PROCEEDINGS OF
THE 2008 INTERNATIONAL CONFERENCE ON
COMPUTER DESIGN

CDES 2008

Editor

Hamid R. Arabnia

Associate Editors

Youngsong Mun, Ashu M. G. Solo

WORLDCOMP'08
July 14-17, 2008
Las Vegas Nevada, USA
www.world-academy-of-science.org

Contents

SESSION: ALGORITHMS, CIRCUIT/HARDWARE DESIGN, AND TOOLS

SESSION: POWER AND ENERGY

Transcoding Load Distribution Policy for Wireless Mobile Clients

Dongmahn Seo, Heonguil Lee, and Inbum Jung
Department of Computer, Information, and Communications
Kangwon National University, Korea

dmseo@snslab.kangwon.ac.kr, hglee@kangwon.ac.kr, ibjung@kangwon.ac.kr(corresponding author)

Abstract

Recent advancement in wireless network technologies has enabled the streaming media service on the mobile devices such as PDAs and cellular phones. Since the wireless network has low bandwidth channels and mobile devices are actually composed of limited hardware specifications, transcoding technologies are needed to adapt a streaming media to given mobile devices. However, the ceasing and jittering phenomena deteriorate the quality of streaming media service. To avoid this problem, original MPEG media should be transcoded and transmitted to clients in the range of limited times. In particular, when large scale mobile clients demand streaming services, the load distribution policies among transcoding servers highly impact on the total number of QoS streams. In this paper, the resource weighted load distribution policy is proposed for the fair load balancing and the more scalable performance in cluster-based transcoding servers. Our proposed policy is based on both the weight of resources consumed for transcoding to classified client grades and the maximum number of QoS streams actually measured in transcoding servers. The proposed policy is implemented on cluster-based transcoding system. In experiments, this policy shows the fair load distribution and scalable performance according to the increase of transcoding servers.

Key words:
streaming media, mobile clients, transcoding, load distribution

1. Introduction

Based on the amazing growth of telecommunication, computer and image compression technologies, the streaming media service has been spotlighted in many multimedia applications. In particular, recent wireless network technologies have enabled the streaming media service on the mobile devices such as PDAs and cell phones. However, the streaming media are composed of a larger and more complex data compared with traditional text, graphic data. Thus, the large amount of network traffics and the high performance computing ability are inevitable to support the QoS streams [1, 2, 3]. However, since a wireless network works on low bandwidth channels and many mobile devices have limited hardware specifications, transcoding technologies are needed to adapt the originally encoded MPEG media to given mobile devices. The ranges of these technologies include not only the change of frame rates, bit rates, video sizes but also the re-encoding of MPEG I, II media into the MPEG IV.

A transcoding system is composed of both the multimedia server with an originally encoded MPEG media and multiple transcoding servers to perform the adaptation to a given environment. The multimedia server retrieves the MPEG media and sends them to a selected transcoding server. Transcoding servers perform transcoding functions and also provide a streaming service for mobile clients. Since each mobile device has its own working specifications, the transcoding loads of a service differ from mobile device to mobile device. According to mobile device types, transcoding servers consume different amounts of CPU time, memory space and network bandwidth. From these resource consumption rates, mobile devices can be classified as several transcoding grades in the server side. To achieve the efficient load balancing between transcoding servers, the transcoding grades should be reflected on a load distribution strategy.

In this paper, the load distribution policies to transcoding jobs for mobile clients are studied in cluster based servers. The cluster server architecture has an advantage of the ratio of performance to cost and is easily extended from the general PCs [18]. This model usually consists of a front-end node and multiple backend nodes. In our research, the front-end node is used as a load distribution server and the backend nodes work as transcoding servers. Based on load distribution strategies, the load distribution server distributes the transcoding requests from mobile clients into transcoding servers. In this paper, to provide the QoS streams for various kinds of mobile clients, a new load distribution policy is proposed in cluster-based transcoding servers. As the criteria of load distribution, we measure both the actual resource consumption rates and the maximum number of

* This research was financially supported by the Ministry of Knowledge Economy(MKE) and Korea Industrial Technology Foundation(KOTEF) through the Human Resource Training Project for Regional Innovation

QoS streams according to transcoding grades. From these measurements, the load weight values are driven. The load weight values to transcoding grades are not only used in the load distribution policy but also utilized as the threshold point of admission control to guarantee QoS for all clients. The proposed policy is implemented on cluster-based transcoding system together with previous load distribution policies. From our experiments, the proposed policy shows the fair load distribution in the heterogeneous transcoding servers and it leads to better performance scalability according to the increase of transcoding servers. This enhanced performance can contribute to the streaming service for large scale mobile clients.

