

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A New Type of Center Data Structure in Cloud Computing

¹Guo Xiaohui, ¹Wei JianYu, ²Wang Beibei and ²Liyongqing

¹Tobacco GuangXi Industrial CO., Ltd. China

²Bei Hu Nan Road No. 28, NanNing, GuangXi, 530001, China

Abstract: This study analyzes and summarizes new characteristics of Cloud Computing data center, a new network structure in the design of cloud computing data center is proposed, the concepts of Cloud Computing and data center are introduced. It analyzes three important issues deeply, the scalability and green energy issues of the data center are analyzed and it makes full consideration of new characteristics of Cloud Computing data center and presents a new data center network structure according to famous Koch Curve, the snowflake structure. It makes full account of the data center's scalability and low proportion of switches and servers and can achieve routing within a shorter average path and smaller network overhead. In the study of snowflake structure of cloud computing data center network building methods are proposed and its properties are discussed, the simulation and experiments are performed to validate the performance of the structure.

Key words: Cloud computing, data center, network structure, snow-flake structure

INTRODUCTION

The structure design of cloud computing data center network is the important factor that must be considered. It can guarantee the high extensible ability r and high utilization of resources in the cloud computing data center (DeCandia *et al.*, 2007; Irwin *et al.*, 2004). This study is aim for the extensibility in the structure of cloud computing data center and saving needs of the energy and the new data center should have the new characteristics, so it adopt already well-known data center structure, on the basis of the Koch curve, proposed the snowflake structure which is a new data center network structure, In the structure we take the scalability of data centers into consideration, under the premise of lower quantity proportion; it can realize the routing mechanism in a relatively short average path between nodes and also with a smaller network expense (Dogan and Osgunger, 2002; Erickson *et al.*, 2009; Pinheiro *et al.*, 2007). The snowflake structure of cloud computing data center network constructing method and its properties are discussed in the study, the structure design between nodes in the routing protocols, algorithms and experiment simulation are performed in order to verify the performance of the new structure (Foster *et al.*, 2008; Germain-Renaud and Rana, 2009).

SEVERAL TYPICAL STRUCTURES OF DATA CENTER NETWORK

There are representative of the data center network architecture, such as a Fat-Tree DCell, the BCube and the

VL2 etc., it will analyze these structure and their respective characteristics (Gilbert and Lynch, 2002).

The server can be divided into k groups of Fat-Tree structure, each subgroup contains two layers of port number for the $k/2$ k switches, $k/2$ of the switch port is connected to the $k/2$ hosts, respectively and each layer and switches are connected; Core layer ($k/2$) 2 core switch port number is k , they most can support the $k^3/4$ hosts as shown in Fig. 1. Fat-tree abandoned traditional data center and adopt the dedicated switch mode, switch to commercial Ethernet switches, improves the ratio of larger. It can contain tens of thousands of servers in the data center and provides high aggregate bandwidth, don't need to modify the host network interface, operating

