

# Optimal Media Service Scheduling in Clouds

Jiali You\*, Nannan Qiao\*\*, Yu Zhuo\*\*, Jinlin Wang\*

\*National Network New Media Engineering Research Center, Institute of Acoustics,  
Chinese Academy of Sciences, China

\*\*University of Chinese Academy of Sciences, China  
{youjl,qiaonn,zhuoy,wangjl}@dsp.ac.cn

**Abstract**—Cloud becomes increasingly important currently, and media services, which usually have multiple steps and complexity computing requirement, are applied in it. Although this scheduling topic has been investigated a lot, most of works focus on balancing jobs or decreasing the execution time of one job. For large scale paralleling media services, the job request usually contains a series tasks, and how to schedule them to a cloud or a grid with high efficiency and low cost is an important problem. To achieve this target, we focus on the fine grain tasks for media services and propose a task-based scheduling algorithm, in which two-stage scheduling approach is designed, including initializing stage and adjustment stage. In this algorithm, after the initializing assignment process, in order to meet the need of the limited cost consumption, the characters of nodes or virtual machines(VMs), such as balance character, capability character, are considered deeply in scheduling adjusting process. Evaluation shows that our algorithm presents obviously better performance in terms of makespan, cost consumption, and balance factor than several benchmarks.

**Keywords**—scheduling, adjusting stage, media cloud

## I. INTRODUCTION

Nowadays, the world becomes interconnected, in which a huge number of things may connect as one or more networks, for instance, Internet and Sensor networks. With the developing of applications in networks, the increasing speed of the amount of data is quite fast[1]. And in this large amount data, Internet video is estimated to reach 62% of the consumer Internet traffic by the end of 2015[2] and the mobile video will occupy 71% of all the mobile data traffic by 2016[3].

At the same time, Cloud computing become prevalent, whose essential is to release the enterprises and individuals from the IT infrastructure management and help them focus on the applications. A lot of cloud platforms are constructed and provide services for customs, such as storage space, virtual machines(VMs)[4-8]. Based on the Cloud computing technologies, large scale jobs can be decomposed into sub-tasks and processed over a cloud platform. With the increasing need of the media services, how to use cloud to support media processing is an important problem [9-11]. As we know, tasks involved in media applications have some characters[11], such as multiple steps, time-related tasks, large scale jobs, high complexity computation. And the real-time response is usually required in media services, which is quite difficult to be achieved. Consequently, for the large scale parallel requests, with the cost limitation, how to cope with them efficiently and reliably is an important topic.

In this paper, a task-based parallel scheduling algorithm with two-stage processing for media service is proposed, in which a task-based dynamic programming algorithm is used to initialize the scheduling scheme for all tasks in parallel jobs and a heuristic strategy is used to refine the task assignments. Compared with several state-of-the-art algorithms, the proposed algorithm shows less total execution time, and the node balance in cloud is obviously improved.

The remainder of this paper is structured as follows. Section II introduces the related work about scheduling algorithms. Section III illustrated the task-based heuristic scheduling algorithm. The section IV's simulation evaluates our approaches. Finally, conclusions are presented in section V.

## II. RELATED WORK

Scheduling algorithms are usually discussed in cloud computing and grid networks, which can be classified into two categories, job scheduling and task scheduling.

In job scheduling, no matter how many steps a job has, the job is the grain size for scheduling, and a lot of parallel requests are distributed to different computing nodes or VMs according to their current workload. Based on this process, the situation that unbalance workload and traffic congestions in a cloud may be effectively avoided. In [12, 13], some jobs are send to the cloud, and the cloud service can generally be divided into three consecutive phases: schedule, computation and transmission. Three concatenated queening systems are used to model the service process to allocate the resource appropriately for different phrases to reduce the mean respond time under the cost constraint. In this work, the granularity of the task is the whole job. When the amount of jobs is very large and multiple sequential tasks in a job should be processed, how to schedule them more efficiently is also a problem. Similarly, Gang scheduling, which is for job-based scheduling problem in Cloud platform, is used in [14, 15], and a job migration approach is proposed for the starving VMs in [14]. Besides, Genetic Algorithms [16] are also used to aim to minimize both the makespan and the energy consumption for the job-level task.