The rest of this paper is organized as follows. Section 2 describes related work for our research. In section 3, the proposed load distribution policy is explained. Section 4 explains our actual experimental environment. In section 5, the performances of load distribution policies are evaluated. Section 6 concludes the paper.

2. Related Work

2.1 Media Grades

Mobile devices have low computing power, small memory space, low network bandwidth. To provide streaming media services for these mobile devices, original MPEG media should be transformed to adapt to the corresponding mobile devices. There are the specifications of MPEG media to support various client types. Table 1 show video sizes, frame rates, bit rates according to classified 4 grades [13]. When a transcoding process is needed, the 4CIF media is an original copy, the SQCIF, QCIF, CIF are used as objective transcoding grades.

Table 1: Classification of MPEG Media

Grade	Video size	Frame rate	Bit rate (kbps)	Client device
SQCIF	128×96	15	50	cellular phone
QCIF	176×44	15	70	PDA
CIF	352×288	26	100	notebook
4CIF	704×576	30	200	Desktop PC

2.2 Transcoding System

To implement transcoding systems, several approaches had been proposed. They include source based static encoding systems, static transcoding server systems, and load distribution transcoding systems, etc [11, 19]. In the source based static encoding system, a server stores the MPEG videos encoded by all client grades. Due to the absence of on-line transcoding overheads, this approach takes an advantage on the side of streaming service. However, it is difficult to prepare the encoded videos for all kinds of various mobile clients. And also, to store multiple copies to a MPEG movie title makes, storage space is wasted. The static transcoding server system chooses a transcoding server close to the wireless base of mobile clients. In this approach, specific servers could be saturated by concentrated transcoding jobs. To address this problem, the load distribution transcoding system is proposed to distribute transcoding jobs. In this approach, a load distribution server monitors the load of all transcoding servers and sustains a load distribution policy.

2.3 Load Distribution Policies

Many researches were undertaken for load distribution policies in cluster-based servers. In particular, cluster based server architecture has been utilized in the Web server, game server and file server areas. As representative load distribution policies, there are RR(Round Robin), LC(Least Connection), WRR(Weighted Round Robin) and DWRR(Dynamic Weighted Round Robin) and so on [18, 19].

The WRR policy designates a different weight to each server according to the capability of servers. For example, if the basic weight value is 1 and server *A*, *B*, *C* have 4, 3, 3 weights respectively, the order of job scheduling is *ABCABCABA*. In this approach, the state changing of servers can not be reflected dynamically. To address the problem, a DWRR policy is suggested. For distributing jobs to servers, this policy considers the current state of backend servers. However, as the number of clients increases abruptly, it incurred heavily communication overheads between a load distribution server and backend servers.

3. Resource Weight Load Distribution Policy

In this paper, we propose the RWLD(*R*esource *W*eight *L*oad *D*istribution) policy to satisfy the real time requirement and to get the more scalable performance in cluster-based transcoding servers. For the RWLD policy, the actual amounts of resources consumption for transcoding processes are measured on the individual transcoding servers. The measurement is performed per the grade of mobile device. In addition, the maximum numbers of QoS streams by transcoding grades are measured on each transcoding server. Based on the two types of measured information, our RWLD policy manages the fair load balancing between heterogeneous cluster servers as well as provides the scalable performance according to the increase of transcoding servers.

3.1 Resource Consumption Rates by Transcoding Grades

Table 2: Specification of Experimental Movies

Title	Video size (horizontal pixel x vertical pixel)	Frame rate (number/sec)	Running time (minutes)
NARUTO	640×480	26	35
COOLNESS AND PASSION	704×384	26	95
TOTORO	640×480	26	30
MONKEYTURN	640×480	26	32
MIDORI LIFE	640×480	26	30
DEVIL	640×480	26	30
HEAVEN	640×480	26	35
MILD WINDS	640×480	26	35
YUGO	640×480	26	30
MIRUMODE	640×480	26	30

To find the actual amount of resource consumption for each transcoding grade, we measure the usage of CPU, memory and network bandwidth exhausted by the classified grades described in the Table 1. A Desk-Top PC has a role for a transcoding server which is composed of 1.4 GHz CPU, 256 Mbytes Memory, and 100 Mbps Network Bandwidth. The Linux operating system is deployed and the FFMPEG program is used for the transcoding of MPEG media [14].

