A Resource Scheduling Algorithms Based on the Minimum Relative Degree of Load Imbalance

Tao Xue and Zhe Fan
Department of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China
Email: xt73@163.com; 150398619@qq.com

Abstract—Dynamic virtual machine migration is a key technique technology in the cloud computing, an algorithm is proposed in this thesis, which is a dynamic resource scheduling based on minimum load imbalancing measurement. First of all, load balancer in scheduler judges the overload phenomenon; we need to migrate VM (Virtual Machine) on physical machine. Then, according to the records in load balancer, we calculate the load imbalancing measurement of the other entire working PM (physical machine) relative to the overload host. Choose the PM which has the minimum load imbalancing measurement as candidate host. Simulation results indentify this algorithm is able to realize efficient load balancing, acquire an optimal resource utilization of cloud computing system and maintain a low level of load imbalancing.

Index Terms—Cloud computing, virtual machine, resource scheduling, dynamic migration, load balancing, QoS

I. INTRODUCTION

Cloud computing [1] is a service model, which is calculated based on the shared network resource pool can be configured (including network, physical storage, applications, and services, etc.) and be convenient, on-demand access to. The emergence of cloud computing platform make computing resources like water and electricity and used by ordinary users, and there is no doubt that this technology will bring huge benefits to the majority of enterprises. Now, many companies have launched commercial cloud computing platform, such as EC2 [2], Google's App Engine [3], as well as Microsoft's Azure Amazon [4], and so on. This paper mainly studies IaaS [5] layer, and virtual machine placement is the key issue of IaaS, including the initial placement of virtual machine placement and dynamic migration. In cloud computing platform, computing nodes often result in the need to migrate the virtual machine deployed on the failed node because of overload or accidental failure, witch in order to ensure the reliability of the entire cloud platform runs. In cloud computing, how to perform live migration of virtual machines quickly and effectively, making cloud computing platform obtain higher high resource utilization, while ensuring the reliability of cloud platforms, it Is a hot and difficult point of current cloud computing research [6].

In recent years, many studies have focused on the cloud cluster load balancing problem [7], and present a number of efficient algorithms. In Literature [8], it put forward a method of migration of minimum cost of virtual machine placement. The algorithm consider the dynamic virtual machine resource allocation continuing change and migration cost constraints, and consider the cost of migration problems during the process in the initial configuration and migration of virtual machines, to achieve the ultimate high resource utilization and steady service quality. However, this algorithm had considered only one CPU computing resources, and cloud computing resources in the resource pool have a multi-dimensional attributes. In Literature [9], it proposed a virtual machine scheduling method based on genetic algorithms, which can achieve the desired load balancing state, and ensure minimal virtual machine migration overhead, but the algorithm ignores the impact created by data center resource utilization and Energy consumption closely related parameters, so the Consideration is not comprehensive enough. In Literature [10], it proposed a resource allocation mechanism, which is based on the migration time maintaining a minimum, while also minimizing the number of the number of virtual machines migrate virtual machine migration strategy. The algorithm takes into account the resource utilization CPU, RAM and network bandwidth, but did not consider the cluster physical resources. In Literature [11], it proposed a virtual machine live migration strategy, which combines the performance prediction algorithm. According to the average usage of CPU, memory, I/O and network bandwidth, the algorithm makes a series of judgments about the virtual machine migration. Such as whether the migration is triggered, which VMs should be migrated, and virtual machines are migrated to a witch physical machine, etc. However, this algorithm relies on statistical analysis of historical data heavily, therefore, it requires a relatively long time to obtain statistically valid data, and then start the optimization procedure. In Literature [12], for low load physical machine in cluster, it propose a virtual machine migration approach, which is to make these physical machines into sleep mode by migrating all the underutilized VMs on some physical machines. In order to ensure the load balancing of the

