

System for capturing, streaming and sharing video files

DAVID MALANÍK, ZDENĚK DRBÁLEK, TOMÁŠ DULÍK, MIROSLAV ČERVENKA
Faculty of Applied Informatics
Tomas Bata University in Zlín
Nad Stráněmi 4511, 76005 Zlín
CZECH REPUBLIC

Abstract: - This paper describes a modular system, which provides capability for capturing videos and screen casts from lectures and presentations in any academic or commercial environment. The system uses client-server architecture. Client node sends streams from available multimedia devices to local area network. The server provides functions for capturing video from streams and for distributing the captured video files using torrent. For protecting the video files, the web-based user interface provides authentication and authorization similar to the traditional Unix file permissions system.

Keywords:- video, video capture, video streaming, client-server application, modular system, .NET, HTML, PHP, MySQL, Python Apache2, torrent, tracker, web application programming

1 Introduction

The main idea of this work was to create a modular system, which provides video streaming and capturing functions for Tomas Bata University in Zlín. However, we did not want to implement a brand new system. Rather, we extended and integrated several existing applications into single system providing advanced features. This system is being developed as a free alternative to MediaSite system, which is offered by the Sonic Foundry company [1].

One of our main requirements was to have our system based entirely on open source components. Therefore, it includes only free software projects. It provides the following functions:

- streaming video from available inputs to local area network or Internet,
- video recompression,
- video capture from various inputs or a network stream,
- video file archiving,
- video file distribution via the torrent network.

The system is fully modular, which significantly simplifies its extension by new functions. The following features are under development at this moment:

- video recompression module - captured files can be distributed in multiple resolutions, encoded by various codecs and qualities,
- module providing synchronisation of user database with other learning systems such as Moodle [2] or STAG [3].

2 Problem formulation

Our goal is to create a complex system providing the following services:

- video streaming to LAN or Internet,
- video capturing from multiple sources (cameras, screen grabbers or a network stream),
- video recompression using arbitrary codecs,
- archiving the captured video on a server,
- video sharing via the torrent technology,
- user account and role administration,
- modular HW architecture,
- modular SW architecture

3 Problem solution

As we are building an IT system based on the free components, we choose the following technologies as the system's foundations:

- Apache HTTPD for web interface,
- PHP 5,
- MySQL 5 database as data and configuration source,
- Python (utilities providing torrent functions),
- modified BitTornado torrent client,
- GNU/Linux as server OS,
- VideoLanClient [4] as multimedia core, providing recompression, streaming and video recording.
- .NET framework for desktop demo application.

You can see the client-server hierarchy of the proposed system in Fig. 1.

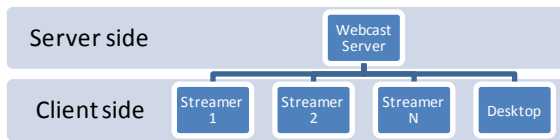


Fig. 1 - Master structure of system

The whole system is split in two parts: the upper part is the “Webcast Server” while in the lower part there are the client “Streamer” units.

The “Webcast Server” provides functions for archiving, capturing and distributing the video via the local area network or Internet. Configuration settings and information about video files are stored in the MySQL database, which stands for the central data repository of the whole system. Because video capture and recompression are very CPU-intensive operations, the Webcast server can also be deployed on a computer farm if needed. Suggested minimal HW requirements are shown in Tab. 1.

HW Part	Specification
CPU	>4 cpu cores (>8)
RAM	>8GB
HDD	>4TB for Video files in RAID
Graph. Card	Onboard
LAN	>100Mbps (>1Gbps)
IP	Private - for local access Public - for Internet access

Tab. 1 - Minimal HW settings for Webcast Server

The Streamer unit provides functions for capturing the video from all the available sources and for its streaming. The Streamer can run on a lightweight PC, a typical thin client computer would do the work - minimal HW specification is shown in Tab. 2. The Streamer machine does not need any storage media, it can boot from network. The machine can have only modest CPU because the streaming task are quite undemanding.

HW Part	Specification
CPU	1,2GHz Intel Celeron
RAM	2GB DDR2 800MHz
HDD	CF Card / SSD HDD
Graph. Card	Onboard
LAN	100Mbps

Tab. 2 - HW setting for Streaming unit

The Streamer unit has two variants. The first option is a mobile Streamer, which is described above. This unit is connected to Webcast server database and share its configuration settings and status with the server. Available sources for mobile Streamer can be seen in Fig. 2.