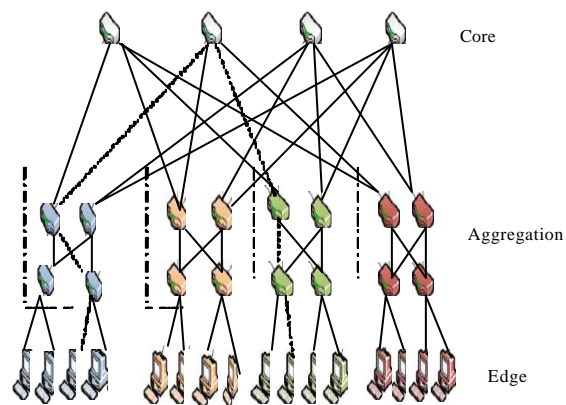


Fig. 1: Fat-tree structure in the cloud computing system

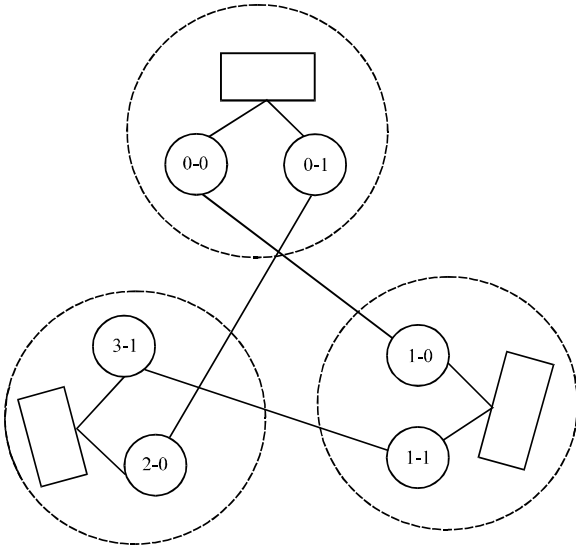


Fig. 2: DCell_structure of the cloud computing system

system, you can build and compatible with Ethernet, TCP/IP communication protocols as well. Fat-link is equal to the number of tree layers, which makes maximum flow rate equal to the maximum throughput of the core layer and it adopts the two levels of routing table and the certain link error detection mechanism can realize fault tolerant routing.

Removes the fat-tree structure tree structure upper link the limitation on the throughput and can offer multiple parallel links for internal communication between the nodes. But fat-tree extensibility is limited to the core switch port number, which are frequently used at present is 48 core switch port with 10G capacity, in the three layers of tree structure, it can support 27648 hosts. In the long run, the scale of data center is unable to meet the application requirements, so the fat-tree exists defects with the lack of extensibility. Fat-tree another disadvantage is the difference in fault tolerance, the manifestation of lacking the ability to handle switch fault and fault tolerant routing protocol is not strong. Research of MSRA show that fat-tree was highly sensitive to the low level switch failures and it can seriously affects the performance of the system. Because fat-tree is still the tree structure, so it has the defects of tree structure essentially.

D-cell is a recursive definition of network structure, the use is located in the $i-1$ layer D-cell and construct the D-Cell the i th layer. When the size of the node degree is increases, the D-Cell is close to 2 exponent index extensions. D-cell₀ usually contains 3 to 8 unit servers and they are connected through the micro-switch just as shown in Fig. 2. D-Cell has better fault tolerance

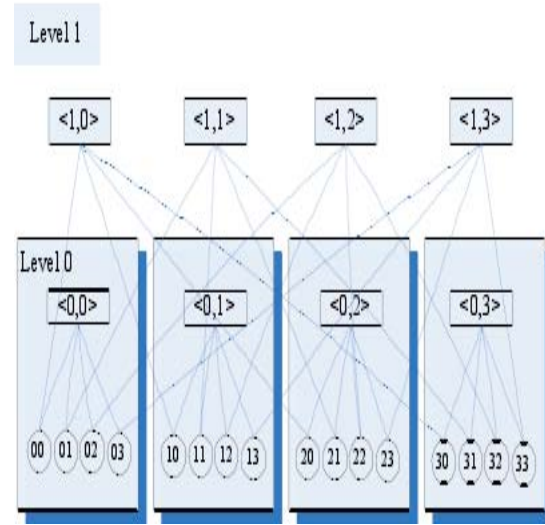


Fig. 3: BCube_1 structure of the cold computing

capability; the distributed protocol implementation is close to the shortest path routing. D-Cell also can provide various services with network capacity which is higher than the traditional tree structure. Additional D-Cell can increase the extension and in the case of incomplete structure it also can show the performance. Although DCell₀ is small but DCell can support the larger number of servers, as known that when DCell₀ contains 6 hosts, DCell₃ can support 3.26 million servers.

The server can realize the D-Cell, link or can be switched under the condition of serious fault and it achieves good routing performance. D-Cell also has some shortcomings; first of all, the complete graph connections may lead to enormous cost, size and the actual link connection and maintenance difficulties. Second, the uneven distribution of flow at different levels in the DCell, level 0 afford too much traffic and it will seriously affect the throughput capacity. Finally, because the DCell adopts the server perform routing, which enlarges the network latency and the routing protocol is not suitable for find the shortest path in link failure and the network delay will be larger.

