An Auto-Adjustment Video Mechanism for Steadily Monitoring of Remote Laboratory

doi:10.3991/ijoe.v5i4.977

Jian-Wei Lin
National Taiwan University of Science and Technology, Taipei, Taiwan

***Abstract*—This work presents an auto-adaptable video streaming Web-based system, enabling on-line simultaneously monitoring of multiple instruments of remote experiment. The system is capability of not only handling multiple concurrent stream processes, but also automatically tuning encoding parameters of the running video streaming processes once overload so that the whole system can always keep in a smooth and steady circumstance.**

***Index Terms*—Video Streaming; Electronic Instrumentation.**

I. INTRODUCTION

Remote instrument control has played an important role in industry and distance learning since it provides users flexible way to operate and learn instrument control at any time and from any location [1-3]. In order to let users have more realistic feeling of the current conditions at operating the remote instruments, the best way is to provide the users with not only a virtual control panel, but also a live (real-time) video capturing of the real panel of instruments [4-6]. Thus to observe the variations of real panel of each instrument, each instrument should be equipped with a digital camera. However, according to our previous work [6], monitoring of multiple instruments certainly taxes on significant system resources and loading since concurrent video stream processes have to run at the system.

Thus, to keep the system loading within rational scope, this work presents a video streaming system with auto-adaptable capability. By continuously measuring the system loading, the system adopts proper control mechanism and is able to dynamically tune the encoding parameter of the running streaming processes in order to keep it to work steadily all the time.

next section briefly describes the design architecture while Section III demonstrates our prototype. The testing experiments of the system are reported in Section IV. Conclusions are given in Section V.

II. PROPOSED DESIGN ARCHITECTURE

The proposed software structure, shown in Fig. 1, comprises two layers: Web Application and Video Stream Component. The technologies and methodology of these two layers are described in detail as follows:

A. Video Stream Component

The Video Stream Component focuses on constructing and distributing video streaming for electronic instrumentation. Thus several important works have to be carried out in this component: detecting all connected video ac-

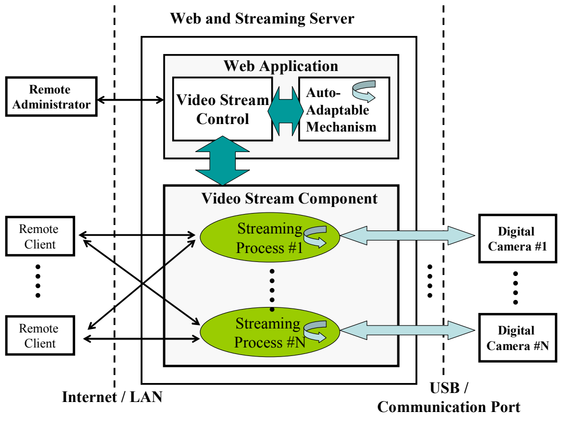


Figure 1. The proposed software architecture

quisition devices, capturing video content from the devices, encoding the video contents, and sending the compressed video content over the Internet. For supporting individual streaming process for viewing each instrument, an extra data structure in this component is declared for each stream process to store its own encoding profile, such as video device name, network protocol and the used port number, encoding mode (e.g. Constant Bit Rate or Variable Bit Rate), encoder (e.g. MPEG4, H.263 or Window Media Video), frame size, frame rate, CBR output bit rate, and VBR quality level.

In a VBR mode, a desired quality level, ranging from 0 to 100, is required to specify. Under a given quality level, the output bit rate after encoding fluctuates depending on the complexity of the stream. That is, high motion will result in a high bit rate and slow motion will result in a lower bit rate. The benefit of quality-based VBR encoding is that the video quality is consistent across all streams. The drawback is that bandwidth requirement of the encoded content can not be predicted. In a CBR mode, an output bit rate is required to specify. Accordingly, the encoding quality is not constant for content which complexity varies. The benefit of CBR encoding is that the output bit rate of the content is known before encoding. Therefore, the bandwidth requirement of the encoded content can be predicted.

The sequential procedures inside each streaming process are to capture an instrument video, encode it with its own profile (e.g. encoder, frame rate, frame size), and then deliver it through network protocol, such as Real-Time Transport Protocol (RTP) or HTTP.

This Video Stream Component (i.e. Microsoft COM) is developed based on the Microsoft Windows Media Encoder Software Development Kit (SDK) which is a multimedia framework for media creation and distribution within Microsoft Windows [7]. The SDK provides useful API to facilitate encoding and delivering video media. Besides, since the SDK supports HTTP to deliver the video stream, HTTP is used as the protocol to transmit the video stream.

