(i, j) \geq 1$. Thus $I(C) \geq 1$. Let $P = (x_0, x_1, \ldots, x_l)$ be a path of maximum l in C. $I(C) = \sum_{i=0}^{l-1} w(x_i, x_{i+1}) \leq \underbrace{n + n + \ldots + n}_{l} \leq l.n \leq (n-1)n (\because l \leq n - 1)$. Hence, $1 \leq I(C) \leq n(n-1)$. ■

Theorem 6.9. Let C be a cloud graph. Then $w(x, y) = n, \forall (x, y) \in A(C)$ if and only if C is strongly connected.

Proof. Let $w(x, y) = n, \forall (x, y) \in A(C)$. Then $|A_{xy}| = n$ where $A_{xy} = \{z/\exists$ a directed walk starting from the arc (x, y) and reaches $z\}$. If the directed walk $x - z$ contain all the vertices of C then there exists a closed spanning directed walk. Therefore, C is strongly connected. If not, let $P = x - z$ directed walk. Then $\exists w \in V(C)$ such that $aw \in A(C)$ where $a \in V(P)$. Since $A_{aw} = n$, $\exists$ a directed walk $a - w$. Let $P' = a - w$ directed walk. If $P \cup P'$ contains all the vertices of C then the result holds. Proceeding like this there exists a closed spanning walk in C. Hence, C is strongly connected. Conversely, let C be strongly connected. Then, for every pair of vertices x and y, there is a $x - y$ path and $y - x$ path in C. Let $(x, y) \in A(C)$. Then x reaches y and y reaches all the remaining $(n-1)$ vertices. Thus $w(x, y) = n, \forall (x, y) \in A(C)$. ■

Corollary 6.10. A nontrivial tournament cloud graph C is Hamiltonian if and only if $w(x, y) = n, \forall (x, y) \in A(C)$.

Proof. Follows from Theorem 3.1 and 6.9. ■

Theorem 6.11. For every nontrivial strong tournament cloud graph C, $I(C) = n(n-1)$.

Proof. Let C be a nontrivial strong tournament cloud graph. Then $w(i, j) = n \; \forall (i, j) \in A(D)$ and by 3.2, C contains a Hamiltonian path. Since C contains a spanning path and weight of each edge in that path is n, $I(C) = n(n-1)$. ■

Theorem 6.12. Let C be a cloud graph. Overall orientations of C, $I(SCC) \geq I(USC) \geq I(WCC)$, where SCC, USC, WSC are strongly, unilaterally, weakly connected cloud graphs respectively.

Proof. Since every strongly connected cloud graphs are unilaterally connected and every unilaterally connected cloud graphs are weakly connected, we get our result. ■

Note 6.13. The weight of each arc (x, y) play a vital role since it indicates the number of clouds x can communicate. If an arc has weight n then that arc is more significant than the others. The more weighted arc is the strongest arc.

6.1. Algorithm for Closeness Centrality using Reachability

```
(1) D[i][j] = Distance Matrix (d_{ij})_{n x n})
(2) set i = 1
(3) if i<=n do 4 through 17
      otherwise STOP
(4) j=1, sum = 0
(5) if(j<=n) do 6 through 15
      GOTO 16
(6)  lx = D[i][j]
(7)  if(lx != 0) && (lx != NULL) do 8 through 14
      otherwise GOTO 15
(8) count=1
(9) set k=j+1
(10) if (k<=n) do 11 through 13
 otherwise GOTO 14
(11)  lx = D[i][k]
(12) count++, D[i][k]= NULL
(13) k++, GOTO 10
(14) compute sum as sum + (1 / lx )*(count / n-1)
(15) j++, GOTO 5
(16) Cccr[i] = sum
(17) i++, GOTO 3
```

Time Complexity

Since the complexity of distance matrix algorithm is $O(n^3)$, the complexity of the proposed algorithm can't be less than $O(n^3)$. The first/outer iteration steps from step 4 through 17 require $O(n)$ times. For each step of first iteration, the algorithmic (inner iteration) steps from 6 through 16 require $O(n)$ times. Thus the two iterative steps need $O(n^2)$ times. The time complexity of innermost iterative steps from 11 through 13 need atmost $O(n^2(n-2))$ times, because the distance zero and NULL cases were excluded. Therefore the total time complexity of the proposed algorithm is $O(Cccr) = O(n^3)$. Thus the algorithm categorized under polynomial time. Since the execution time of the algorithm completely depend on size of input that is dimension of distance matrix and the execution time of suggested approach is $O(n^k)$ where n is the size of input and k is constant.

Implementation and Performance of the algorithm

Our C_{ccr} algorithm is implemented in Java. Thus the approach of algorithm executes faster. In evaluating performance, it is important to distinguish between the performance of the implementation and the performance of the algorithm. Performance of the implementation is measured by the total time required for the program to process a pair of graphs [11].

7. Conclusion

Clouds can provide users with a number of different benefits. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. It motivated us to identify and characterize the dominant clouds and dominant users in cloud computing using Graph theoretical strategies.

Acknowledgments

We are grateful to the Ministry of Human Resource Development, New Delhi for providing financial support to establish a National Centre of Excellence in Statistical and Mathematical Modelling on Bioresources Management in Thiagarajar College.

References

[1] Georgiana Avram (Olaru), M., Advantages and Challenges of adopting cloud computing from an enterprise perspective. Procedia Technology, 12, 529–534 (2014).

[2] Aggarwal, K., Cloud Computing: The Future of Computing. Journal of Engineering, Computers & Applied Sciences (JEC & AS), Volume 2, No. 1, 14–18 (2013).

[3] Pandiaraja, D., Arun Nagendran, N., Chandrasekaran, S., Aditya, R. P., Graph theoretical modelling and analysis of fragile honey bee pollination network. Current Science, Volume 107, No. 12, 1988–1996 (2014).

[4] Chandrasekaran, S., Arun Nagendran, N., Krishnakutty, N., Pandiaraja, D., Saravanan, S., Kamaladhasan, N., Kamalakannan, B., Disposed paper cups and declining bees. Current Science 101, 1262 (2011).

[5] Zegzhda, P. D., Zegzhda, D. P., Nikolskiy, A. V., Using graph theory for cloud system security modeling. In: MMM-ACNS'12 Proceedings of the 6th international conference on Mathematical Methods, Models and Architectures for Computer Network Security: computer network security, 309–318 (2012).

[6] Bondy, J. A., Murty, U.S.R., 1976. Graph Theory with Applications, Elsevier, Amsterdam.

[7] Chartrand, G., Zhang, P., Introduction to Graph Theory, Tata McGraw Hill Pub. Co. Ltd., New Delhi, (2006).

[8] Green, O., Bader, D. A.: Faster Betweenness Centrality Based on Data Structure Experimentation. Procedia Computer Science, 18, 399–408 (2013).

[9] Wang, G., Shen, Y., Luan, E., A measure of centrality based on modularity matrix, Progress in Natural Science, 18, 1043–1047 (2008).

[10] Gavoille, C., Peleg, D., Perennes, S., Razb, R., Distance labeling in Graphs. Journal of Algorithms, 53, 85–112 (2004).

[11] Schmidt, D. C., Druffel, L. E., A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices. Journal of the Association for Computing Machinery, Volume 23, No. 3, 433–445 (1976).