Table 2 shows the detail specification of 10 movies used in our experiments. They are MPEG-4 avi media and have enough running time to evaluate their performance in our system. These 10 movies are the originally encoded to the 4CIF grade. In our experiments, these 10 movies are transcoded into the SQCIF, QCIF and CIF grade respectively and the resource consumption rates are measured while the transcoding operation proceeds.

Table 3: Resource Consumption Rates

Grade	CPU (GHz, %)	Memory (Mbyte, %)	Network (Kbps, %)
SQCIF	0.116, 8.3	5.7, 2.2	50, 0.05
QCIF	0.119, 8.5	5.8, 2.3	70, 0.07
CIF	0.228, 16.3	6.4, 2.5	100, 0.1

Table 3 shows the experimental results for transcoding 10 4CIF grade movies into SQCIF, QCIF and CIF grade respectively. As experimental results, we find that the transcoding for the same grade results in almost same resource consumption rates regardless of which movies are selected. The results are from the average of 5 times measurements under same experimental environments. As shown in this Table, the transcoding into the CIF grade shows the most resource consumption rates. And also, the CPU consumption rates take the highest portion. Based on these observations, the resource consumption weights to a corresponding transcoding grade can be driven in the next section.

3.2 Resource Weight Table

In our RWLD policy, a load distribution server has a RWT(*R*esource *W*eight *T*able) for the fair load balancing and the admission control. The RWT preserves the information of 4 items to each transcoding server. They are relative resource consumption weights, maximum number of streams, total resource weights and consumed weights.

Table 4: Pseudo Codes for Relative Resource Consumption Weights

```
// C : available CPU capacity
// M : available total Memory space
// N : available Network bandwidth
// g : transcoding grade

// Number of transcoding jobs by CPU capacity
N_tr_cpu = C / Qcg ;
// Number of transcoding jobs by Memory space
N_tr_memory = M / Qmg ;
// Number of transcoding jobs by Network bandwidth
N_tr_network = N / Qng ;

if ( N_tr_cpu < N_tr_memory && N_tr_cpu < N_tr_network){
// CPU resource is exhausted firstly
Wg = (Qcg × 100) / (Σ_{k=1}^{3} Qck)   ( g = 1,2,3 )  --- equation (1)
} else if ( N_tr_memory < N_tr_cpu && N_tr_memory <
    N_tr_network){
// Memory resource is exhausted firstly
Wg = (Qmg × 100) / (Σ_{k=1}^{3} Qmk)   ( g = 1,2,3 )  --- equation (2)
} else if( N_tr_network < N_tr_cpu && N_tr_network <
    N_tr_memory){
{
// Network resource is exhausted firstly
Wg = (Qng × 100) / (Σ_{k=1}^{3} Qnk)   ( g = 1,2,3 )  --- equation (3)
}
```

Table 4 shows the pseudo codes for computing the relative resource consumption weights. C is the available CPU capacity. M is the available memory space and N is the available network bandwidth. The index g indicates transcoding grades. In our experiment, three grades are used. The first one is from the 4CIF to the SQCIF grade. The second one is from the 4CIF to the QCIF grade. The third one is from the 4CIF to the CIF grade. The Q_{cg}, Q_{mg} and Q_{ng} are a CPU usage, memory usage and network usage. For example, when a 4CIF MPEG media is

transcoded to SQCIF, QCIF, CIF, the CPU usage Q_{cg} are notated as Q_{c1}, Q_{c2}, and Q_{c3} respectively and the memory usage Q_{mg} are notated as Q_{m1}, Q_{m2}, and Q_{m3}.