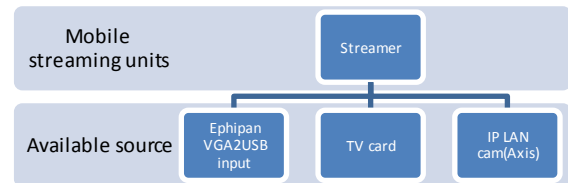


Fig. 2 - Available video sources for Streamer

The second option is a desktop application based on the .NET Framework named “Capture”. It was developed especially for online/offline streaming/capturing. The Capture application stores its configuration settings and all information about video files in xml files. It can stream video from sources listed in Fig. 3. It also supports offline media creation. User can record a screencast and upload it later onto the Webcast server, making it available to all students.

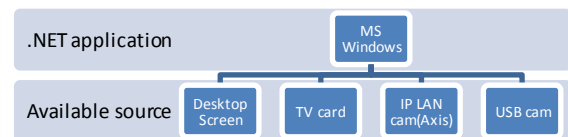


Fig. 3 - Available video sources for .NET application

3.1 Webcast server

This is the central node, which consists of web server with multimedia database. It stores all data. Its purpose is to provide fast video processing and distribution using the torrents. The reason for using the peer-to-peer torrent architecture lies in comparison with classics path of the distribution. In case we had an internet server with upload speed of 10MiB/s, we would be able to download 1GiB file in approximately 100s. If this file is downloaded by 100 users at the same time, the download speed for 1 user would sink to circa 0,1MiB/s, taking 10 000 seconds (2.77 hours) to fetch the requested file. Using the torrent technology, we can bypass this limitation in the way that even 100 users can get a stable downstream at

10MiB/s or even faster. This is due to parallel data exchange among the users who already have this file and therefore the server does not to provide all the data to every single user.

Differences between these two methods are shown in Tab. 3 and Tab. 4, in Fig. 4 and Fig. 5 respectively. We used the following formulas to calculate the transfer speed:

Direct download :

$$\text{Mbps/user} = (\text{LAN}/\text{Users}) * \text{Crop. F.} \quad (1)$$

Torrent download

$$\text{Mbps/user} = ((\text{LAN}/\text{Users}) * \text{Crop. F.}) + ((\text{users} - 1) * \text{LAN} * \text{Crop. F.}) \quad (2)$$

where *LAN* stands for network bandwidth that all users share and *Crop. F.* is a factor transforms theoretical bandwidth to real.

Users	Mbps/user	LAN	Crop. F.
1	80	100	0.8
10	0.8	100	0.8
20	0.4	100	0.8
30	2.67	100	0.8
40	2	100	0.8
50	1.6	100	0.8
100	0.8	100	0.8
200	0.4	100	0.8
400	0.2	100	0.8
800	0.1	100	0.8
1600	0.05	100	0.8

Tab. 3 - Bandwidth with direct download

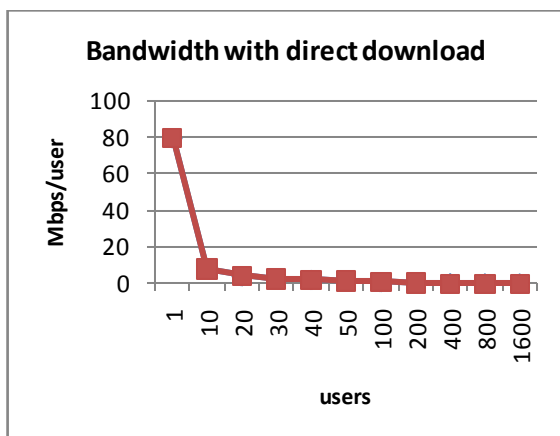


Fig. 4 - Bandwidth with direct download

Users	Mbps/user	LAN	Crop. F.
1	80	100	0.8
10	728	100	0.8
20	1524	100	0.8
30	2322	100	0.8
40	3122	100	0.8
50	3921	100	0.8
100	7920	100	0.8
200	15920	100	0.8
400	31920	100	0.8
800	63920	100	0.8
1600	127920	100	0.8

Tab. 4 - Bandwidth with direct download

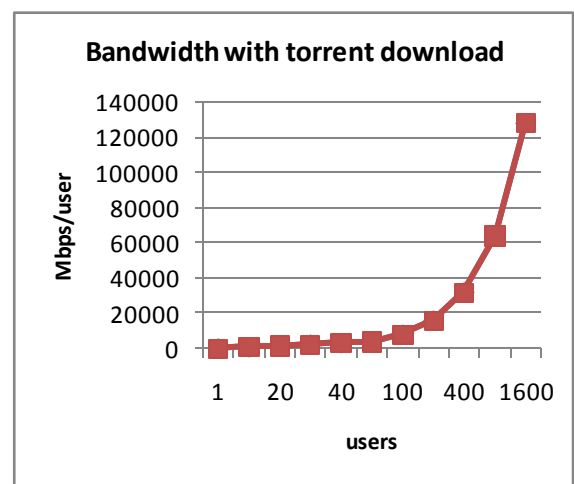


Fig. 5 - Bandwidth with direct download

Webcast server consists of the following modules:

Module Stream – this module enables administration of active stream in the system. User can view which streams are active and see information about these streams. Streams whose capturing is started from Streamer are added to system and published automatically. Other streams can be added to the system using the web interface of this module. This module also provides functionality for adding, editing and erasing the streams.