In practice, a job may be composed by several tasks. Some of them can be executed in parallel, while some tasks must be processed serially. How to calculate the optimal task assignment scheme is the purpose of task scheduling. In this process, most work focus on finding the appropriate assignment scheme for each job, and the goal of task-level

scheduling is to assign tasks to VMs or nodes so that the total execution time of a job can be minimized with the cost limitation. Related work is done by [17-21]. In [18], a directed acyclic graph is used to model the precedence constraints among tasks, in which the task-level scheduling for the sequential, the parallel, and the mixed structures are optimized, respectively. And a heuristic method is proposed to accelerate the optimizing scheduling. With the similar purpose, in Grid computing, scheduling workflow applications [19] is also the task-level problem. In this work, Multiobjective Differential Evolution [22] is used to solve the multiple-object optimization problem. Besides, a cost-based workflow scheduling algorithm for time-critical workflow applications is proposed by [20], in which the workflow is divided into some task partitions, and the scheduling process is based on the task partitions with the deadline constraint. In each branch partition, the sequential tasks are modeled by Markov Decision Process, which can effectively solve the decision problem.

In a real system, job scheduling and task scheduling can be seen as two levels of scheduling process, and they represent different aspects. For instance, a CDN provider will update a large number of video files which should be pre-processed before uploaded, such as transcoding or feature extracting, and a lot of requests will be proposed concurrently. Similarly, a large scale images or audios in a website should be extracted features and generated indexes periodically, which are also parallel jobs that contain several time related tasks. For all of these kinds of applications, not only the job-level scheduling, but also the task-level scheduling for each job should be taken into account. However, these situations are not investigated deeply in the state-of-the-art works. Therefore, for a lot of jobs, especially for media services, how to schedule them in a cloud to minimize their total execution time is our goal. In this paper, a task-based heuristic scheduling algorithm is proposed, in which different effects, such as cost, workload, computing capability, are all taken into count to get a preferable result. And the details of the proposed algorithm will be depicted in the following section.

### III. TASK-BASED SCHEDULING ALGORITHM FOR JOBS WITH TIME RELATED TASKS

In this section, the scheduling model will be represented firstly; then the two-stage framework, including the assignment initializing and scheduling process refining will be introduced, respectively. Finally, the assignment initializing approach will be introduced in detail.

#### A. Model Description

For a cloud, it contains  $N$  nodes, which compose a node set  $\theta$ . It is assumed that there are  $J$  jobs  $O = \{o_1, o_2, \dots, o_j, \dots, o_J\}$ , which should be processed by this cloud, and each job  $o_j$  can be divided into  $Q_j$  tasks, where  $o_j = \{g_1, \dots, g_q, \dots, g_{Q_j}\}$ . The goal of scheduling is to assign all tasks in  $J$  jobs to appropriate nodes to optimize the total execution time with the constraint of maximum cost consumption  $C_{limit}$ . In this issue, if the task  $g_q$  of  $o_j$  is scheduled to node  $v_i$ ,  $s_{i,j,q}$  is set as 1; otherwise, it is 0. The

execution time of job  $o_j$  is  $t_j$  and the value for  $g_q$  processed by  $v_i$  is  $t_{i,j,q}$ , in which the waiting time of  $g_q$  is also concluded. Thus, the objective function is shown as formula (1).

$$\text{Minimize Max } t_j \quad (1)$$

Subject to

$$C_{total} \leq C_{limit}$$

$$C_{total} = \sum_{j=1}^J \sum_{q=1}^{Q_j} \sum_{i=1}^N t_{i,j,q} s_{i,j,q} c_i$$

$$\forall i = 1, 2, \dots, N, \forall j = 1, 2, \dots, J, \forall q = 1, 2, \dots, Q_j \quad s_{i,j,q} \in \{0, 1\}$$

Here,  $c_i$  means the cost consumption of node  $v_i$  in unit of time.

Actually, the above formula is a NP-hard problem, and the optimal result is hard to get. Usually, heuristic algorithm is used to calculate the sub-optimal solution. In our work, we proposed a task-based two-stage scheduling algorithm to calculate the scheduling assignments for all jobs.

