Desarrollo de una Aplicación Cliente/Servidor para un Wall View en base a la Plataforma – Cruzada Opensource – FFMPEG (Colección de Software Libre que puede Grabar, Convertir y hacer Streaming de Audio y Vídeo)

Montenegro C.*; Mullo C.; Samaniego C.***; Calderón X.******
** Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica*
Quito, Ecuador (e-mail: christian.montenegro@mailfie.net)
*** Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica*
Quito, Ecuador (e-mail: cesar.mullo@mailfie.net)
**** Escuela Politécnica Nacional, Departamento de Informática y Ciencias de la Computación*
Quito, Ecuador (e-mail: gustavo.samaniego@epn.edu.ec)
***** Escuela Politécnica Nacional, Departamento de Electrónica, Telecomunicaciones y Redes de Información*
Quito, Ecuador (e-mail: xavier.calderon@epn.edu.ec)

Resumen:*Actualmente se nota un considerable crecimiento en el uso de pantallas gigantes o dispositivos de proyección, para presentar videos pertinentes a publicidad, educación y afines. Sin embargo, el alto precio de dispositivos dedicados exclusivamente a la reproducción multimedia, se torna como una limitante para la implementación de este tipo de sistemas; de allí nace esta iniciativa que plantea opciones que minimicen en cierta medida los costos asociados al software de desarrollo, y a los dispositivosadicionalesque se necesitan para montar un Wall View.*

La finalidad del proyecto es presentar una aplicación cliente-servidor basada en software libre, que permita implementar un Wall View construido con monitores comunes, siendo importante tomar en cuenta la sección de video que cada uno debe representar;además se recurre al uso de herramientas libres como una alternativa viable que permite obviar el costo de licencias para el desarrollo de la misma.

El aplicativo desarrollado es capaz de mostrar en el Wall View un video educativo, informativo y/o de publicidad, un evento en vivo a través de una webcam, o una captura de escritorio, desde el servidor.

Palabras clave:*Aplicación Cliente- Servidor, FFMPEG, Wall View.*

Abstract: *Actually thereis a considerable growth in the use of screens or projection devices,which present videos about advertising, education and related. However, the expensive price of dedicated devices to multimedia playback turns into a limiting factor for the implementation of such systems, hence comes this initiative in order to provide options that reduce some of the associated costs.*

The goal of this project is showing a client-server application based on free software, which allows implementing a View Wall built with common monitors, considering the video section that everyone should represent too, combining the use of free tools as a viable alternative that allows avoiding the cost of licenses for software development.

The developed application is able to show in the Wall View an educational video, informative and/or advertising, a live event via webcam, or a desktop capture from the server.

Keywords:*Client-Server Application, FFMPEG, Wall View.*

1. INTRODUCCIÓN

La arquitectura cliente – servidor [2] es la que más se ajusta a los requerimientos del proyecto; es decir que la aplicación desarrollada emplea un esquema de red simple de dos capas. Complementariamente se debe destacarla amplia acogida que tiene el uso de pantallas gigantes o dispositivos de proyección en el sector publicitario, educativo, empresarial e incluso personal.

Este proyecto tiene como base la plataforma-cruzada FFmpeg, apoyado a su vez en las características propias del sistema operativo GNU/Linux. Sin embargo, es pertinente señalar que existen ciertas consideraciones y configuraciones importantes para el correcto funcionamiento de la aplicación, como por ejemplo la fuente de la señal de video, para

dividirla y procesarla usando las herramientas de FFmpeg, y luego ser presentada en un arreglo de monitores que conforman el Wall View. Asimismo, no se deben perder de vista los requerimientos de hardware apropiados para un desempeño óptimo de dicha aplicación, a la cual se la ha denominado FFWallView.

1.1. FFMPEG[8][11]

FFmpeg es una herramienta rápida en línea de comandos que permite convertir archivos de audio y/o video de un formato a otro, capturar y codificar en tiempo real desde distintas fuentes como una webcam o una tarjeta de TV;e incluso realizar un screencast (captura de audio y videodel escritorio). Además es posible manipular parámetros de audio y video como imágenes/cuadros por segundo (frames per second, fps), resolución, relación de aspecto, tasa de bits, compresión, número de canales de audio, entre otros[6]. En la Figura. 1 se ilustran las aplicaciones de FFmpeg.