Manuscript received May 24, 2015; revised September 23, 2015.
This work was supported by the China's National Development and Reform Commission (NDRC) and High-tech industrialization projects (Shaanxi NDRC [2009] no. 1365); Xi'an Polytechnic University doctoral research start-up fund (BS0725).
Corresponding author email: 150398619@qq.com
doi:10.12720/jcm.10.10.760-765

system and save energy costs. But the cloud platform has the characteristics of random assignment request and a physical machine load real-time dynamic changes, it will lead to frequent physical machine ON and OFF. It can be seen that as to the current cloud computing for how to migrate virtual machines dynamically to achieve load balancing cluster problems, we proposed many efficient algorithms, but most algorithms are still significant deficiencies to be further improved. In cloud computing resource scheduling process [13], taking into account the re-deployment of virtual machine migration and integrated nature of the host is a big difficulty. Because the task's requests are random and physical machines have real-time dynamic load, we migrate all the virtual machines from the cluster of low load physical machine to close zero load physical machine which may cause frequent physical machine ON and OFF. So this paper proposed the virtual machine migration algorithm which focuses on how to exceed load threshold high load physical machine virtual machine live migration. This paper has made improvements in the following two aspects:

Traditional load balancing algorithms [14] only pay attention to one aspect of different load of the physical machine. In this paper, we also consider the CPU, memory and Multidimensional attribute of band width.

It make judgment according to the relatively unbalanced degree of physical machine load and choose the physical machine which has the lowest degree of relative load balancing to make the virtual machine migration redeployment, to achieve load balancing, and can effectively improve the system resource utilization and guarantee a certain QoS [15].

II. System Model and Problem Description

Firstly, build a cloud system model shown in Fig. 1.

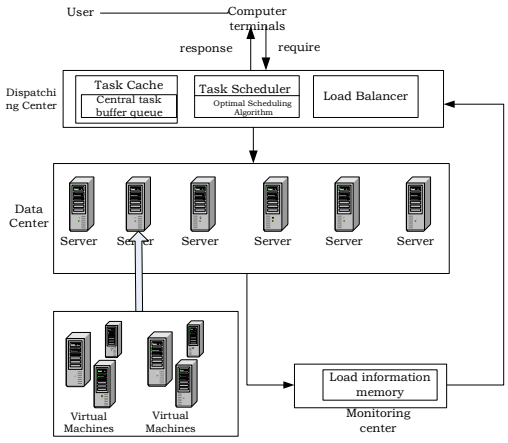

Fig. 1. Cloud system model

The Cloud System Model constructed in this paper, including a dispatch center, a data center and a monitoring center. There is a central task buffer queue in task cache of the dispatch. Assuming that customers in the cloud system according to the actual needs request to create a virtual machine in a physical machine, each request is known for a "task", and these tasks are stored in the cloud client based on the central task of the cache queue. Dispatch centers also have a load balancer, which periods *T* as a unit, to keep track of every single physical machine all load information (dynamic load parameters) [16]. The load memory in monitoring center is used to store the load information of each physical machine. Combined with the load balancer's information, according to the optimization algorithm, the task dispatcher of the dispatching center assign tasks to the relative load imbalance of the smallest physical machine. The physical machine instead of *PM* and single *PM* is Marked by PM_r, and $r = 1,2,3 \ldots R$. The resources of the physical machine on *PM* is defined as *I* dimensional, and the amount of host's resources is represented by N_{ri}, and $i = 1,2,3 \ldots I$. As we know, cloud computing resources include CPU, memory, bandwidth, storage space, etc., and we use *i* to represent the dimension of computer resources to keep generality.

In this paper, we take the multidimensional attribute of computing resources into account. And select the minimum relative degree of load imbalance to do virtual machine migration and redeployment, according to the physical machine load's relatively uneven degree.

III. Relative Load Imbalance Scheduling Algorithm

In the cloud system of this paper, the task can be completed within a period of time (*T*). The information storage of load balancer is used for storing the physical machine load information of the time *S* and *S-1*. First, for the physical machine under normal operation in cloud system, the system needs to determine whether there is overload in physical machine. If there is overload, we need dynamic migration running virtual machine in the physical machine.