Bcube is DCell modular version, its connection is outside BCube₀ hierarchy connection and it is realized by using the micro switch as shown in Fig. 3. The number of units in each layer is different from the DCell, if there has n in the first k DCell _{$k-1$} in DCell layer, The BCube at any level will the same unit number n , so it can easily obtained that BCube _{k} has n^{k+1} servers and n is the number of servers in a BCube₀. Switches as the medium of connection make BCube has many redundant paths, this can ensure that the fault tolerant routing and

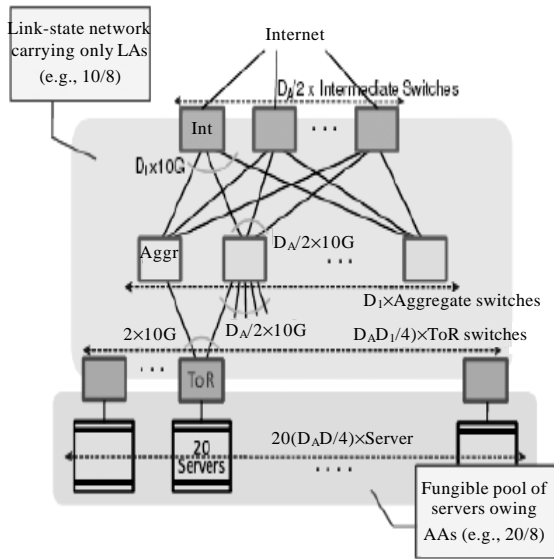


Fig. 4: The structure of VL2 in the cold computing system

convenient modular connection and routing speed are faster than DCell. Another BCube adopts BSR (BCube Source Routing) in selecting path and the path adaptive protocol, which can develop the network and realize the shortest parallel paths and reliable data transmission.

BCube can execute efficiently the One to One, One to All, One to Several, All to All types of communication without bandwidth limitation and it supports GFS and MapReduce applications. BCube connection method and the recursive structure allow data centers to modular construction and the realization is the good. Under the condition of incomplete structure the BCube has better performance than DCell. Experiments show that when given 2048 servers, in Fat-Tree, DCell and BCube structure are not complete, the network throughput and fault-tolerant performance of BCube are the best one and this is because the incomplete structure of BCube adopts the complete structure of switch strategies. The deficiency of BCube is on the extensibility, in the BCube, when in $k = 3$, $n = 8$ (k is BCube levels, n is BCube_0 server number), it only can support 4096 servers, which is smaller than the DCell.

As shown in Fig. 4, VL2 is an extensible and flexible data center network structure, it can be able to support large scale data centers, provide a balanced high bandwidth communication performance for servers, the performance isolation between services and layer 2 Ethernet semantics. Semantics refers to the second tier 2, all of the virtual domain into a unified domain, at this level all hosts in the same domain. VL2 structurally changed is

the connecting way of layer 3 switches, the use of special virtual layer 2 protocol implementation; And other structural changes in the physical connection is a whole. In addition, the said address and routing protocol in the VL2 is even more important, it is directly related to the realization of virtual layer 2. In the VL2, the VLB (Valiant Load Balancing) routing is adopted to balance various runoff, VL2 among each switch is set to the same IP, adopt the way of random choosing an intermediate switch routing.

SNOWFLAKE STRUCTURE

This section it put forward the method of building data center network structure of the snowflake and give out some of their special properties. It call be called the snowflake structure, because according to the construction of the structure of the Koch curve and its shape is like the Koch snowflake.

Snowflake structure and its construction method:

Snowflake structure contains two parts: the server and the micro switches. The Salt DCell and BCube structure, they both are recursive method is adopted to define their structure. In the snow structure, it adopt the method of recursive definition, also at level $(n-1)$ if a snowflake structure on the basis of adding, add a level 0 snowflake structure to constitute the n -tier structure of snowflakes, Several methods will be described in detailed later.