In this Table, the W_g means the relative resource consumption weights to each transcoding grade. As new mobile clients arrive, the firstly exhausted resource among three resources dominates the total number of transcoding requests. Therefore, the W_g for each transcoding grade is determined by the firstly exhausted resource. The C/Q_{cg}, M/Q_{mg} and B/Q_{ng} represent the available transcoding requests supplied by the preserved CPU, memory and network resources. Among these values, the smallest one determines the value of W_g in a transcoding server. For example, if the CPU is the firstly exhausted resource, the equation (1) is chosen to compute the W_g. From this equation, the resource weights of transcoding grades are driven by dividing Q_{cg} (the CPU usage of a transcoding grade) by $\sum_{k=1}^{3} Q_{ck}$ (the sum of CPU usages). For example, as described in the Table 3, if a CPU usage to transcode the SQCIF grade is 8.3%, the QCIF grade is 8.5% and the CIF grade is 16.3%, the resource weight for SQCIF grade is 20.08, the QCIF is 25.68 and the CIF is 49.24.

Table 5: Snapshot of Resource Weight Table on Initial Stage

	Transcoding server *A*				
	resource weight	maximum streams	total resource weight	consumed weight	
SQCIF	25	8	200	0	
QCIF	35	7	245	0	
CIF	40	6	240	0	

The Table 5 shows an initial state of RWT. A server *A* is involved in transcoding works. The resource weight column is relative weights driven by equations in Table 4. The maximum stream column is the maximum number of transcoding grades supported by a transcoding server. This value is achieved via actual measurements to each server. The total resource weight is computed by multiplying the resource weight and the maximum stream. This value represents total relative resource weights preserved in a transcoding server to each transcoding grade. The last column represents the consumed resource weights by currently executing transcoding jobs. Because of the initial stage, the consumed weight is zero.

3.3 Load Balance and Admission Control

In the cluster-based server architecture, each server has the same hardware specifications or not. Using the heterogeneous transcoding servers, each server has different resource consumption rates when transcoding jobs are executed. In our RWLD, the characteristic of transcoding servers is reflected on the items of the RWT. Based on the RWT, the fair load distribution can be applied between heterogeneous cluster-based transcoding servers.

When the number of transcoding requests is highly increased, the resources in transcoding servers are exhausted abruptly. As a result, since transcoding servers do not execute their jobs within the limited time, the QoS does not guarantee to all serviced streams. To avoid this phenomenon, the admission control is inevitable whenever a new client is added. If a new transcoding request aggravates the QoS of currently serviced all streams, the admission control should reject the new client request to protect the existing clients. In our RWLD strategy, the load distribution server performs the load balancing role as well as the admission control mission.

Table 6: Pseudo Codes for Admission Control

```
admission_control(transcoding_grade, id)
{        // id: selected server number
// transcoding_grade : transcoding grade requested by clients
int weight, total_weight;

total_weight =
  Server[id][transcoding_grade].weight_data[TOTAL_WEIGHT];
weight =
 Server[id][transcoding_grade].weight_data[RESOURCE_WEIGHT
];

if((Server[id][transcoding_grade].weight_data[CONSUMED_WEIG
HT] + weight) > total_weight)
    return 0; // reject a new client request

    return 1 ; // accept a new client request
}
```

In the RWLD strategy, the maximum number of streams on each transcoding grade is exploited to determine the admission control to every new client. Table 6 shows that the admission control is performed to a new client request. The variable id is the selected transcoding server. The variable transcoding_grade means the transcoding grade requested by the client. The variable total_weight indicates the total resource weight to the requested transcoding grade. If the consumed weight including the new request is over this total resource weight, the selected transcoding server can not process the new transcoding job. In this case, since the new client request can destroy the QoS of currently serviced all clients, it returns 0 as the sign of admission rejection.

4. Experimental Environment

4.1 Cluster-based Transcoding Servers

For experiments, a SMP server is employed as a load distribution server. In addition we deploy diverse

transcoding servers on the basis of the cluster server architecture. The transcoding servers are composed of the 3 kinds of cluster systems. As shown in the Table 7, all nodes within a cluster system have the same hardware specification but each cluster system has different hardware specifications.

Table 7: Specification for Cluster Systems

	Cluster 1	Cluster 2	Cluster 3
CPU	Intel(R) Pentium 4 1.60GHz	Intel(R) Celeron 2.00GHz	AMD Athlon MP 2200+ Dual CPU
Memory	256M SDRAM	1GB SDRAM	1GB DDR
OS	RedHat 7.3	RedHat 7.3	RedHat 7.3
Network	100Mbps fast-ehternet	100Mbps fast-ehternet	100Mbps fast-ehternet
Number of Node	8	8	7

4.2 Performance Measurement Tools

We use the yardstick program to measure the performance of our cluster-based transcoding servers [16]. The yardstick program consists of the virtual load generator and the virtual client daemon.