#### B. The Task-based Two-Stage Scheduling Framework

For a job, the total execution time of it, named makespan, is determined by the critical path, which is the series of tasks (or even a single task) that dictates the longest path. Obviously, by speeding up the execution of critical tasks, we can reduce the total execution time of it. In order to find the critical tasks, Breadth-First Search (BFS) and Depth-First Search (DFS) algorithms are usually used to analyze the critical path and compute start times, proportion and deadlines. Therefore, our purpose in this paper focus on reducing the execution time of the critical tasks of jobs with the limitation of cost consumption of all jobs.

Based on the above analysis, we proposed a task-based two-stage scheduling algorithm as figure 1, in which initializing stage and adjusting stage are executed iteratively. Firstly, tasks of all jobs compose a task set  $\Phi$ . If  $\Phi$  is not null, tasks in it should be assigned to different nodes based on initializing stage, which will be introduced in section III.C. Then, if the cost over current scheme is more than the cost constraint  $C_{limit}$ , the adjusting stage is evoked. In this stage, the critical path of each job is extracted, and the assignment for critical tasks may be tuned to another node based on the tradeoff among the execution time, the cost and the workload of nodes. And the goal of this step is to move the tasks to the node which can obviously reduce the cost and just increase a little makespan. At the same time, the workload of nodes is also considered to avoid the overload problem. In this calculating process, if the cost constraint is satisfied, all tasks will be assigned to the new nodes and the scheduling calculation is terminated; otherwise, a new iteration will be evoked until all tasks are completed processed. Because of the time-related character of tasks, tasks in  $\Phi$  are ranked by their execution order, and the one with the lowest score have the priority to be implemented.

After scheduling calculation, tasks should be processed by different nodes based on the assignment scheme. If the tasks

with the lowest score are completed implemented, the expected execution time and the cost consumption should be both updated by the real value, and these tasks are removed from  $\Phi$ . Then, the above steps should be iteratively executed until  $\Phi$  becomes null.

In this two-stage framework, an important step is to calculate the tradeoff value, in which two factors are considered. One is that nodes with more capability may retain more tasks; the other is that the task is prone to be move to the node which can reduce more cost and increase less execution time. Besides, when the amount of jobs is large, each node may receive a lot of requests at the same time. In our work, all received tasks are scored by their executing steps in a job, for instance, the tasks in a same level of jobs have the same score and the service order is based on this score.

Input: Task set  $\Phi$  of all jobs, total cost consumption constraint  $C_{limit}$

Output: A schedule for all tasks

While  $\Phi$  is not Null

    Initializing the assignment scheme of tasks in  $\Phi$

    Calculate  $C_{total}$  based on above assignment, where  $C_{total}$  is the expected cost.

    While  $C_{total} > C_{limit}$

        Sort jobs in descending order based on their needed execution time and get a job set  $O'$

        Get the critical path  $G$  of the head job  $o_j$  in  $O'$ ;  $g_q$  is any critical task of  $o_j$ ; and the corresponding execution time of  $o_j$  is  $t_j$

        For each  $g_q$  of  $o_j$

            For each  $v_i$  in  $\theta$

                Estimate the execution time  $t_{i,j,q}$  of  $g_q$  in  $v_i$ , then

$$\Delta t_{j,q} = t_{i,j,q} - t_{i,j,q}, \Delta c_{i,k} = c_{i,j,q} - c_{i,j,q} \text{ and}$$

$$\gamma_{i,q} = \alpha \frac{\Delta t_{i,q}}{\Delta c_{i,q}} / \sum_{i'=1}^N \frac{\Delta t_{i',q}}{\Delta c_{i',q}} + \beta \frac{l_{i,q}}{\sum_{v_i' \in \theta} l_{i',q}},$$

where  $t_{i,j,q}$  is the old execution time of  $g_q$  and  $v_i$  is any node in  $\theta$ .  $\alpha$  and  $\beta$  are the pre-set parameters by experience, and  $\alpha + \beta = 1$ .  $l_i$  represents the completion time to execute task  $g_q$  in node  $v_i$ , which is also a factor to indicate the workload of nodes.