First of all, it take the fixed structure way in the modular structure, which will be helpful to the structure of modular connection; Second, when the n -tier structure is not be extend completely, it expand $(n+1)$ level structure is relatively easy. At this point, if the system discovers that the n -tier structure did not extend completely, it can be continued to supplement the n -tier structure of incomplete without changing $(n+1)$ level structure, it is helpful to the expansion of the structure. Snowflake structure construction method can be described in detailed as below.

Level 0 snowflake structure (Snow_0) is composed of a micro switch and a number of servers. According to data center structure, the number of server is at least 3 and not more than 8. As shown in Fig. 5, which takes the three servers for example ($k = 3$) and it connects the three servers with the micro switch port, which is shown in Fig. 5, if the theoretical position of the server is changed and the three servers are connected directly with dotted line. The three dotted lines are not the actual connections of the servers, it can be called virtual connection, there is no actual connection and it is just for the convenience.

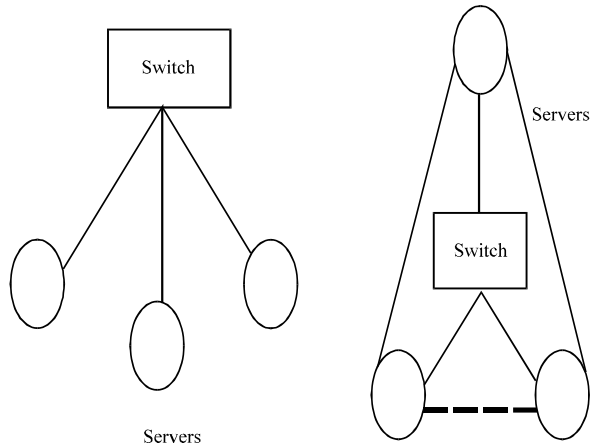


Fig. 5: The structure of Snow_0 when $k = 3$ in the cold computing system

Based on Snow_0, the three virtual connections can be canceled which is as shown in Fig. 5, every virtual connections are disconnected, namely that adding a 0 level snowflake structure and add the new level 0 structure of micro switches, respectively and the virtual connection nodes are connected on both ends. Compare with the virtual connection, this connection is called real connection, or disconnect the virtual connection rebuilt the actual connection.

Real connection is real connection; here is the newly added structure of the switch with the original structure.

Server connection: So, the Snow_1 snowflake structure can be obtained, which is as shown in Fig. 6.

Because it is add disconnect on the virtual connection level 0 snowflakes, the state of the virtual connection has become real connection at this time namely that the one virtual connection has been changed into two physical connections.

The special note is that the connection is not just a real connection, it is reflected by the virtual connection to real connection, the connection reflects the change of states, so in Snow_0 server which is connected to the switch, three places are not as a real connection, although it is a real connection, but there is no state change. Through careful observation, it can be found that, disconnect virtual connection is to add the structure and there is a difference between Snow_0 level 0, it lacks a virtual connection and it is not really a Snow_0 structure. In order to distinguish between the two connections, the later broken open virtual connection and keep adding and which is lack of a virtual connection structure of level 0, it is the cell to distinguish Snow_0.