For one physical machine, the average utilization of the PM_r is (w_{ri}) in time *T*. The total amount of *i* classes computing resources of each physical machine is τ_{ri} in cloud system, then the load average value of *i* classes computing resources is:

$$\delta_i = \frac{\sum \tau_{ri} * w_{ri}}{\sum w_{ri}} \tag{1}$$

E is relatively small constant, the load warning value is

$$\gamma_i = E + \delta_i \tag{2}$$

On the basis of the determination of the state of the physical machine in cloud system according to the alarm threshold set in advance, if the load exceeds the alarm value, the dispatching center will need to adopt a scheduling algorithm to dynamically migrate the virtual machines [17]. Suppose that the overload phenomenon of

the physical machine PM_r occurs in this period *S*, it's a need for making the virtual machine PM_r migration. Then, based on the load information of PM_r in the last time *S-1*, all the load information of other normal operation servers can be calculated into the following formula,

$$B_r = \sum_{i=1}^{I} \frac{a_i C_{ri} N_{ri}}{C_{mi} N_{mi}} \quad (3)$$

In this formula, B_r represents the load relative physical machine relative load imbalance degree of the physical machine PM_r. C_{ri} represents the average utilization of computing resource of the normal operation physical machine PM_r at the time of *S-1*. Such as CPU, memory, and so on the average utilization rate. N_{ri} represents the capacity of the computing resources of the physical machine PM_r. C_{mi} is the average utilization rate of the overloads physical machine resources at time S-1, and N_{mi} is the resource capacity of the overload physical machine at the time of *S-1*. a_i represents the weighting factor of computing resources. The initial value is 1, you can appropriately increase or decrease the value of a_i to emphasize or weaken the load requirements of resource, depending on the different emphasis in the process of instantiate establishment of the cloud platform.

Therefore, after using the formula (1) can be derived the load information ratio of all running servers in cloud platform to the reference physical machine, the dispatching center will select the physical machine with relatively minimal degree of load imbalance as an alternative host to migrate the virtual machine.

After selecting the alternate host, the need will be for further study of the load state of the alternative. Combined with requirements of the task, the alternate host will be selected as host of the eventual deployment of virtual machine under the circumstance that the alternative host will not be overloaded after virtual machine migrated to the alternative host. Assuming that the demand for the necessary resources to migrate virtual machines is an *I*-dimensional vector, each dimension represents the demand for each item of computing resources, as follows:

$$H = (h_1, h_2 \ldots h_i \ldots h_I), i = 1, 2, 3 \ldots I \quad (4)$$

The load information of alternative host at time *S* got from the storage of load information is still written in vector form:

$$N_{ri}^{available} = (n_1, n_2 \ldots n_i \ldots n_I), i = 1, 2, 3 \ldots I \quad (5)$$

Compare the size of h_i with n_i, and $h_i \le n_i$, $i = 1, 2, 3 \ldots I$.

Through the check on the alternate host according to the above constraints to, if the above conditions are not satisfied, we can select the host with the smallest degree of relative load imbalance as the alternate host except that host. It will be checked again according to the constraint till we can select the appropriate alternative host to make dynamic migration of virtual machine.

In this algorithm, the embodied ideas are as follows:

Firstly, determine the load or overload according to the load information of host to perform the live migration of virtual machine.

Secondly, select the host with the smallest relative degree of load imbalance as the alternate host to make the conduct of virtual machine migration according to the relative load imbalance degree calculated from the load information of this host and other normal operating hosts.

Finally, detect the resource capacity of the alternate to select the one co; forming to criteria to make the migration of virtual machine; otherwise, re-select the host with the smallest relative degree of load imbalance from the other hosts except the hosts have been checked. It will be checked again according to the constraint till we can select the appropriate alternative host to make dynamic migration of virtual machine.

The algorithm in this paper is presented as follows:

Algorithm: Minimum relative degree of load imbalance scheduling algorithm.

1) At Time *S*, we determine the physical machine appear overload phenomenon, you need to make the dynamic migration of virtual machines from the physical machine.

2) Get the load information at time *S-1* of this physical machine and other normal running physical machines.

3) Calculate relative degree of load imbalance scheduling algorithm of all the normal operation of the physical machine relative to the load and overload to the physical machine.

4) Select the physical machine with the smallest degree of the relative load imbalance as an alternative host from the item 3.

5) Check on the alternate host in item 4 according to the constraints to, if the above conditions are not satisfied, we can select the host with the smallest degree of relative load imbalance as the alternate host except that host. It will be checked again according to the constraint till we can select the appropriate alternative host to make dynamic migration of virtual machine.