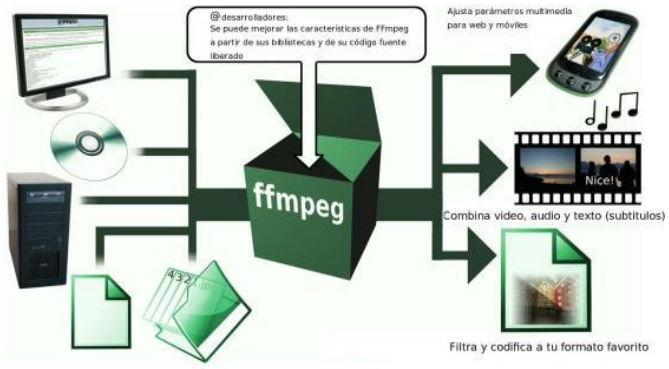

Figura 1. Aplicaciones de FFmpeg.

A. Descripción del Funcionamiento[3]

El diagrama de la Fig. 2 describe el proceso de codificación y decodificación que realiza FFmpeg.

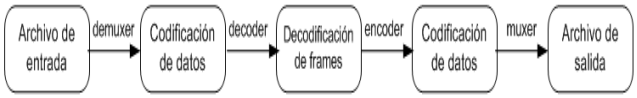

Figura 2. Funcionamiento de FFmpeg.

1.2. FFSERVER[5]

FFserver es un servidor de streaming [1] de audio/video que viene embebido dentro de la plataforma FFmpeg y que está habilitado para recoger varias fuentes de entrada (normalmente aplicaciones FFmpeg), para luego transcodificar, remultiplexar, y/o difundir cada una de ellas utilizando múltiples flujos de salida, distribuyéndolos a uno o varios clientes para su visualización (ver Fig. 3).

A. Descripción del Funcionamiento[10]

El desempeño de FFserver se ajusta a las peticiones de los clientes de manera predefinida o dinámica a través de un archivo de configuración simple "/etc/ffserver.conf".

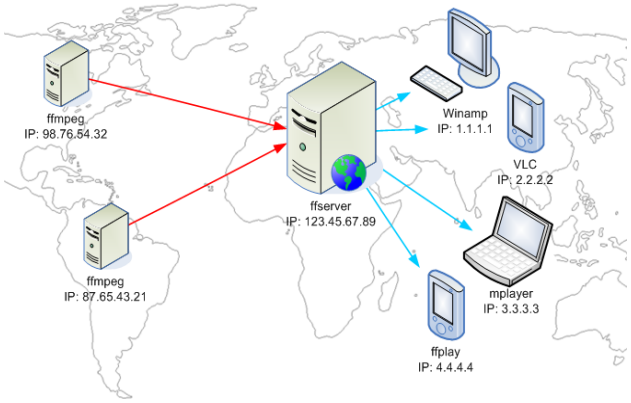

Figura 3. Funcionamiento de FFserver.

FFserver presenta una ligera desventaja con respecto al menor número de códecs que soporta, entre ellos AVI, FLV y MPEG. Sin embargo, esto no representa una limitante crítica o de consideración que impida el correcto funcionamiento de un streaming de audio/video.

1.3. FFPLAY[4][9]

FFplay es un reproductor multimedia ligero que, al igual que FFserver, viene embebido en FFmpeg (ver Fig. 4). Por esta razón, no existen inconvenientes de compatibilidad en el esquema global "streaming-procesamiento-reproducción" de la aplicación para conformar el Wall View.

Figura 4. FFplay.

A. Descripción del Funcionamiento

FFplay es capaz de reproducir casi cualquier formato de audio y video (combinados o por separado), ya que cuenta con varias bibliotecas que manejan apropiadamente la mayoría de códecs de este tipo de archivos multimedia.

2. METODOLOGÍAY DISEÑO

La metodología de desarrollo de software seleccionada para el diseño e implementación de la aplicación fue la Metodología Incremental [12].