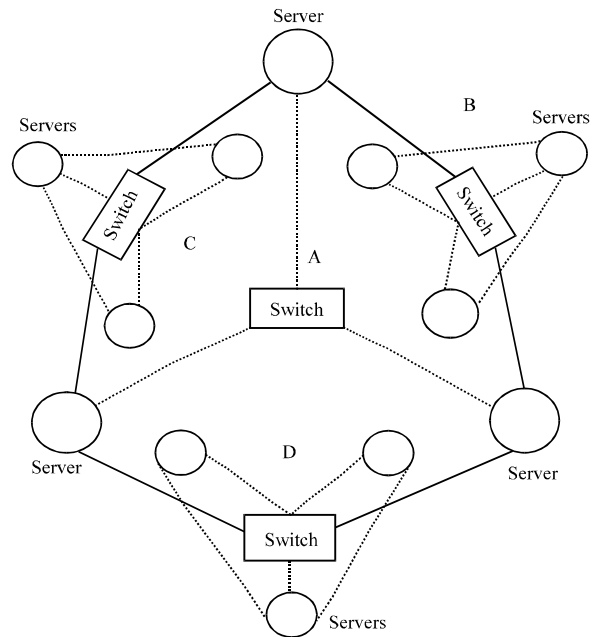


Fig. 6: The data structure of Snow_1 when $k = 3$ in the cloud computing system

In the Snow_1 as shown in Fig. 7, it contains six real connections and six virtual connections. Snowflake structure in each level, always keep off the next higher level, which contains all of the real and virtual connections, if Cell is added, a new advanced structure can be formed.

Snowflake structure performance analysis: In this section, the performances of the snowflake structure from the network exchange are analyzed. i.e., the capacity and bottleneck link, time delay, green energy-saving four.

Network exchange capacity: If it is assumed that the bandwidth is AMB/s , the network width between switch and server is about BMb/s and generally the A is larger than B , when the bandwidth between the two servers is limited to the minimum bandwidth value of the path. Snowflake structure can maintain a low number of switches under the premise of good extensibility. In addition, in snow flower and the structure, the basic server is connected to the switch and there are almost no existences of server connecting with the server directly.

So, that it can ensure high data transfer rate. Therefore, snowflake structure is suitable for the data when there needs frequent communication between nodes.

Bottleneck link: Our bottleneck can be divided into two kinds: Bottleneck node and path