IV. Simulation

In this paper, we use the simulation experiment CloudSim [18] to simulate cloud computing environment, and all experiments are been configured as a 64-bit X86 processor, 32G memory, hard disk space on the server 2T. In this test, we do experimental analysis according to the CPU, memory, and bandwidth.

We create three different types of physical machines in a data center in CloudSim platform, these three physical machine's configuration is as follows:

TABLE I: PM CONFIGURATION

Type	PM1	PM2	PM3
CPU/GHz	12	24	32
MEM/GB	16	24	32
Band Width/MB	100	150	200

In this test, the virtual machine creation types are six kinds, they were randomly assigned with equal probability to create a virtual machine tasks. Virtual machine configuration as follows:

TABLE II: VM CONFIGURATION

Type	CPU/GHz	MEM/GB	Band Width/MB
VM1	2	4	4
VM2	4	8	8
VM3	3	9	10
VM4	16	12	20
VM5	20	24	32
VM6	24	32	64

It is not easy to simulate virtual machine migration environment on Cloud Sim platform needed for this to simulate cloud platform by adding loads such as the probability of a random batch jobs and batch jobs long time to adjust. Modify CloudSim the Datacenter Broker module, every 10s tasks randomly added to the virtual machine. The addition of random factors, so the load is not stable, there will be fluctuations. Different virtual machines and the host via the monitor data are detected, at the same time the monitoring is not the same. The simulation will be automatically terminated when the process is completed all the tasks, and therefore modify the simulation end of the flag. When there is no task, it does not close the physical and virtual machines revoked.

In this paper, in order to verify the effectiveness of the algorithm, experiments will conduct comparative tests in the following three aspects. 1. Request a different number of tasks, the number of virtual machine migration. 2. a different number of task request, the utilization of computing resources within the system. 3. Before and after the virtual machine migration, the system load is not balanced contrast. Through the above three comparison test, to prove that the algorithm proposed in this paper has the advantage that it is effective.

In Experiment 1, the number of tasks to perform the same number of virtual machine migration point of view, to compare the scheduling algorithm proposed in this paper, the minimum cost migration scheduling algorithms and random scheduling algorithm [19]. According to the number of tasks, such as probability distribution, this will ensure that the virtual machine instance is created for each type of request for the same amount. Monitors and load balancers data collection interval is 5 seconds. The results shown in Fig. 2:

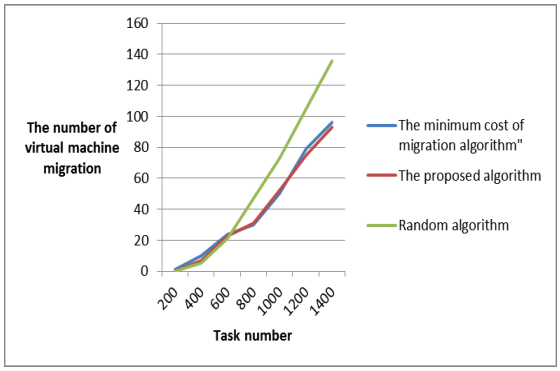

Fig. 2. The number of tasks in different number of virtual machine migration

The above analysis shows that the Random Scheduling Algorithm widely used in the commercial cloud platform has little actual migration of virtual machines when in the small number of tasks, due to the more opening physical machines, but when the numbers of tasks increase, it becomes poor and need more virtual machines to be migrated. This is because of the uneven distribution of the lead load in the cluster. Compared to the minimum cost of migration algorithm, the algorithm proposed in this paper can be found small performance gap to it. This is because these two algorithms to choose the timing of the virtual machine to a physical machine selection are load exceeds a threshold value to determine, and have focused on maintaining a balanced overall load. When a task requests a smaller number, these two algorithms to ensure the system load balancing, and physical host overload occurs can be kept in a small number. When the number of tasks increases, the virtual machine migration increased because of the virtual machine can deploy the limited physical host and calculate the effective utilization of the resources are limited.