2.1. Consideraciones

1. Para el análisis de requisitos se consideró fundamentalmente las tareas principales que debe cumplir la aplicación:
   - Capturar la información multimedia (básicamente audio y video).
   - Segmentar el video en función del número de monitores (empleando las opciones propias de la herramienta FFmpeg).
   - Hacer uso del sistema operativo para controlar las tarjetas de video pertinentes, lo cual corresponde a un procesamiento y ajuste en la imagen.
   - Codificar los esquemas pertinentes de audio y video.
   - Insertar la información dentro del servidor.
   - Realizar la transmisión al cliente (con previa verificación de conectividad), respondiendo al tipo de petición del mismo.
2. En la etapa de diseño se consideró el esquema general (ver Fig. 5) que debía manejarse tanto en el cliente como en el servidor.

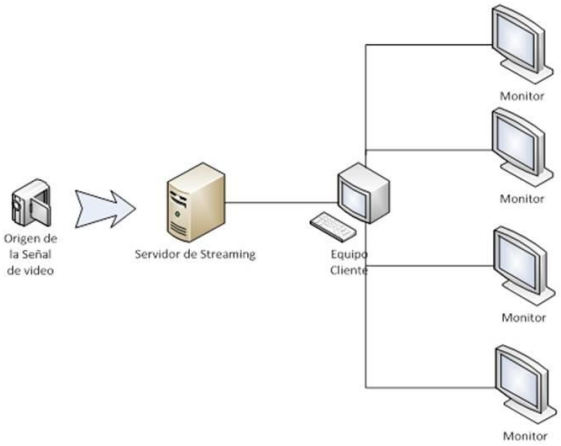

Figura 5. Esquema general de diseño para FFWallView.

3. En la programación el conflicto central respondía a la problemática sobre qué lenguaje de programación utilizar. En consecuencia, desde la primera iteración, se optó por el lenguaje bash script puesto que éste mantiene la estructura simple de un intérprete de comandos;es decir, se mantiene la misma sintaxis de la CLI (Interfaz de Línea de Comandos) de Linux, lo que representa una baja latencia en la respuesta de las directrices que se le da a la aplicación.Adicionalmente se imbricó dicho lenguaje con el utilitario Zenity, esto con la finalidadde proveer una interfaz gráfica sencilla que interacciona a través de ventanas de diálogos con el cliente (usuario final).
4. Finalmente, en función del avance del proyecto, se realizaron pruebas para evaluar el desempeño de la aplicación en cada iteración. En total se realizaron 5 iteraciones (ver Fig. 6), considerando la existencia o no de criterios insatisfactorios en cada una de ellas, hasta definir el prototipo final.

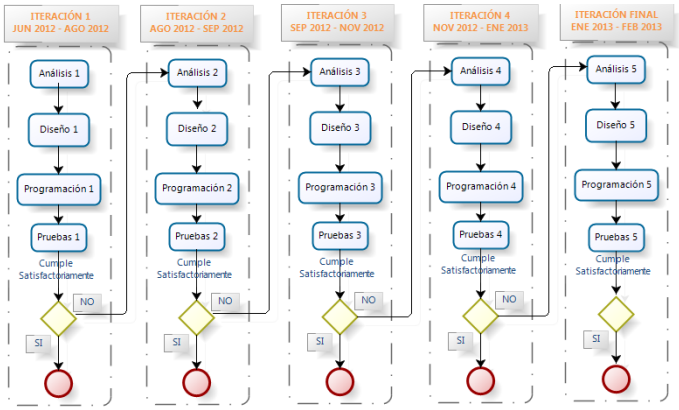

Figura 6. Esquema general de iteraciones para FFWallView.

En el servidor se observan básicamente cuatro etapas (ver Fig.7):

- Selección del modo de operación.
- Selección del modo de transmisión.
- Prueba de conectividad (hacia el cliente). En caso de que no recibir respuesta exitosa desde el cliente, se retorna al punto inicial de dicha prueba para ingresar nuevamente los parámetros correspondientes.
- Transmisión del streaming (hacia el cliente).

