5, results of the proposed approach presents and results of the simulation will be expressed.

2. Task Migration

To a better understanding of the proposed approach, the existing literature in task migration area is briefly studied in this part of the paper. Then the way different kinds of task migration methods perform will be described.
Migration of virtual machines: Using virtualization technology today most of data centers increase their system efficiency via increasing the number of customers. This technology provides a virtual environment and factually presents the physical machine with all its modes and data in form of a logical file. Therefore, virtual machines can be transmitted from one place to another. This transmission is called virtual machine migration. One of widely used environments of this technique is data centers which in load and requests amount are dynamically changing.
Different types of task migration are classifies like this:

- Live Migration: In this method, source machine receives users' request without delaying (without blackout Machine) and transmits them to destination machine so that services are provided even during migration process which is one of the advantages of this method.
- Non-live Migration: First virtual machine stops completely in the source and then all processor modes and memory pages are transmitted to destination machine. After receiving all modes and pages from the last mode of machine stored before transmission, destination machine starts up.
- Task Migration in Mesh Networks: In mesh networks, each node is a computer that can include processor, memory, and communicational channels to be in communication with its other neighbor nodes. When a task enters this system, all processors will be allocated to that. Figure 1 shows task migration from one sub network to another sub network. In 3D Mesh network system M(D,W,H) encompasses N=D*W*H nodes distributed on the network. Each processor in this system has some numbers. For instance, a node with the number of M(D,W,H) includes {(0,0,0),….(D-1,W-1,H-1)} processors. SM(d,w,h) mesh sub networks consider M as {(x1,y1,z1) ,….(x2,y2,z2)} which in (x1,y1,z1) is its coordinates of the bottom left corner and (x2,y2,z2) represents its coordinates of top right corner. In this case the equations of h=z2-z1، w=y2-y1، d=x2-x1 are dominant there. It is assumed in this paper that Worm Hole Switching method and the next regular routing method such as XY routing have been used. In this method, every node is able to receive and send a message simultaneously [4].

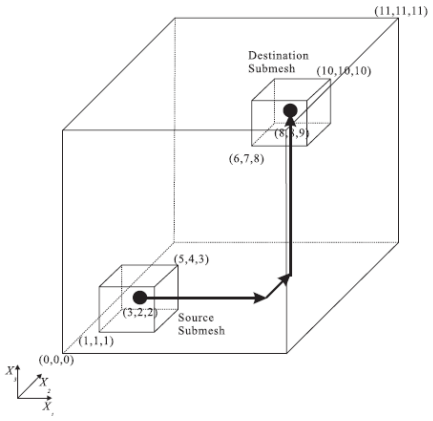

Fig .1 Task migration from one sub network to another

3. Related Work

Task Migration in multiprocessor networks such as cloud computing are similar to processor migration in distributed systems. Processor migration has been implemented in multiple systems. Also various algorithms have been presented for task migration. These algorithms are totally made up of migration beginning stages, stop running task in source processor, task transmission to destination processor, start running task in destination, task running in destination processor, and removing remained information from source processor. Continuing this part, types of presented migration algorithms will be studied [5].

- Pre Copy: In this algorithm, running processor will not be stopped during its transmission from source machine. pre copy algorithm also has three stages as mentioned bellow [6 and 7].
  - First stage: Repetition of sending data relevant to the virtual machine like memory pages and various kinds of processor modes by source machine to destination one.
  - Second stage: In this stage, the virtual machine goes to suspension mode and then other pages changed during transmission in the first stage and also processor modes necessary to start up destination machine will be sent.
  - Third stage: Destination machine starts up just from the last mode stored in the source before suspension. The point in this algorithm is that the less changes in pages in the first stage, the less time in the second stage the service is inaccessible.
- Post copy algorithm: First in this method all processor modes and the least needed information are sent to destination machine which results in destination machines start up. Then sending memory pages from source to destination will be started. This method

assures that each page will be sent to destination machine only for once so that overhead out of one page's multiple sending found in pre-copy method will be omitted [8 and 9].