In Experiment 2, to consider the request of different tasks, it compares the effects of three types of scheduling algorithms for average system resource utilization. The results shown in Fig. 3:

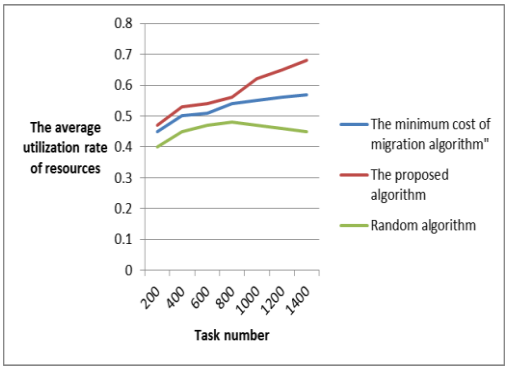

Fig. 3. The average system resource utilization with different tasks number

Above the simulation results we can found the average utilization rate of resources through this paper is much higher than the minimum cost of migration algorithm and stochastic algorithms. When the initial operation, the load

is not a minimum migration expense algorithm proposed in this paper to improve the system and the resulting resource utilization approximation algorithm, but the running time is increased when the load increases, and the resulting algorithm average resource utilization has proposed significantly improved. This is because when there is a physical machine after overload proposed algorithm before migrating a virtual machine, and select the system is relatively minimal degree of load imbalance host migration as the destination host to ensure that the load balancing system; The minimum load during the migration process migration algorithm considers only the largest of the target host resource utilization, overall system resource utilization was not considered. This shows that, in consideration of the system and load-balanced conditions, the proposed algorithm in the implementation of a number of tasks, can significantly improve the utilization of system resources, enhance the performance of the entire cloud system.

In Experiment 3, the probability of randomly generated request six types of virtual machines, and gradually increase the number of tasks, view the system load changes without equalization degrees. The results shown in Fig. 4:

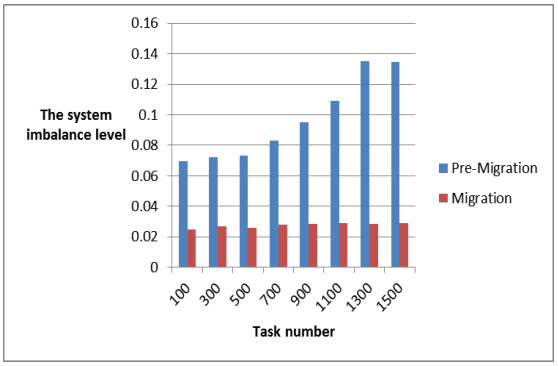

Fig. 4. The inequality of the virtual machine before and after migrate

From the chart we can clearly found that after the proposed method after virtual machine migration, load imbalance of the whole system has been significantly reduced. This is because prior to migration, there is a system load imbalance, some physical machine long period of high load state, and some of the physical machine but a large number of idle resources. After the virtual machine migration, the entire system to achieve efficient load balancing, resource utilization increased, resulting in the utilization of each resource variance to solve the inequality is lower.

V. Conclusion

Studying the pros and cons of resource scheduling algorithms mentioned by Scholars at home and abroad ,It is theoretically reveals how to achieve a condition in the cloud system under the condition of guaranteeing QoS and load balancing, the normal operation of the system is relatively Overload physical machine hosts are not relative load equilibrium is minimum. Meanwhile, Through CloudSim simulation platform, from the experimental point of view, It has obvious performance advantages for random scheduling algorithm used in commercial cloud platform, high resource utilization and load the inequality remained at a low level.

In future work, it is main task to put the scheduling algorithm used in this paper into commercial cloud platform in the load balancing process. Such as CloudStack, for further test the possibility of practical application of the algorithm.

References

[1] B. P. Rimal and E. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. Fifth International Joint Conference on INC, IMS and IDC. IEEE Computer Society*, 2009, pp. 44-51.

[2] S. Chaisiri, R. Kaewpuang, B. S. Lee, *et al*, "Cost minimization for provisioning virtual servers in amazon elastic compute cloud," in *Proc. IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. IEEE*, 2011, pp. 85-95.

[3] R. Prodan, M. Sperk, and S. Ostermann, "Evaluating high-performance computing on google app engine," *Software IEEE*, vol. 29, nol. 2, pp. 52-58, March 2012.