        End For

    End For

    Sort all  $\gamma_{i,q}$  in the ascending order.

    Select the minimum  $\gamma_{i,q}$  and replace the old scheduling node with  $v_i$ .

    Update  $C_{total}$  based on the new scheme

End While

All tasks in  $\Phi$  should be scored by their execution order. For example, task  $g_q$  is scored by  $q$ .

Execute task with the minimum score. The real execution

time of them is  $t_{real}$  and the real cost is  $c_{real}$

then,  $C_{limit} = C_{limit} - c_{real}$ ;

Update the expected execution time  $t_{total}$  by the real value.

Remove these tasks from  $\Phi$ .

End While

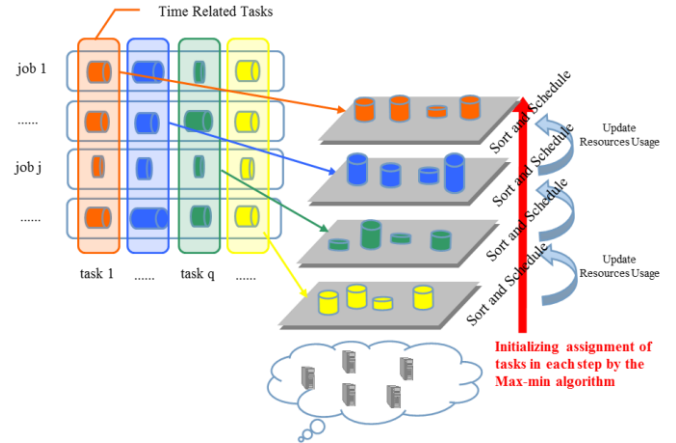
Return  $t_{total}$ ;

**Figure 1.** Task-based Two-Stage Heuristic Scheduling Algorithm

### C. Initializing the Scheduling Assignment

In order to get an initial assignment which is close to the optimal scheduling scheme, a task-based assignment approach is used to initialize the jobs to resources as figure 2. Firstly, the tasks of all jobs in the same level are sorted based on their resource requirement in a descending order. Then, the tasks in a same level are assigned to nodes based on the Max-min algorithm, and the objective is to minimize the maximum execution time. After this process, the state of nodes should be updated and the assigning process should be implemented for tasks in the next level until all tasks have been scheduled.

Based on the above work, all jobs can get the initializing scheduling assignments. If the cost is larger than the constraint, a refining process will be evoked, in which some tasks may be moved to another nodes and the workload, capability of nodes and cost consumption are all considered as figure 1 shown.



**Figure 2.** The initialization process of jobs.

## IV. SIMULATION AND EVALUATION

### A. Simulation Setup

To simulate a media cloud, it is assumed that there is a data center, which is composed of 20 nodes or VMs for computing. MIPS, which means Million Instructions Per Second, is used to represent the computing ability of nodes or VMs, and Million Instructions(MI) represents the requirement of jobs. The capability of each node follows the uniform distribution in [1000, 20000] MIPS, and the requirement of jobs follows the uniform distribution in [100000, 1000000], which is composed of 3 serial steps, and the corresponding size of each job is from 200M to 1G bytes. The cost of nodes is not the same, which is designed, based on their ability, for example, the nodes with high computing ability is more expensive than

others, and the value is selected from 0.01 to 0.2 cents per second.

We implement three algorithms as the benchmarks for comparing with our algorithm.

- Random Job Scheduling: All jobs are assigned to different nodes randomly.
- Min-min: Assign the smallest job to the node which can execute it completely earliest.
- Max-min: Assign the largest job to the node which can execute it completely earliest.

### B. Evaluation Metrics

We evaluate different metrics as table 1 shown to measure the performance of scheduling algorithms.

TABLE 1. THE EVALUATION METRICS

Metrics	Description
<b>Makespan</b>	The time difference between the start and end of a sequence of jobs or tasks.
<b>Workload</b>	The workload of the nodes, which is calculated by the real workload divided by the computing capability of nodes.
<b>Balance Factor</b>	The standard deviation of nodes workload
<b>Capability Factor</b>	The capability of a node in unit cost consumption, which is calculated by the running time of the node divided by the cost.
<b>Miss Rate</b>	The ratio of jobs that cannot be completed with the constraint.