- Three Phase Migration (TPM): TPM algorithm has the least suspension time for transmission of the whole system with its all modes. In fact TPM technique is the same with disc pre-copy technique which is also able to displace virtual machine's disc [10].
- CR/TR Motion Algorithm: This algorithm is used for operating synchronization based on processing power in destination machine. It means that log will be sent instead of sending pages. Therefore, sent data amount during two machines' synchronization will be considerably decreased [11].
- Migration algorithm dependence of ware: this algorithm is used when a virtual machine has no interaction with the world outside so that it can transmit a part of data to destination machine just with one time sending. There is a part in this algorithm called administrator which the decision of one or several time displacing processes should be done is made based on its information.

4. The Propose Approach

Presented approach is a combination of Yu routing and post-copy migration. Factually, the results of assessments in the next parts of this paper show that the proposed approach has the advantages of both former methods.
Performance of the proposed approach: Using Yu method, the proposed approach first provides the possibility of migration of several sub tasks in parallel through different paths. Consequently there will be maximum parallel and simultaneous transmission of tasks so transmission time of sub tasks and their migration from one virtual machine to another will decrease. Also using method of post-copy reduces the amount of overhead out of repetition of applied changes on memory pages which results in removal of sending data from source machine to destination. In the proposed approach, each computer is considered as a node that can include processor, memory, and communicational channel. With the entrance of any new task, a sub network of nodes is getting allocated to that at the same time routers manages.