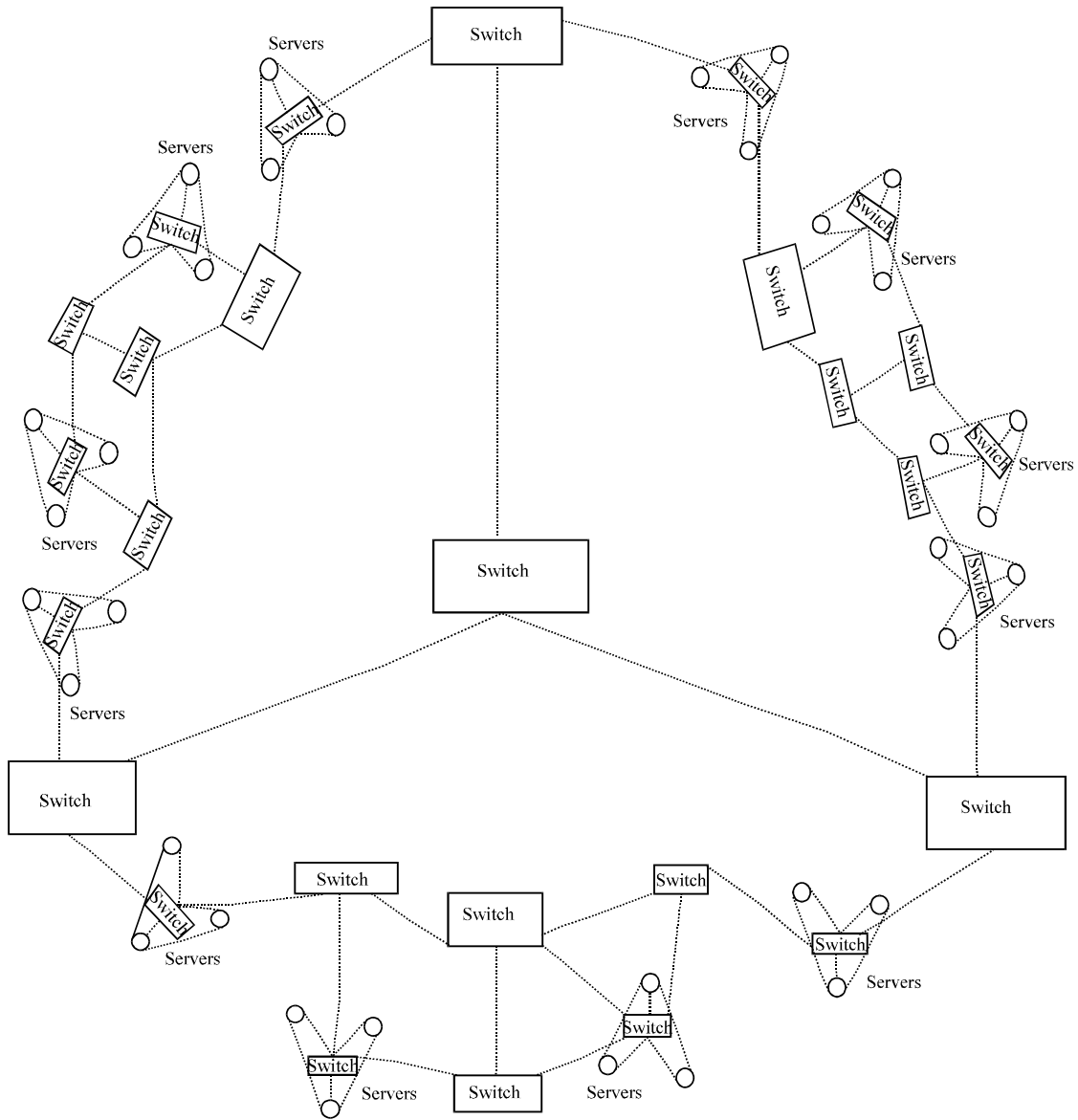


Fig. 7: The improved Snow_1 structure in the cloud computing system

bottleneck. Bottleneck node refers to a node (server or switch) due to the large data traffic bottleneck.

The more the easier to be the bottleneck node level low. So some server in the hierarchy can improve for switches; the switches can be replaced with high throughput, such as signs in the switch, in order to ease the pressure on data transmission.

Path bottleneck refers that one and only one path between the two subnets, when the path is disconnected and the two subnets loses connection to form a bottleneck; the path can be called path bottleneck. Any two servers will prove, Snow_n contains parallel paths

and at least two and not more than $2n$ in the theory, Due to this theory, there is no the problem of path bottleneck in snow structure.

Time delay: RTT (Round Trip Time) is the total delay time from the sender to send data to and receive from reception to the sending end and the confirmation the receiver send confirmation immediately after receipt of the data. When between two points RTT value of the larger population of the time delay between the two; On the other hand is the time delay between two points is smaller. It can be seen from the definition of RTT, RTT is proportional to the

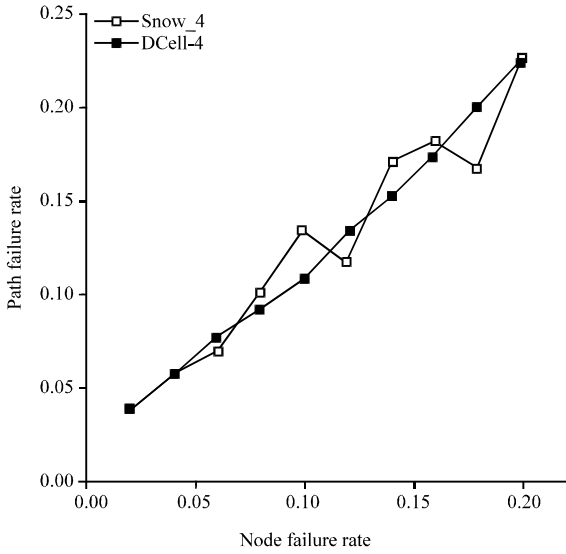


Fig. 8: Change of average shortest path length with different failure rate of link

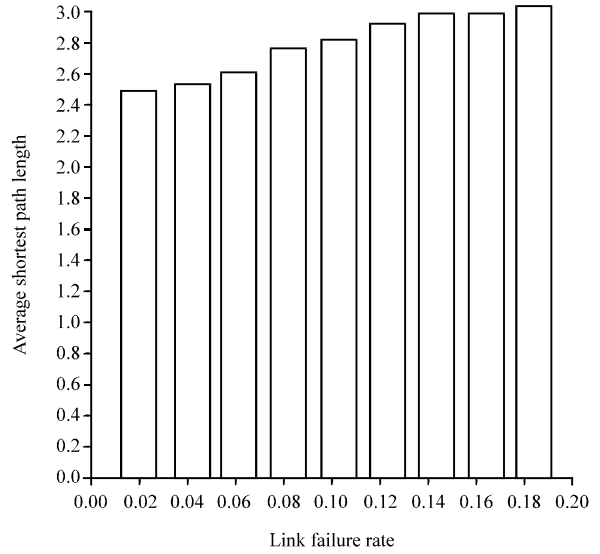


Fig. 9: The Path failure rate between Snow_1 and DCell_4 under different node failure rate