### C. Evaluation of Different Algorithms without the Cost Constraint

For the benchmark algorithms, most of them do not consider the constraint of the cost or other factors, and just try the best to complete the computing. To evaluate the performance of different algorithms, we set task-based algorithm without the cost constraint and compare it with

benchmarks. The corresponding performance is illustrated in table 2.

TABLE 2. THE BALANCE FACTOR OF DIFFERENT ALGORITHMS WITHOUT THE COST CONSTRAINT.

Algorithm	MakeSpan(s)	Cost(cents)	Balance Factor	Capability Factor
<b>Task-based</b>	628	3580	2.44	5.70
<b>Max-Min</b>	685	3735	74.05	5.40
<b>Min-Min</b>	844	4180	263.06	4.90
<b>Random Scheduling</b>	6791	2648	1699.50	0.39

From the above result, we can see that without the cost limitation, our algorithm shows less makespan and cost consumption simultaneously than most benchmarks. The balance factor of task-based algorithm is significantly better than the other algorithms. The probably reason for this excellent balance is that the fine grain task can make the scheduling more precision. Besides, with the increasing of the cost constraint, the capability factor of task-based scheduling algorithm is obviously promoted, which means that the nodes with strong computing ability are prone to be selected to reduce the makespan when the cost consumption is not constrained too much.

### D. Performance of the Two-Stage Algorithm with Different Cost Constraint

In some situations, the limitation of the cost is different. Thus, we compared the performance of task-based algorithm with different value of cost constraint as figure 3. From this figure, we know that when we get the same makespan as the Max-min algorithm, around 9% cost consumption can be reduced; compared with the Min-min algorithm, about 30% cost consumption can be saved, when they have the close execution time. Similarly, either Balance factor or Capability factor from the task-based scheduling method shows their satisfying performance.

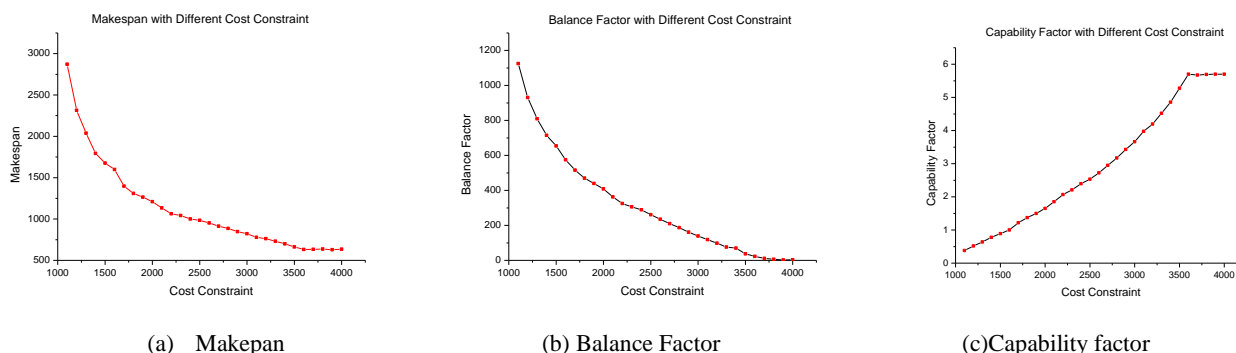
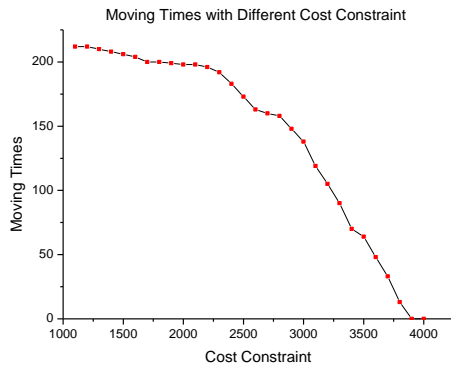


Figure 3. Performance of task-based scheduling algorithm with different cost constraint.