[4] M. Simms and M. Thomassy. (October 2014). Microsoft Inc. Best practices for the design of large-scale services on Windows Azure cloud services. [Online]. Available: http://msdn.microsoft.com/en-us/library/Azure/jj717232.aspx

[5] P. Kranas, A. Menychtas, V. Anagnostopoulos, *et al*., "ElaaS: An innovative elasticity as a service framework for dynamic management across the cloud stack layers," in *Proc. International Conference on Complex, Intelligent and Software Intensive Systems*, 2012, pp. 1042-1049.

[6] G. Fenu and S. Surcis, "A cloud computing based real timefinancial system," in *Proc. Eighth International Conference on Networks*, March 2009, vol. 48, pp. 374-379.

[7] X. Ren, R. Lin, and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast," in *Proc. Intelligence Systems IEEE International Conference on Cloud Computing*, September 2011, vol. 15, pp. 220-224.

[8] J. W. Jiang, T. Lan, S. Ha, *et al*., "Joint VM placement and routing for data center traffic engineering," *Infocom Proceedings IEEE*, vol. 131, no. 5, pp. 2876-2880, March 2012.

[9] S. Kaur and A. Verma, "An efficient approach to genetic algorithm for task scheduling in cloud computing environment," *International Journal of Information Technology and Computer Science*, vol. 4, no. 10, pp. 74-79, September 2012.

[10] J. C. Moore, H. R. Rao, and A. B. Whinston, "Information processing for a finite resource allocation mechanism," *Economic Theory*, vol. 8, nol. 2, pp. 267-290, 1996.

[11] X. Wang, X. Liu, L. Fan, and X. Jia, "A decentralized virtual machine migration approach of data centers for cloud computing," *Mathematical Problems in Engineering*, vol. 2013, pp. 831-842, March 2013.

[12] Y. Gao, H. Guan, Z. Qi, *et al*., "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *Journal of Computer & System Sciences*, vol. 79, nol. 8, pp. 1230-1242, February 2013.

[13] D. Ergu, G. Kou, Y. Peng, and Y. Shi, "The analytic hierarchy process: Task scheduling and resource allocation in cloud computing environment," *The Journal of Supercomputing*, vol. 64, no. 3, pp. 835-848, June 2013.

[14] S. Sud, R. Want, T. Pering, K. Lyons, B. Rosario, and M. X. Gong, "Dynamic migration of computation through virtualization of the

mobile platform," *Mobile Networks and Applications*, vol. 35, pp. 59-71, February 2012.
[15] A. M. Shamsul, J. W. Mark, and X. M. Shen, "Relay selection and resource allocation for multi-user cooperative OFDMA networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2193-2205, May 2013.
[16] Y. Zhang, X. Liao, H. Jin, L. Lin, and F. Lu, "An adaptive switching scheme for iterative computing in the cloud," *Frontiers of Computer Science*, vol. 8, no. 6, pp. 872-884, December 2014.
[17] H. Liu, H. Jin, C. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster Computing*, vol. 16, no. 2, pp. 249-264, June 2013.
[18] R. N. Calheiros, R. Ranjan, and A. Beloglazov, *et al.*, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software Practice & Experience*, vol. 41, no. 1, pp. 23-25, January 2011.
[19] P. Giaccone, *et al.*, "An implementable parallel scheduler for input-queued switches," *IEEE Micro Magazine*, vol. 22, pp. 19-25, January 2002.

Tao Xue was born in Shaanxi Province, China, in 1973. He received the B.S. degree in mechanical engineering from the Xi 'an Jiaotong University of China (XJTU), Xi 'an, in 1995 and the M.S. degree in computer science from the Northwest University of China (NWU), Xi 'an, in 2000 and the Ph.D. degree in computer software and theory from the XJTU, in 2005. He is currently an Associate Professor in the Computer Science Dept at Xi'an Polytechnic University. His research interests include Cloud Computing, Big Data and Content-based Networking.

Zhe Fan was born in Henan Province, China, in 1990. He received the B.S. degree in software engineering from the HuangHuai University of China, Zhumadian, in 2013. He is currently pursuing the M.S. degree with the Department of Computer technology, Xi'an Polytechnic University. His research interests include Cloud Computing, Big Data and Content-based Networking.