The virtual load generator is located in the load distributed server. It generates client's transcoding requests based on the distribution of transcoding grades, client's preferences to movies and client's arrival rate. Among the mobile devices, since the cellular phone takes a larger portion, we apply the Zipf distribution with the skew factor 0.271 to the transcoding from 4CIF grade to SQCIF grade [15]. we regard that the popularity of each movie also follows a Zipf distribution with the skew factor 0.271. To the client's arrival rate, we use the Poisson distribution with =0.25[16, 17].

The virtual client daemon locates in test-bed PCs for clients. It plays the role of receiving movie data from transcoding servers. The virtual client is implemented based on our test-bed PCs. In our experiments, it can be found that a PC plays enough roles for 30 virtual clients.

4.3 Maximum Streams by Transcoding Grades

Table 8: Maximum Streams for Transcoding Grades

Grade	A transcoding server of cluster system 1	A transcoding server of cluster system 2	A transcoding server of cluster system 3
SQCIF	8	9	14
QCIF	7	8	12
CIF	5	6	9

For RWLD strategy, it is necessary to measure the maximum number of QoS streams measured in each transcoding server. This measurement is to find the exact number of transcoding jobs involved in the range of not suffering the QoS of all serviced streams. Based on the yardstick program described in the above section, we measure the maximum number of QoS streams by transcoding grades. Table 8 shows the maximum number of QoS streams by transcoding grades in a transcoding server.

5. Performance Evaluation

5.1 CPU Consumption Rates

Figure 1, 2, 3 shows the amount of CPU usage of transcoding servers under RR, DWRR, RWLD strategies. We used 23 transcoding servers involved in 3 kinds of cluster system. On account of space in these Figures, we chose 2 transcoding servers from each cluster system. The A node and B node is from the cluster system 1. The C node and D node belongs to the cluster system 2. The E node and F node is from the cluster system 3.

As shown in the Figure 1, the RR strategy results in the different amounts of CPU usage among transcoding servers. The reason is that transcoding jobs are distributed based on just the arrival order. In particular, since the RR strategy does not distinguish transcoding grades, it allows the overloaded transcoding servers and the underloaded servers to exist together. In the point of 120 clients, the CPU of the server A, C, E becomes saturate as 100% utilization rates, whereas the other servers do not. In addition, we found that the overloaded servers did not support the QoS streams according to the transcoding grades. From our yardstick program, the serviced streams in these overloaded servers were shown the ceasing and jittering phenomenon. The reason is that the admission control can not be applied to the RR strategy. Thus, the overloaded servers could not support streaming services any more. On the other hand, the server B, D, F represents relatively low CPU usage because the light weighed transcoding jobs are allocated to them. As a result, the RR strategy shows unfair load balancing among transcoding servers.

Figure 2 shows the amount of CPU usage of transcoding servers on the DWRR strategy. In the DWRR strategy, transcoding servers send their current resource usages to the load distribution server by periodically. Based on this information, this strategy maintains the load balancing among transcoding servers. If the CPU usage of some transcoding servers reaches 100% utilization, this strategy does not require additional transcoding jobs to these servers. Since the workload congestion to some specific transcoding servers is avoided, the DWRR strategy does not destroy the QoS of all serviced streams. However, the load distribution server has overheads to communicate with transcoding

servers. In addition, since the DWRR strategy uses just the CPU utilization rate as an admission control, it does not reflect the intrinsic characteristic of streaming media in real time requirement. Thus, even if the CPU utilization reaches 100%, the additional transcoding requests could be serviced to clients within a specific range. The DWRR strategy shows fair load balancing among transcoding servers and does not ruin the QoS of all serviced streams.