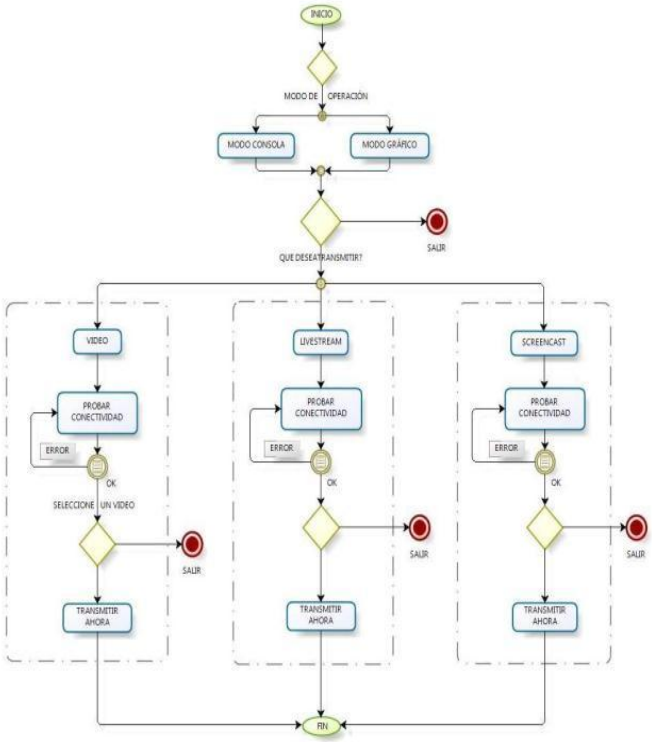

Figura 7. Diagrama de flujo del servidor FFWallView.

Homólogamente en el cliente se distinguen cinco etapas (ver Fig.8):

- Selección del modo de operación.
- Selección del modo de transmisión.
- Prueba de conectividad (hacia el servidor). En caso de no recibir respuesta exitosa desde el servidor, se retorna al punto inicial de dicha prueba para ingresar nuevamente los parámetros correspondientes.
- Selección del modo de presentación (para el Wall View).
- Reproducción o presentación del streaming en el Wall View.

Recordar que el streaming se envía ininterrumpidamente hacia el cliente, lo cual implica que su reproducción (en el Wall View) consecuentemente sea continua.

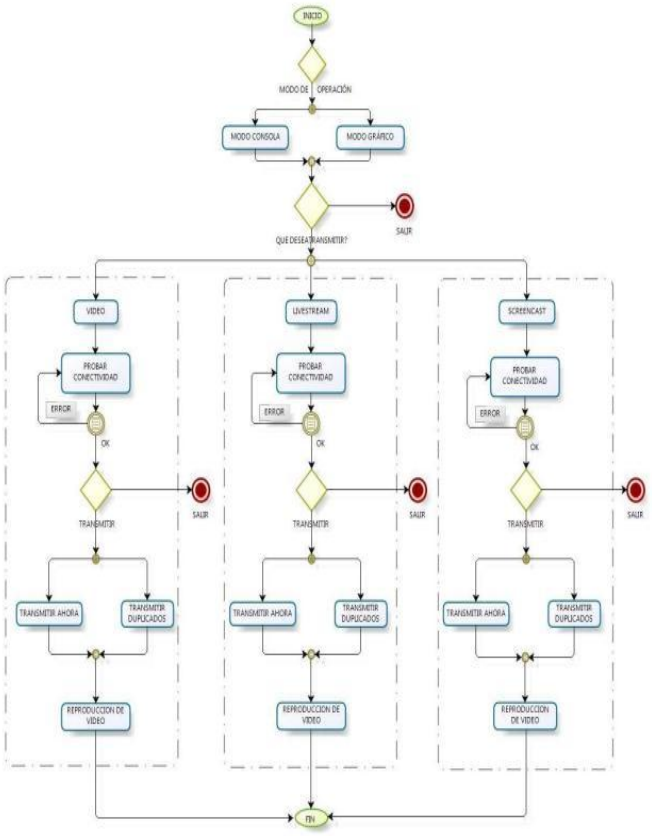

Figura 8. Diagrama de flujo del cliente FFWallView.