shortest path between two points. When the distance between two points a long, RTT is larger; On the other hand, the RTT is smaller.

So, it can conclude that the longest shortest path to jump $(2n+1)$. Therefore, delay time is $O(n)$.

SIMULATION AND EXPERIMENT RESULT ANALYSIS

In this section, the simulations are adopted to evaluate the snowflake structure performance. Simulation program in written in Java language, development tools is Eclipse 3.5.0, run environment is on Lenovo computer with 4 cores 2.7 GHz cup.

Situation 1: When there is no node failure, the total number of nodes at different levels and average shortest path length.

Situation 2: When there exist the node failures, When $k = 3$, the detailed node is set with failure ratio, the test along with the node effect the increase of the proportion in the Snow_4 and average shortest path length will be changed. As it can be seen from the Fig. 8, when a node failure is kept at 0.05-0.05, the average path length fluctuation is not big and the value is between 2.6 and 2.7. When $k = 3$, path of failure rate in the Snow_4 and e DCell are the contrasted through experimental results as shown in the Fig. 9, it is path failure comparison between Snow_4 and DCell_4.

Table 1: Average shortest path of each node to the node Snow_0 when there no node fails

Snow levels	Snow 2	Snow 3	Snow 4
Total No. of servers	48	192	768
Total No. of switches	16	64	256
No. of nodes	47	191	767
Inaccessible node No.	0	0	0
Shortest path to the combined	80	416	2048
Average shortest path length	1.702	2.178	2.670

It can see from the Fig. 9, the failure path of the Snow efficiency around DCell fluctuate, the difference is not very big. When under the no-node failure conditions, in the Table 1, it can be found that when $k = 3$, the average distance from the Snow_0 server to the rest of the nodes is the shortest. As the node failure situation is not considered, therefore number of inaccessible node is the zero. Through the average shortest path length, it can found in Snow_4 range, average path can be maintained. Situation 3: When there exists a link fails, Here the link failure refers to the link between the disconnect switch and server, or switch failure, the failure is not the server itself.

As can be seen from the Fig. 9, with the increase of link failure rate, average shortest path length changes are not big, the link failure rate is within 0.2, average shortest path length between 2.6 to 2.9.

When $k = 3$, some test of the path of failure rate as the link failure rate increases are made, in the Snow_4. some comparisons with the DCell results in the experimental are also given out. As shown in Fig. 10, it is path failure comparison between Snow_4 with DCell_4. It can be found that the Snow path failure rate is stable which is always maintained between 0.001 0.004.