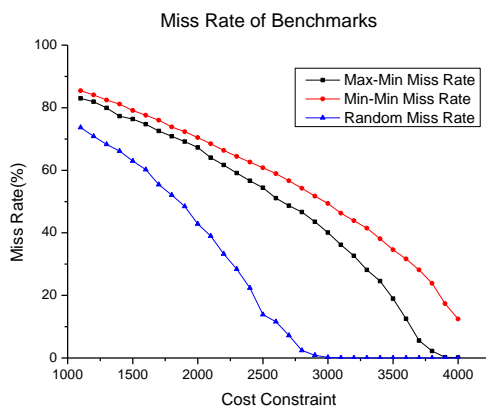
In our algorithm, when the cost is restricted, the assignment of tasks should be adjusted to cater for the cost limitation. The more tasks are moved, the more cost consumption may be saved. Figure 4 presents the moving times with different cost constraint, which proves this conclusion.



**Figure 4.** Moving times of task-based scheduling algorithm with different cost constraint.

### E. Discussion and Analysis

Actually, Max-min algorithm can achieve an acceptable performance when cost is not constrained. However, in some situations, when cost is limited, this algorithm and other benchmarks may miss a lot of jobs. With different constraint value, the service rejection rate is figured in figure 5. For example, when 10% cost of Max-min algorithm is limited, around 30% jobs are missed, and this value is much higher for the Min-min algorithm. Consequently, fine grain scheduling and accuracy adjustment is needed for media service when cost consumption is restricted. In addition, the initialization stage has significant impact for the final performance, because the adjustment process is based on the initializing assignment. Thereby, we will focus on how to optimize this process in our future work.



**Figure 5.** Miss rate of benchmarks when cost consumption is constrained.

## V. CONCLUSIONS

Based on the characters of media services, a task-based hierarchical scheduling algorithm is proposed, which has two stages, including initializing stage and adjustment stage. In the first stage, fine grain tasks are assigned to the appropriate nodes or VMs based on the nodes capability. Then, with the cost constrained, the initialized assignment should be refined in view of several characters, including cost reduction, balance factor, capability of nodes, whose purpose is to try to get a high efficient assignment with the low cost. In the

simulation, several algorithms are implemented as benchmarks and compared with task-based scheduling algorithm. Results show that our algorithm can significantly reduce the makespan when cost is not limited. If the cost is restricted, it also shows obviously better performance in balance factor and miss rate than benchmarks. In addition, we found that the initializing stage impact the performance significantly, thereby, we will focus on this problem in our future work.

## ACKNOWLEDGMENT

The work is supported by part of the National Key Technology R&D Program(2012BAH02B03), the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06010302) and the Innovation Project of Institute of Acoustics, Chinese Academy of Sciences(Y154211601).

## REFERENCES

- [1] "Big Data Infographic: Solve your Big Data Problems?", <http://www.intel.in/content/www/in/en/big-data/solving-big-dataproblems-infographic.html>.
- [2] "Visual Networking Index", [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/whitepaper/c11-481360ns827/Networking Solutions White Paper.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/whitepaper/c11-481360ns827/Networking%20Solutions%20White%20Paper.html).
- [3] "Visual Networking Index," [http://www.cisco.com/en/US/netsol/ns827/networking\\_solutions\\_sub\\_solution.html#forecast](http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html#forecast).
- [4] A. Weiss, "Computing in the Clouds," *NetWorker*, vol. 11, no. 4, pp.16–25, 2007.
- [5] G. Lin, D. Fu, J. Zhu, and G. Dasmalchi, "Cloud Computing: It as a Service," *IT Professional*, vol. 11, no. 2, pp. 10–13, 2009.
- [6] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud Computing: An Overview," *Proceedings of the 1st International Conference on Cloud Computing*, Springer-Verlag, pp. 626–631, 2009.
- [7] D. Hilley, "Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings," *CERCS, Georgia Institute of Technology*, Tech. Rep. GIT-CERCS-09-13, April 2009.
- [8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, et al., "Above the Clouds: A Berkeley View of Cloud Computing," *EECS Dept., Uni. of California, Berkeley*, Tech. Rep. UCB/EECS-2009-28, Feb '09.
- [9] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia Cloud Computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.
- [10] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: When Cloud on Demand Meets Video on Demand," *the 31st International Conference on Distributed Computing Systems (ICDCS)*, pp. 268 - 277, 2011.
- [11] Z. Zhao, Y. Zhao, Z. Gao, X. Nan, M. Mei, H. Zhang, et al., "Bupt-mcprl at Trecvid 2009," *TREC Video Retrieval Evaluation Online Proceedings*, Gaithersburg, MD, USA, 2009.
- [12] X. Nan, Y. He, and L. Guan, "Optimal Resource Allocation for Multimedia Cloud in Priority Service Scheme," *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1111-1114, 2012.
- [13] X. Nan, Y. He, and L. Guan, "Optimal Resource Allocation for Multimedia Cloud Based on Queuing Model," *IEEE 13th International Workshop on Multimedia Signal Processing (MMSp)*, pp. 1-6, 2011.
- [14] H. D. K. Ioannis A. Moschakis, "Performance and Cost Evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling," *2011 IEEE Symposium on Computers and Communications (ISCC)*, pp. 418 - 423, 2011.
- [15] I. A. Moschakis and H. D. Karatzas, "Evaluation of Gang Scheduling Performance and Cost in a Cloud Computing System," *The Journal of Supercomputing*, Volume 59 Issue 2, pp. 975-992, February 2012.