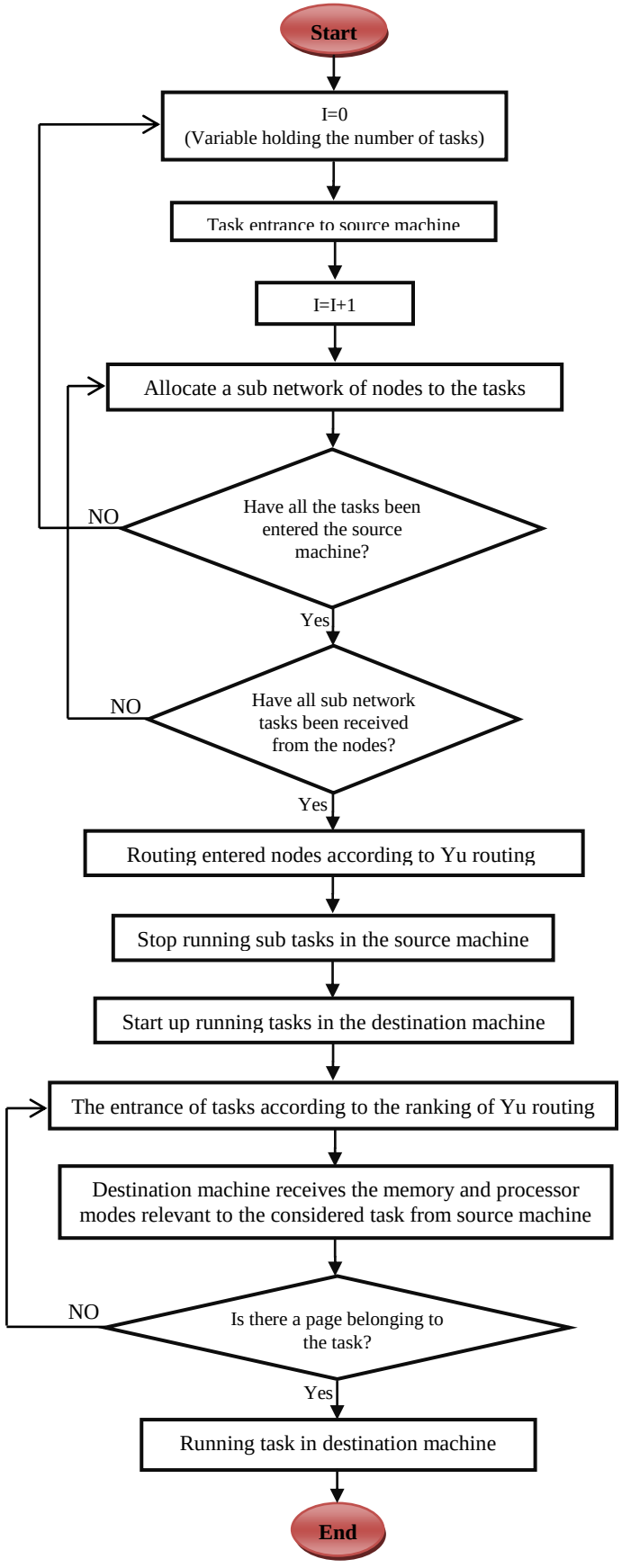

Fig .2 The proposed method flowchart

4-1. The Algorithm of Proposed Approach

Stage1: First sub tasks running in different nodes of source machine are sent to destination sub network based on Yu routing algorithm.
Stage2: According to post-copy algorithm, processor modes will be sent to destination machine. This process causes destination machine start up.
Stage3: Memory pages will be sent from source machine to destination.
Stage4: If one memory page was not found, there will be a delay in destination and that page will be requested from source machine.
Stage5: This stage is the end of running transmitted sub task from source machine to destination machine.

4-2.Comparing Performance of Migration Algorithms

To compare the performance of the proposed approach with the other mentioned Approach, the proposed Approach performance is studied over sent data amount, overhead amount, migration time, and quality of efficiency in this part in table 1.

Table 1: Comparing migration algorithm with the proposed method

Approach	*Pre Copy*	*Post Copy*	*Three Phase CR/TR Motion*	*Dependence-Aware*	*Proposed Approach*
Sent Data Amount	H	L	H	M	L
Overhead Amount	L	L	M	M	L
Migration Time	L	L	L	S	L
Efficiency	L	L	L	L	M

Where in table 1, latters H, M, L and S sequentially are stand for High, Medium, Low and Short.
Comparing the proposed approach to the other approaches shows that using the proposed approach reduces sent data amount, overhead amount, and also migration time. Therefore, considering the results of simulation in MATLAB environment, system will have higher efficiency using the proposed approach in accordance to the other ones in table 1.

4-3. Results of Proposed Approach Simulation

Considering weaknesses of past presented methods, to analyze services migration in the proposed approach and also to increase efficiency, three factors bellow have been considered: Transmission time, Network Traffic Overhead and Services Stop Time.
Results of this experiment done in four real environments of Bit Torrent, SPECWeb 2005, Kernel Compile and NetPerf have been demonstrated in figure 3. Vertical axis shows time measured in second. Actually it shows the whole migration time in the interval of time from the beginning of migration operation till the modes and information in both source and destination machines are fully synchronized. This time decreases because of reduction in transmitted data amount (there is no repetition in post-copy method). In figure 4, vertical axis represents time measured in millisecond shows the very little amount of sent data in the proposed approach. The amount of sent data is low because of no repetition in sending data during migration so the amount of sent data will be low and also data overhead reaches zero. But stop time is still high which can express the weakness of the proposed approach.

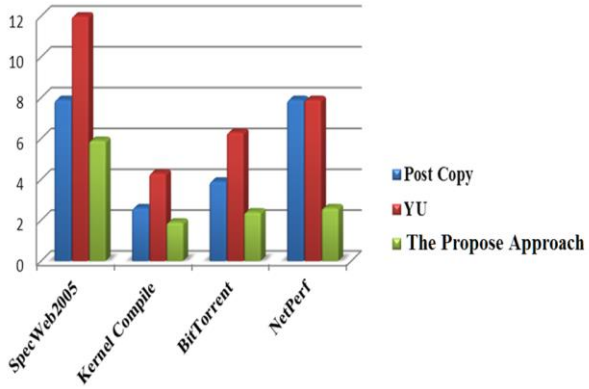

Fig. 3 Tasks migration time from source machine to destination machine

In the proposed approach which in combined algorithm has been also used, sub networks are divided into networks with the size of p×q×r. In this case, the number of partitions is to be calculated by equation (1).

$$[\frac{d}{p}] \times [\frac{w}{p}] \times [\frac{h}{r}] \quad (1)$$

Consequently in combined method, collected sub tasks migrate to the considered destination machine which in this phase the number of stages follows equation (2).

$$\text{Max}([\frac{d}{p}] \times [\frac{w}{p}] \times [\frac{h}{r}]) \quad (2)$$

Considering the results of figures 1 and 2, the proposed approach can largely solves the problem of transmission time and sent data amount.