B. Web Application

This part consists of two main modules: Auto-Adaptable Mechanism (AAM) and Video Stream Control (VSC).

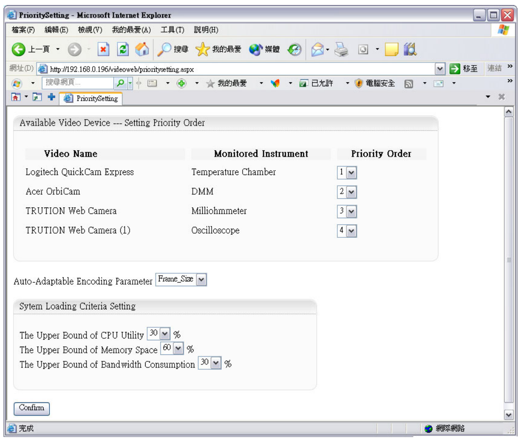
The AAM, which continuously measures the system loading (e.g. bandwidth, CPU and memory consumption) where measuring interval is three seconds herein, is responsible for keeping system loading to meet the pre-determined overload criteria arranged by the administrator. The overloading criteria include the upper bound (threshold) of bandwidth consumption and CPU utilization and memory space. Once detecting one of their measured value exceed its upper bound more than the specified times (e.g. ten times herein), we define that such condition is overload happened. Afterward AAM will automatically adjust encoding parameters (e.g. frame size and frame rate) of the streaming processes via VSC in iterative way so that the system loading will eventually meet the pre-determined criteria. In other words, once overload, the streaming processes will be adjusted one by one in a given priority sequences, which is also pre-ranked by the administrator, till the loading (e.g. CPU utilization) drops to its desired range. Currently, two encoding parameters, frame size and frame rate, are available for adjustment. Frame sizes is varied in the range [640x480, 320x240, 160x120] while frame rates is varied in the range [30, 25, 20, 15, 10].

The VSC aims at offering the system administrator to control and manage live video services (e.g. launching or stopping a video stream), depending on experimental requirements. For example, the administrator can enlarge the frame size or upgrade video quality for a clearer view of an instrument display panel. However, once overload detected, the VSC will degrade the frame rate (or frame size) of the first priority stream (e.g. frame rate from 30 to 25). If still overload, this downgrading procedure will repeat till the adjusted frame rate of the first priority stream process has reached bottom line (i.e. 10). However, if overload still being detected, the system will switch to the second priority stream process to degrade its frame rate once again till the frame rate has reached bottom line. Such procedures are employed iteratively on all streaming processes until the system loading is under the pre-determined criteria setting. However, if overload still rises after the frame rate of all streaming process have reached bottom line, the AAM then switch to degrade the frame size of the streaming processes in accordance with the same procedure above till meeting with criteria requirements.

Likewise, if frame rate is specified as the auto-adaptable parameter, the mechanism will first degrade frame rate from 640x480, 320x240 to 160x120 iteratively instead of degrading frame size first.

III. PROTOTYPE

Currently, our prototype consists of one Web server and four USB webcams. After an authorized administrator


PrioritySetting - Microsoft Internet Explorer
http://192.168.0.196/videoweb/prioritysetting.aspx
PrioritySetting

Available Video Device --- Setting Priority Order

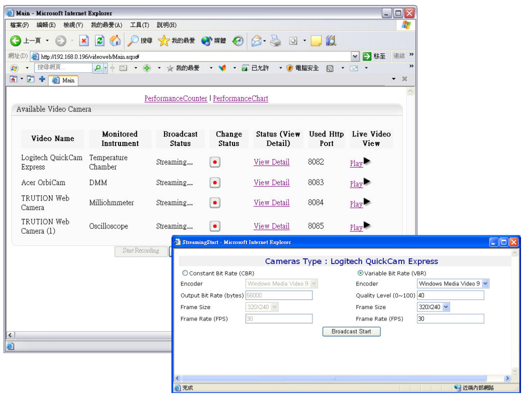
Video Name	Monitored Instrument	Priority Order
Logitech QuickCam Express	Temperature Chamber	1
Acer OrbiCam	DMM	2
TRUTION Web Camera	Milliohmmeter	3
TRUTION Web Camera (1)	Oscilloscope	4

Auto-Adaptable Encoding Parameter Frame_Size

Sytem Loading Criteria Setting

The Upper Bound of CPU Utility 30 %
The Upper Bound of Memory Space 60 %
The Upper Bound of Bandwidth Consumption 30 %