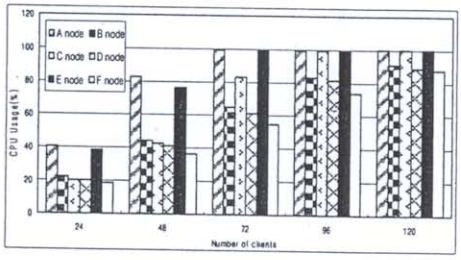

Figure 1: RR (Round Robin) Strategy

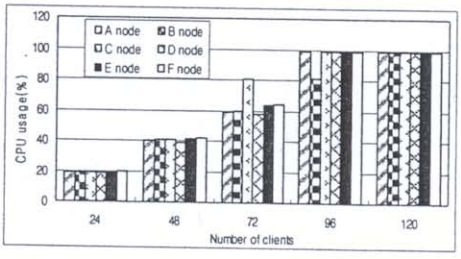

Figure 2: DWRR (Dynamic Weighted Round Robin) Strategy

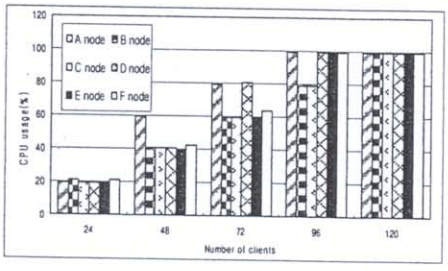

Figure 3: RWLD (Resource Weighted Load Distribution) Strategy

Figure 3 shows the amount of CPU usage of transcoding servers on the RWLD strategy. As shown in this Figure, the RWLD strategy maintains the fair load balancing among transcoding servers like the DWRR strategy. Since the DWRR uses the information of RWT already measured, there are no communication overheads between transcoding servers and a load distribution server. In addition, even if the CPU utilization reaches 100%, the additional transcoding jobs could be accepted in the range of proposed admission control mechanism. Our RWLD strategy contributes the fair load balancing as well as the scalable performance in cluster-based transcoding servers.

5.2 Performance Scalability

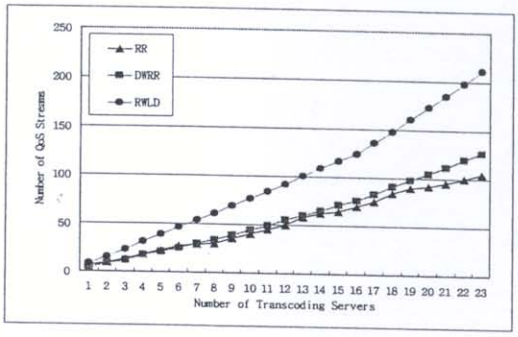

Figure 4: Performance Scalability

Figure 4 shows the total number of QoS streams supported by RR, DWRR, RWLD strategies accordingly as the number of transcoding servers is increased. The QoS is the most important mandatory requirement in the streaming media service. If the QoS requirement does not guarantee in the serviced streams, those streams can not be involved in the total number of QoS streams.

As illustrated in this figure, the maximum number of clients increases proportional to the number of transcoding servers in all strategies. As shown in this Figure, the RWLD strategy has better scalable performance than the RR and the DWRR strategy.

In the RR strategy, the overloaded servers with the congestion of transcoding jobs can not satisfy the QoS requirement. In particular, an additional transcoding job allocated to the saturated servers has a negative impact on other QoS streams being serviced.

The DWRR strategy uses the CPU utilization rate as the metrics of the load balancing and the admission control because the CPU is the fastest exhausted resource in our experiments. This strategy does not consider the minimum amount of CPU consumed for transcoding to the desired transcoding grade. Even if the CPU utilization reaches 100%, it is possible to perform additional transcoding and streaming jobs within the range of satisfying the QoS requirement. The DWRR strategy does not consider this characteristic of streaming media.

On the other hand, the RWLD strategy uses both the resource weight consumed and the maximum number of streams by transcoding grades as the criterion of the load balancing and the admission control. Based on these two types of pre-measured information, this strategy not only fully reflects the intrinsic property of streaming media but

also has no communication overheads to monitor the state information of the resources in transcoding servers. Based on these advantages, even if the CPU utilization reaches 100 %, the RWLD strategy can require the additional transcoding jobs within the range of satisfying the QoS requirement corresponding to each transcoding grade. As a result, the RWLD strategy has been the best scalable performance among the experimented load distribution strategies.

6. Conclusion

In this paper, the load distribution strategies are studied in the cluster-based transcoding servers. The load distribution strategy should provide the fair load balancing and scalable performance. We proposed the RWLD strategy used the actual amount of resources consumed by transcoding grades and the maximum number of QoS streams in transcoding servers.