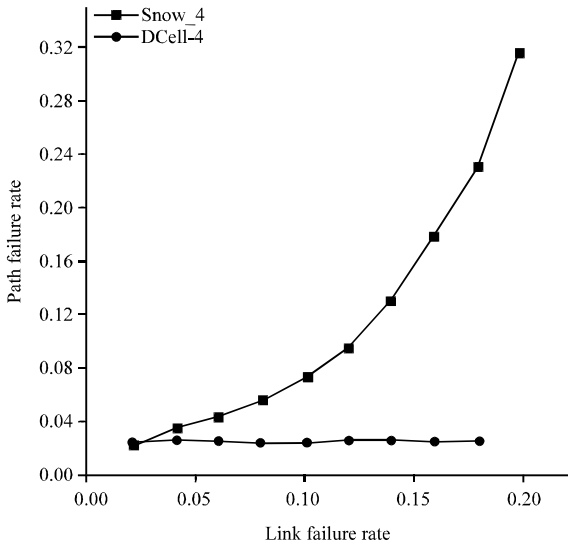


Fig. 10: Path failure of Snow_4 and DCell_4 under same link failure rate

Table 2: Comparison of characteristic between DCell, BCube and Snow

Network structure	DCell	BCube	Snow
The proportion of exchange with the server	High	High	Low (advantages)
Node failure to the path to failure	Better	Better	With a DCell effect
Link failures with path failure	Better	Better	Compared with the DCell is poor
Network throughput	Network expansibility is good, Have higher network throughput	Compared with the DCell is poor	Low-level server and switches may become the system bottleneck (disadvantages)

Finally, the comparison between DCell and BCube Snow are presented, the results are as shown in the Table 2.

CONCLUSION

Karjoth, (2003), Yu and Vahdat (2002), Higgs *et al.* (1997), Shan *et al.* (2003) and Foster and Kesselman (1997) have made related research, but the research results are not suitable in the cloud computing and the new data structure in cloud computing is very necessary.

In order to meet the scalability in the structure of cloud computing data center and green energy demand, in the study, the new data center network structure based on the famous Koch curve, the structure of snowflakes is put forward. In the structure, the scalability of data centers are considered, under the premise of guarantee switches with the server, it can realize smaller network expense in a relatively short average path between nodes within the routing mechanism.

In my opinion the center data structure such as the snowflake structure of cloud computing data center and network building methods, can improve the performance of the cloud computing. And the research is necessary and will enhance the speed and other parameters of the cloud computing. The security of the proposed structure in the cloud computing is very well and the security issues is not the research content of the study, but from the related research progress, it can be found that it is not the problem.

REFERENCES

DeCandia, G., D. Hastorun, M. Jampami, G. Kakulapati and A. Lakshman *et al.*, 2007. Dynamo: Amazon's highly available key-value store. Proceedings of the Symposium on Operating Systems Principles, October 14-17, 2007, Stevenson, WA., pp: 1-6.

Dogan, A. and F. Osgunger, 2002. Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing. IEEE Trans. Parallel Distrib. Syst., 13: 308-323.

Erickson, J., S. Spence, M. Rhodes, D. Banks and J. Rutherford *et al.*, 2009. Content-centered collaboration spaces in the cloud. IEEE Internet Comput., 13: 34-42.

Foster, I. and C. Kesselman, 1997. Globus: A metacomputing infrastructure toolkit. Int. J. High Perform. Comput. Appl., 11: 115-128.

Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. Proceedings of the Grid Computing Environments Workshop, November 16, 2008, Austin, Texas, pp: 1-10.

Germain-Renaud, C. and O.F. Rana, 2009. The convergence of clouds, grids and autonomies. IEEE Internet Comput., 13: 9-9.

Gilbert, S. and N. Lynch, 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, 3: 51-59.

Higgs, R.E., K.G. Bemis, I.A. Watson and J.H. Wikel, 1997. Experimental designs for selecting molecules from large chemical databases. J. Chem. Inform. Comput. Sci., 37: 861-870.

Irwin, D.E., L.E. Grit and J.S. Chase, 2004. Balancing risk and reward in a market-based task service. Department of Computer Science, Duke University, Durham, USA. <http://www.cs.duke.edu/nicl/pub/papers/hpdc04.pdf>

- Karjoth, G., 2003. Access control with IBM Tivoli access manager. *ACM Trans. Inform. Syst. Secur.*, 6: 232-257.
- Pinheiro, E., W.D. Weber and L.A. Barroso, 2007. Failure trends in a large disk drive population. *Proceedings of the 5th USENIX Conference on File and Storage Technologies*, February 13-16, 2007, SanJose, CA, USA.
- Shan, H., L. Olikier and R. Biswas, 2003. Job superscheduler architecture and performance in computational grid environments. *Proceedings of the Conference on Supercomputing*, November 15-21, 2003, Phoenix, Arizona, USA.
- Yu, H. And A. Vahdat, 2002. Minimal replication cost for availability. *PODC*, Monterey, California, USA.