2.2. Dimensionamiento del servidor[7]

Se utiliza la ecuación (1) para calcular el tamaño de un video y estimar el espacio de almacenamiento que ocupará en el disco duro.

$$Tamaño\_Video = Duración\ (s) x bitrate\ (Kbps) \quad (1)$$

Con la ecuación (2)se calcula en bit rate.

$$Bit\ rate[Kbps] = \frac{(R\ x\ fps) x\ (fm) x 0.07}{1000} \quad (2)$$

Dónde:

- *R: resolución del video, o número de pixeles (ancho x alto).*
- *fm: factor de movimiento = 1, 2 ó 4.*
- *fps: frames por segundo.*

Considerando las siguientes características de video y utilizando (2), calculamos el bit rate:

- Resolución = 1360 x 768
- Frames por segundo = 24
- Factor de movimiento = 2

$$Bit\ rate = (1360 x\ 768 x\ 24 x\ 2 x\ 0.07)/1000$$

$$Bit\ rate = 3509.4\ Kbps = 3.5\ Mbps$$

Considerando:

- Duración del video 5 minutos.
- Tasa de bits 3.5 Mbps.

De(1) se obtiene:

$$Tamaño\_Video = 5\ \text{min}\ x\ \frac{60\ segundos}{1\ \text{min}}\ x\ 3.5\ Mbps\ x\ \frac{1 Byte}{8\ bits}$$

$$Tamaño\_Video = 131.25\ MB$$

Tras estos cálculos, en la Tabla 1 se muestra un aproximado de la capacidad de almacenamiento que se necesitaría en disco duro, tanto para la distribución de Linux, como para la aplicación FFWallView y sus utilitarios adicionales.

Tabla 1. Resultados de cálculo para capacidad de almacenamiento

APLICACIÓN	CAPACIDAD DE ALMACENAMIENTO
Ubuntu óFedora	10 GB
Videos (considerando 1000)	131,25 GB
Aplicación FFWallView y utilitarios de software	< 50 MB
Total	141,3 GB

Tarjeta de red.

Con la ecuación (3) se calcula el ancho de banda que requiere la aplicación para la transmisión de un video.

$$\text{Ancho de Banda} = \frac{\textit{Tamaño del Video}}{\textit{Tiempo de Duración}} \qquad (3)$$

Se considera un video con las siguientes características para utilizar (3):

- Formato mp4.
- 5 minutos de duración.
- Un espacio en disco de aproximadamente 131 MB.

$$\text{Ancho de Banda} = \frac{131\,MB}{5\ \text{min}}\ x\ \frac{1\ \text{min}}{60\ s}\ x\ \frac{8\ bits}{1\ Byte} = 3.5\ Mbps$$

Con dichas estimaciones de tráfico que maneja la aplicación y considerando las tarjetas de red existentes en el mercado actual, que ofrecen tasas de transmisión de 10/100/1000 Mbps, no se prevé inconvenientes en este sentido.

- Fuente de poder.
  Debe proveer la energía suficiente para el correcto funcionamiento del CPU y de sus dispositivos adicionales como las tarjetas de video, sin que sufra daños y/o averías por recalentamiento.
- Unidad de disco óptico.
  Necesaria para copiar información al disco duro desde dispositivos externos CD/DVD.
- Sistema operativo.
  Si bien FFmpeg es una plataforma cruzada, la aplicación está diseñada para sistemas operativos basados en Linux.

Con lo expuesto anteriormente y considerando las tendencias actuales (a nivel tecnológico – informático) en el mercado ecuatoriano, en la Tabla 2se resumen los requerimientos para el servidor FFWallView.

Tabla 2 Hardware recomendado para el servidor FFWallView.

DESCRIPCIÓN	CANTIDAD
Computador de Escritorio: Procesador Intel® Core i7, Disco Duro de 1 TB, Memoria RAM de 4 GB, Tarjeta de red 100/1000 MB, y Periféricos de E/S	1
Tarjeta de Video DualHead	1
Monitor	1

3. FFWALLVIEW

FFWallView es la aplicación que surge tras varias iteraciones en función de los diagramas de flujo que se muestran en la Fig. 7 y en la Fig. 8.

Es importante mencionar que, de manera concisa, se ha dado tratamiento a los esquemas de conectividad y a las eventualidades que surgen en el streaming de audio y video, tomando en cuenta los protocolos apropiados para lograr una transmisión exitosa. La verificación de conectividad se realiza con el comando *fping*, mientras que para el streaming se emplea el protocolo *UDP*.