Figure 2. The Web page of setting the system loading criteria


Main - Microsoft Internet Explorer
http://192.168.0.196/videoweb/Main.aspx#
Main
PerformanceCounter | PerformanceChart
Start Recording
StreamingStart - Microsoft Internet Explorer
Broadcast Start

Available Video Camera

Video Name	Monitored Instrument	Broadcast Status	Change Status	Status (View Detail)	Used Http Port	Live Video View
Logitech QuickCam Express	Temperature Chamber	Streaming...	●	View Detail	8082	Play ▶
Acer OrbiCam	DMM	Streaming...	●	View Detail	8083	Play ▶
TRUTION Web Camera	Milliohmmeter	Streaming...	●	View Detail	8084	Play ▶
TRUTION Web Camera (1)	Oscilloscope	Streaming...	●	View Detail	8085	Play ▶

Cameras Type : Logitech QuickCam Express

Constant Bit Rate (CBR)		[✓ Variable Bit Rate (VBR)]	
Encoder	Windows Media Video 9	Encoder	Windows Media Video 9
Output Bit Rate (bytes)	56000	Quality Level (0~100)	40
Frame Size	320X240	Frame Size	320X240
Frame Rate (FPS)	30	Frame Rate (FPS)	30

Figure 3. The Web pages of managing video streaming (upper left) and encoding profile setup (bottom right)

logs into the system successfully, one administrative page, shown in Fig. 2, reveals the setting for the system loading criteria which have to be settled before streaming processing running. In this figure, the priorities for these video devices are set from 1st to 4th; the frame size is selected as the auto-adjusting encoding parameter. The upper bounds of CPU and bandwidth consumption are settled in 30% while that of memory consumption is 60%.

After finishing criteria setting, another administrative page, shown in upper-left page in Figure 3, will list all attached video devices for managing the video services, allowing launching or stopping a stream of webcam, looking the statistics of a live stream, and previewing a stream. In an attempt to launch a video stream associated to an instrument, the encoding profile (e.g. CBR or VBR mode) has to be set. As shown in bottom-right page in Figure 3, encoder, output bit rate, frame size, and frame rate need to be settled down in a CBR mode, while encoder, quality level ranging from 0 to 100, frame size, and frame rate need to be settled down in a VBR mode. This quality level supported in SDK is used to represent the quality of viewing a video.

IV. SYSTEM EXPERIMENT

To evaluate the system feasibility and its Auto-Adaptable Mechanism, some experiments have been tested. Among these investigations, two different encoding modes, CBR and VBR, are adopted in order to individually analyze their impacts. In these experiments, the upper bounds of CPU and bandwidth consumption are

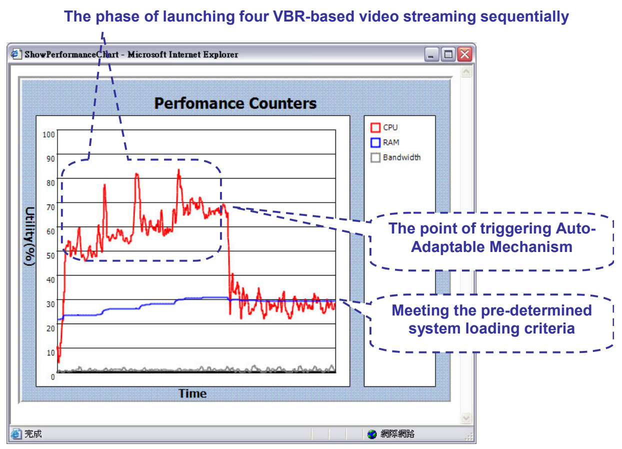


Figure 4. The experiment result in VBR mode

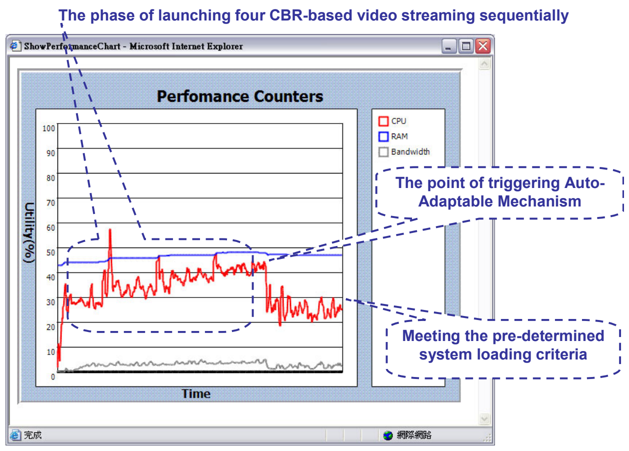


Figure 5. The experimental result in CBR mode

settled in 30% while that of memory consumption is 60%. In this case, frame rate is specified as the auto-adaptable parameter.