Module Lectors – this module is intended for lectors/users administration. Database of lectors is stored in the MySQL database. This module has functions for creating xml files from database data, which can be later used in the Capture for offline processing. There is also an advanced functionality for re-importing these xml files back with an intelligent synchronisation. This module also provides functions for adding, editing, erasing, exporting and importing lectors.

Module Subjects – this module represents administration of available subjects taught on the university. It provides all needed functions for adding, editing, erasing, exporting and importing subjects.

Module Workstations – this module is designed for administration of workstations in system. A workstation might be a Windows desktop with application for capturing videos, streamer unit or IP camera. This module keeps a complete database of all available streaming units. The web interface of this module brings the user single sign-in functionality to all registered devices for accessing its generated streams and their potential recording. Further functions are adding, editing and erasing workstations from the list.

Module Files – wraps all administration of video files in the system. Its web interface displays all information related to a specific file – creator, subject, theme, date and time when this video has been recorded and, its availability. Based on the user's role, it is possible to modify this information. There is direct way how obtain the media files using this module – it only provides either a torrent or an xml file. Module also provides functions for uploading, editing and erasing video files in system.

Module Tracker – gives access to administration of torrent files in system. It provides functions for generating torrents from a video file, starting their distribution and a tracker function. This module can work as both, the torrent tracker and the torrent client. When a video file is already shared by a higher number of students, it can stop distribution from the central node (Webcast server) and thus save some bandwidth. Security settings prevent the tracker responding to unknown torrents, which are not registered in the database. Further options include setting of the maximum speed for download, upload, and transfer ports and other. It is possible to create, start stop and erase torrents in the system. This is master module for video file distribution.

Module Record – this module is for recording a stream from network. At this moment, one media recording can consist of three simultaneous streams, but virtually, this number is not limited. Stream recordings can be annotated by additional information (like lecturer, subject, title and other). Recording can be started either directly from this module or automatically by Streamer. It is possible to define a maximal length of a stream, after which its recording stops. Only after this event is the video file visible for users, however, they can access the stream in the moment when it is broadcasted. This module is not

primarily designed for video recompression; it only stores the incoming streams with additional information about its source. It provides functions for starting and stopping recordings in the system.

All events in the system are logged and it is possible to track down any activity, which happened in the system. A custom IP filter can prevent users to log in from unauthorised networks and this list can be easily modified by competent super-user.

3.2 Mobile streaming unit

Mobile streaming unit (Streamer) represents an independent platform solution for capturing on other platforms than Windows. This tool consists of a web server, which provides an interface to VLC. The VLC [4] recompresses and forwards data from various sources (desktop image, various cameras and other devices) to the Webcast server.

All the modules of Webcast server are compatible with Streamer. Streamer alone contains only one module.

Module Streaming – this module provides all stream-related functions on Streamer. There are two modes of streaming. The former simply streams captured data to network without knowing who consumes it. On the Webcast server you can see open streams and you can attach them to one or multiple recording. The latter mode creates a stream with direct recording on the Webcast server.

3.3. Desktop application (Capture)

This application provides offline functions for video streaming and capturing. It uses xml files for storing information about recorded video files. These xml files can be uploaded to the Webcast server. This tool is primarily designed for offline video capturing from available sources, like for example screen on desktop PC, a USB or an IP camera. Capture in cooperation with VLC [4] can easily record user's desktop to file or stream it to network. This is an excellent solution for creating screen casts. A lecturer can make his tutorial offline at home and upload it to server later. The Capture application was created as a demonstrator presenting the immense variability of the proposed system.

4 Conclusion

This document brings a brief overview of a system for recording lectures and other educational activities on the Tomas Bata University in Zlin. We described the modular architecture and functionality of the proposed system, which is now in state of beta testing. Our intention was to create a free, zero cost alternative to MediaSite. Now we are finishing the first phase of development with expected heavy testing. Further modules extending the system functionality are about to come after the users' feedback and additional requests that we expect to come with the testing process.

Acknowledgements:

This work was supported by the EU Sixth Framework Programme under the project iCamp IST-027168. This assistance is very gratefully acknowledged.

References :

- [1] MediaSite [online] Available from URL:<
<http://www.sonicfoundry.com/products/mediasite/>>
- [2] Moodle [online] Available from URL:<
<http://moodle.org> >
- [3] STAG [online] Available from URL:<
<http://stag.zcu.cz> >
- [4] VideoLanClient(VLC) [online] Available from
URL:< <http://www.videolan.org> >