3.1 Descripción del Funcionamiento

FFWallView es una aplicación basada en FFmpeg que es capaz de presentar un video educativo, informativo y/o de publicidad, un evento en vivo a través de una webcam, o una captura de escritorio, desde un servidor destinado para tales efectos. En este sentido se tienen tres modos de operación de FFWallView (ver Fig. 9):

- Transmitir Video: Este es el modo de operación principal de la aplicación, donde se transmite un video almacenado en el servidor.
- Transmitir LiveStream: Este modo de operación permite transmitir un evento en vivo capturado por una webcam desde el servidor.
- Transmitir ScreenCast: Este modo de operación en cambio facilita la transmisión de una captura de escritorio desde el servidor.

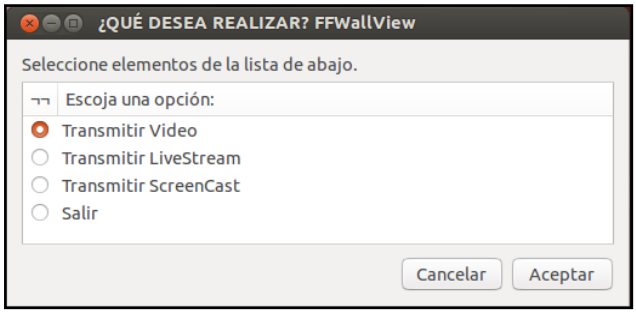

Figura 9. Menú principal de FFWallView.

4. PRUEBAS Y RESULTADOS

Se debe recalcar que la aplicación FFWallView fue probada en una matriz de cuatro monitores (2x2), obteniendo una resolución en conjunto de 2720x1536 (1360x768 en cada monitor).

Los comandos empleados para realizar el streaming (servidor) y la presentación en los monitores (cliente) son:

ffmpeg -i "ENTRADA" -f mpegtsudp://"Dirección IP cliente":"N° de puerto" #SERVIDOR[3]

ffplay -vfcrop="largo":"ancho":"posiciónx":"posición y" -window_title "Título de la ventana (opcional)" udp://"Dirección IP servidor":"N° de puerto"#CLIENTE[4]

Cabe señalar que para la reproducción del streaming se debe considerar la sección de video que cada monitor deberá mostrar.

A. Modo de Operación "Transmitir Video"

En la Fig. 10 se puede apreciar la transmisión de un video publicitario almacenado en el servidor.

Figura 10. Transmisión de Video.

B. Modo de Operación "Transmitir LiveStream"

En la Fig. 11 se observa el resultado de una transmisión de un evento en vivo (video-conferencia) a través de la webcam del servidor. Los espacios vacíos (negros) corresponden a zonas sin información; esto debido a la resolución nativa que maneja la webcam.

Figura 11. Transmisión LiveStream.

En este modo de operación, tras varias correcciones, finalmente se percibe, un desfase de 3 a 4 segundos entre audio y video; esto en función de los elementos empleados en dicha prueba.

C. Modo de Operación "Transmitir ScreenCast"

En la Fig. 12 se muestra una captura de escritorio desde el servidor, mientras se reproduce un video musical en pantalla completa.

Figura 12. Transmisión ScreenCast.

De forma general se puede decir que FFWallView muestra un desempeño adecuado, donde cada modo de operación arroja los resultados esperados.

5. CONCLUSIONES

FFmpeg utiliza varias bibliotecas de códecs de audio y video, permitiendo de esta forma tener soporte para una amplia variedad de formatos comerciales, por ejemplo los manejados por Apple QuickTime (.MOV), por RealNetwok (.RV, .RAM, .RM, .ODER .RMVB), o por Microsoft Windows Media Player (.WMV); por tal motivo, FFmpeg inicialmente podría ser considerada como una herramienta netamente orientada a edición multimedia. Sin embargo, se debe recordar que FFmpeg va más allá, siendo una completa plataforma que abarca tareas tan sencillas como transformar el formato de un archivo de audio/video, hasta tareas complejas como realizar un streaming administrando flujos y puertos de salida, en base a protocolos de comunicación en tiempo real.