- [16] J. Kolodziej, S. U. Khan, and F. Xhafa, "Genetic Algorithms for Energy-aware Scheduling in Computational Grids," the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp.17 - 24, 2011.
- [17] M. Silberstein, D. Geiger, A. Schuster, and M. Livny, "Scheduling Mixed Workloads in Multi-grids: The Grid Execution Hierarchy," HPDC, pp. 291 - 302, 2006.
- [18] Y. He, X. Nan, and L. Guan, "Optimal Task-level Scheduling for Cloud based Multimedia Applications," ICASSP, pp.3771-3775, 2013.
- [19] A. K. M. K. A. Talukder, M. Kirley, and R. Buyya, "Multiobjective Differential Evolution for Scheduling Workflow Applications on Global Grids," Concurrency and Computation: Practice and Experience, Vol. 21, issue 13, pp. 1742-1756, 2009.
- [20] J. Yu, R. Buyya, and C. K. Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids," First International Conference on e-Science and Grid Computing, 2005.
- [21] J. Jin, J. Luo, A. Song, F. Dong, and R. Xiong, "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing," the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 295 - 304, 2011.
- [22] R. Sarker and H. A. Abbass, "Differential Evolution for Solving Multi-objective Optimization Problems," Asia-Pacific Journal of Operational Research, Volume 21, Issue 02, 2004.

receiving system, IP network technology and network streaming media, structure and new service of wideband network, mobile terminal technology and its value-added service, the third generation mobile communication technology, modern wireless communication technology.



**Jiali You** is an associate professor of the National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences. She received her B.S in Computer Science from Communication University of China in 2003, and Ph.D in Signal and Information Processing from the Institute of Acoustics (IOA), Chinese Academy of Sciences (CAS) in 2008. From July 2008, she joined the National Network New Media Engineering Research Center, IOA, CAS. Her research interests include future network,

content distribution network, media cloud, and P2P streaming system.



**Nannan Qiao** received a bachelor's degree from Beijing University of Posts and Telecommunications in 2013. Now, she is pursuing the master degree in the National Network New Media Engineering Research Center. Her research interests include cloud computing, future networks.



**Yu Zhuo**, graduated from University of Science and Technology of China at 2012, who is now studying at the Institute of Acoustics, Chinese Academy of Sciences as a Ph.D student and her research areas include virtualization network and future networks.



**Jinlin Wang** is a Professor, Doctor Supervisor, Director of DSP Center, Director of Network and New Media Technology Research Center. He graduated from Mathematics Department of University of Science and Technology of China with his bachelor degree in 1986. After he got his master degree from Institute of Acoustics, Chinese Academy of Sciences in 1989, he began to work in the institute and was engaged in the study of digital

signal processing. He had been the principal of many projects affiliated to "863 Program". His current research interests include digital signal processing, application of DSP, digital TV source and channel decoding technology and