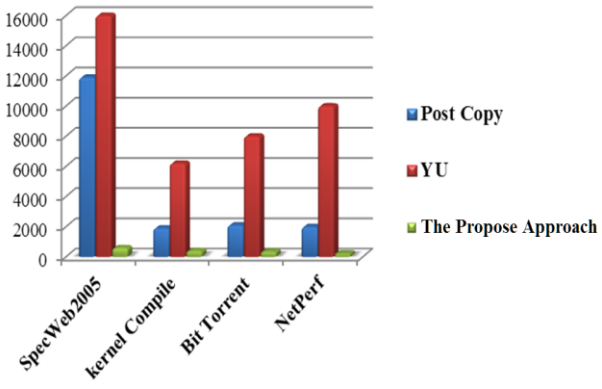

Fig. 4 Sent data amount

5. Conclusion

In this paper migration methods of virtual machines were studied. Also a new approach for task migration in cloud computing was proposed. Via the combination of post-copy algorithm in cloud computing and Yu algorithm in multi processor systems, the proposed approach can have the advantages of both mentioned methods. The proposed approach makes parallel and simultaneous task transmission possible through Yu algorithm which leads to the reduction of task transmission time. Furthermore, considering the results of simulation in MATLAB software, this can be observed that using post-copy method makes data overhead decrease but yet there will be a delay in providing service. Generally this can be concluded that in case of not using migration techniques, there will be unfavorable consequences such as increasing costs, unfavorable quality of service and consequently losing customers of cloud computing providers.

References

[1] T.Velte and A.Velte and R.Elsenpeter, "Cloud Computing, A Practical Approach, New York", USA, McGraw-Hill, 2010.

[2] B.P.Rimal and E.Choi and I.Lumb, "A Taxonomy and Survey of Cloud Computing Systems", in Fifth International Joint Conference on INC, IMS and IDC(NCM '09), Aug., 2009, pp. 25-27.

[3] C.P. Sapuntzakis, R. Chandra and B. Pfaff, J. Chow, M.S. Lam and M. Rosenblum, "Optimization the Migratin of Virtual Computers", In Proceeding of 5th USENIX Symposium on Operating Systems Design and Implementation (OSDI-02), Dec., 2002.

[4] G. Jong Yu, C. Yung Chan and T. Shi Chen, "Task Migration in Nd Wormhole-Route Mesh Multi computer" ,Journal of Systelns Architecture, Vol. 50, No 4, 2004, pp.177-192.

[5] K. Sammy, R. Shengbing and Ch. Wilson, "Energy Efficient Security Preserving VM Live Migration In Data Centers For Cloud Computing", International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 2, No 3, March 2012, pp.33-39.

[6] C. Clark and K. Fraser and S. Hand and J.G. Hansen and E. Jul, C. Limpach and I. Pratt and A. Warfield, "Live Migration of Virtual Machines", in Proc. of the Second USENIX Symposium on Networked Systems Design and Implementation (NSDI),Boston, MA, USA, May 2005.

[7] D. Cheriton, "The V distributed system",Communication of the ACM,3l (3), 1988, pp.314-333.

[8] J. Fejzaj1, I. Tafa and E. Kajo, "The improvement of Live Migration in Data Center in different virtual environment", International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 6, No 2, November 2012, pp.460- 463.

[9] M. Richmond and M. Hitchens. "A new process migration algorithrn", ACM SIGOPS Operating Systems Review, 31(l), 1997, pp.31 -42.

[10] Y. Luo, B. Zhang, X. Wang, Z. Wang, and Y. Sun, "Live and incremental whole-system migration of virtual machines using block-bitmap", in Proceedings of Cluster 2008: IEEE International Conference on Cluster Computing. IEEE Computer Society, 2008, pp. 99-106.

[11] H. Liu, H. Jin, X. Liao, L. Hu and C. Yu, "Live migration of virtual machine based on full system trace and replay", in Proceedings of the 18th International Symposium on High Performance Distributed Computing (HPDC'09), 2009, pp. 101–110.

[12] A. Nocentino and P. Ruth, "Toward Dependency-Aware Live Virtual Machine Migration", in Proceedings of the 3rd International Workshop on Virtualization Technology in Distributed Computing (VTDC 2009), Barcelona, Spain, Jun. 2009, pp. 59-66.