A. In VBR Mode

In this mode, four video streaming are launched sequentially, all of whose frame size and quality level are set 640x480 and 60 respectively. Besides, each streaming delivers video to two viewers to evaluate their bandwidth.

Figure 4 manifests the certain duration of the system performance pertaining to CPU, memory (RAM) and bandwidth consumption representing in red, blue and gray line respectively. The Y-Axis title is consumption ranging from 0 to 100 % while the X-Axis title is the duration of monitoring the loading since system launching. The experimental result reveals that the number of activated processes has less impact on bandwidth and memory consumption comparing to CPU. However, while triggering auto-adaptable mechanism, CPU utilization instantly drops below the pre-determined upper bound (30%). In fact, the AAM has degraded the frame size of the first priority process from 640x480 to 160x120 and that of the second priority one from 640x 480 to 320x240.

Notably, the second of launching a video stream will cause a CPU utilization pulse, resulting in that there are several peak values of CPU utilization during the phase of launching the four processes in Figure 4. However, the CPU utilization will trend toward stably after a while.

B. In CBR Mode

In this mode, the testing environment is the same to the previous experiment except that output bit rate of CBR mode is settled in 1024 Kbps. Figure 5 also exhibits that the number of activated processes has much more impact on CPU utilization. While triggering auto-adaptable mechanism, CPU utilization also instantly drops below 30%. In fact, the AAM has only degraded the frame size of the first priority process from 640x480 to 160x120. However, the trend of Figure 4 varies more dramatically, comparing to Figure 5, since keeping consistent highest quality of all streams will heavily consume CPU computing. The CPU usage of four concurrent processes already reaches around 70% for the VBR quality level of 60 while around 40% for CBR with 1024Kbps output bit rate.

V. CONCLUSION

This paper presents an auto-adaptable streaming system that handles multiple concurrent streaming processes and enables clients to view live video of their interested instruments. Based on continuously measuring the system's loading, the encoding parameters of the streaming processes are able to automatically be tuned in an iterative way in order to restrain system from overload. Based on the proposed two-layer architecture, it becomes easily and efficiently to achieve this aim.

REFERENCES

[1] C. C. Ko, B. M. Chen, S. Hu, V. Ramakrishnan, C. D. Cheng, Y. Zhuang, and J. Chen, "A Web-based Virtual Laboratory on a Frequency Modulation Experiment," IEEE Transactions on Instrumentation And Measurement, vol.31, no.3, pp.295-303, Aug. 2001.

[2] Ferrero, S. Salicone, C. Bonora, and M. Parmigiani, "ReMLab: A Java-Based Remote, Didactic Measurement Laboratory," IEEE Transactions on Instrumentation And Measurement, vol.52, no.3, pp.710-715, June 2003. (doi:10.1109/TIM.2003.814695)

[3] J. Guo, D. Kettler, M. Al-Dahhan, "A chemical engineering laboratory over distributed control and measurement systems," Computer Applications in Engineering Education, Vol. 15, NO.2, 2007, pp.174-184 (doi:10.1002/cae.20108)

[4] C. C. Ko, B. M. Chen, S. H. Chen, V. Ramakrishnan, R. Chen, S. Y. Hu, and Y. Zhuang, "A Large-scale Web-based Virtual Oscilloscope Laboratory Experiment," IEE Engineering Science and Education Journal, Vol.9, No.2, pp.69-76, April 2000. (doi:10.1049/esej:20000204)

[5] N. Ranaldo, S. Rapuano, M. Riccio, and F. Zoino, "On The Use of Video-Streaming Technologies for Remote Monitoring of Instrumentation," IEEE Instrumentation and Measurement Technology Conference, Italy, pp.861-866, April 2006.

[6] J. W. Lin and Y. C. Lai, "A Manageable Web-based Video Streaming System for On-Line Monitoring of Remote Laboratory Experiment," IEEE International Instrumentation and Measurement Technology Conference, pp. 1150-1154, Canada, May 2008.

[7] Microsoft Inc., "Windows Media Encoder 9 Series SDK Help," http://www.microsoft.com.

AUTHORS

Jian-Wei Lin is with National Taiwan University of Science and Technology, Taipei, Taiwan (e-mail: D9109104@mail.ntust.edu.tw).

Submitted 23 June 2009. Published as resubmitted by the authors on 2 November 2009.