In our heterogeneous cluster-based transcoding servers, we had evaluated the fair load balancing and the scalable performance of the RR, DWRR and RWLD strategies. Since the RR strategy did not distinguish the transcoding grades for client devices, it showed unfair load balancing among transcoding servers. And also, all streams serviced in overloaded servers did not satisfy the QoS requirement.

The DWRR strategy showed fair load balancing because it used the current resource state of transcoding servers for the load distribution. However, the DWRR strategy passed over the intrinsic characteristic of streaming media in real time requirement. In our experiments, even if the DWRR strategy supported the fair load balancing, it showed the relatively low performance scalability.

The RWLD strategy maintained the fair load balancing among transcoding servers like the DWRR strategy. This strategy used the resource weights and the maximum streams as the criteria of the load balancing and the admission control. By the two types of pre-measured information, this strategy not only reflects the intrinsic property of streaming media but also has no communication overheads to monitor the working state of transcoding servers. From our experiments, since the RWLD strategy performed the admission control based on the QoS requirements of the classified transcoding grades, it showed more linear performance scalability than other strategies.

References

[1] Dinkar Sitaram, Asit Dan, "Multimedia Servers: Applications, Environments, and Design," Morgan Kaufmann Publishers, 2000

[2] W.C. Feng and M. Lie, "Critical Bandwidth Allocation Techniques for Stored Video Delivery Across Best-Effort Networks," The 20th International Conference on Distributed Computing Systems, pp.201-207, April 2000.

[3] D.H.C. Du and Y. J. Lee, "Scalable Server and Storage Architectures for Video Streaming," IEEE International Conference on Multimedia Computing and Systems, pp.191-206, June 1999.

[4] Florin Lahan, Irek Defee, Marius Vlad, Aurelian Pop, Prakash Sastry, "Integrated system for multimedia delivery over broadband ip networks," IEEE Transactions on Consumer Electronics, Vol. 48, No.3, pp.564~565, 2002

[5] H.Bhradvaj, A. Joshi and S. Auephanwiriyakul. "An active transcoding proxy to support mobile web access," In Proceedings of International Conference on Reliable Distrubuted System, pp 118-123, 1998.

[6] Vetro. A.; Sun, H., "Media Conversions to Support Mobile Users," IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), May 2001

[7] A. Iyengar, E. MacNair, and T. Nguyen, "An analysis of web server performance," Computer Networks and ISDN Systems, vol.30, no.1-7, pp.347-357, 1998.

[8] W. Zhang, "Linux virtual servers for scalable network services," Proc. Ottawa Linux Symposium, 2000.

[9] J.P.Chew, A.K. Gupta, "Using Dynamic Weights for Improving Fairness in the ATM ABR Service," 5th IEEE Symposium on Computers and Communications, pp.372-377, July, 2000

[10] http://www.ieee802.org

[11] Sumit Roy, Michele Covell, John Ankcorn, and Susie Wee, "A System Architecture for Managing Mobile Streaming Media Services," 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03), pp.408-419, 2003

[12] J. Song, E.Levy, A.Iyengar, and D. Dias, "Design alternatives for scalable web server accelerators," Proceedings of the 2000 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp184-192, April 2000.

[13] http://www.mpeg.org

[14] http://ffmpeg.sourceforge.net

[15] C.C.Aggarwal, J.L.Wolf, and P.S.Yu, "On optimal batching policies for viedo-on-demand storage servers," Proc. of IEEE ICMCS'96, pp.253-258, Hiroshima, Japan, June 1996

[16] Brian K. Schmidt, Monica S. Lam, J. Duane Northcutt, "The interactive performance of SLIM: a stateless, thin-client architecture," ACM SOSP'99, pp. 31-47, 1999

[17] Surendar Chandra, Carla Schlatter Ellis and Amin Vahdat, "Differentiated Multimedia Web Services Using Quality Aware Transcoding," Proceedings of IEEE INFOCOM Conference, March 2000

[18] C. Li, G. Peng, K. Gopalan, and T. Chiueh, "Performance guarantees for cluster-based internet services", Proceedings of the 23rd International Conference on Distributed Computing Systems, pp378- 385, May 2003

[19] J. Guo, F. Chen, L. Bhuyan, and R. Kumar, "A cluster-based active router architecture supporting video/audio stream transcoding services", Proceedings of the 17th International Parallel and Distributed Processing Symposium, pp. 446- 453, April 2003