FFWallView en base al uso adecuado de herramientas como FFmpeg, hace posible brindar una solución para la implementación, incluso doméstica, de un Wall View capaz de mostrar una gran cantidad de información con un alto impacto visual; escenario que antes era manejado como tecnología exclusiva para operaciones de monitoreo de tráfico aéreo, estaciones de televisión, y sistemas de seguridad y vigilancia; es decir que dicha tecnología tenía como destinatarios principales a empresas con capacidad de realizar inversiones significativas, debido a los costos y complejidad de toda la infraestructura física que esto implica.

FFWallView muestra una ventaja económica frente a las soluciones comerciales, ya que alser una alternativa basada en software que no se ve limitada por las características del hardware, implica que se puede trabajar con cualquier tipo de pantalla (LCD, LED, CRT, plasma) de diferentes tamaños y resoluciones, con

cualquier tipo de PC, y con interfaces de red Ethernet convencionales; sin incurrir en cuantiosos gastos para su implementación.

FFWallView, con la ayuda de aplicaciones sencillas de software libre como Zenity, brinda la posibilidad de contar con una interfaz gráfica simple y amigable con el usuario final; es decir, a diferencia de ciertas alternativas comerciales, presenta una gran facilidad en su manejo y utilización.

Para verificar el desempeño de la aplicación se estableció el uso de cuatro monitores que conformen el Wall View, sin embargo es posible realizar un escalamiento con el objetivo de utilizar más monitores, teniendo como único condicionante la utilización de tarjetas gráficas (comunes) pertinentes para dicho efecto.
Una restricción de FFWallView, es el intervalo típico de 1 a 5 segundos de desfase entre el envío y recepción de un streaming de audio/video, lo cual se debe al tiempo que demora FFmpeg en estabilizar la tasa de frames por segundo "fps" con la que se envía un archivo multimedia por la red. La tasa por defecto considerada por FFmpeg es 25 fps, y toma alrededor de 4 segundos en estabilizarse.

Para los modos de operación "LiveStream" y "ScreenCast" de FFWallView, donde el streaming de audio y video se realiza por separado, FFmpeg siempre da prioridad al tratamiento del audio, independientemente del número de flujos de video que se estén enviando al mismo tiempo por la red local.

En el modo de operación "LiveStream", tras varias correcciones, finalmente se percibe, un desfase de 4 segundos entre audio y video; esto como limitante en función de los elementos empleados en dicho modo de operación; es decir, los resultados mejorarían significativamente de utilizarse dispositivos de audio y video (micrófono y webcam) con mejores características.

FFWallView está diseñada de tal forma que quienes operen los equipos cliente y servidor, no deban configurar parámetros adicionales dentro de la aplicación, es decir, desde la óptica del (los) usuario(s) final(es), FFWallView cumple de manera fácil y sencilla las tareas para las cuales fue programada.

Por su versatilidad y gracias al impacto visual que generan los muros de pantallas en el público, FFWallView está orientada a varios ámbitos de aplicación, como educación, publicidad, información, entretenimiento, entre otras.

6. RECONOCIMIENTO

En este punto es preciso extender un agradecimiento especial en memoria del Ing. Pablo Salinas (†), quien nos extendió la primicia sobre la temática del proyecto que se describe en este documento.

7. REFERENCIAS

[1] MACKIE, David, "Streaming Video & MPEG4IP", Cisco Technology Center, 2002.
[2] COMER, Douglas, STEVENS, David, "Internetworking with TCP/IP Vol III: Client-Server Programming and Applications", Prentice-Hall International, Inc., 2nd Edition, 1998.
[3] FFMPEG DEVELOPERS, Linux Man Page, "ffmpeg", versión 1.0.
[4] FFMPEG DEVELOPERS, Linux Man Page, "ffplay", versión 1.0.
[5] FFMPEG DEVELOPERS, Linux Man Page, "ffserver", versión 1.0.
[6] FFMPEG DEVELOPERS, Linux Man Page, "ffprobe", versión 1.0.
[7] EZS3, "What bitrate should I use when encoding my video?, how do I optimize my video for the web?", [Online] http://www.ezs3.com/public/What_bitrate_should_I_use_when_encoding_my_video_How_do_I_optimize_my_video_for_the_web.cfm
[8] FFMPEG homepage, "FFmpeg", [Online]. Disponible en: http://ffmpeg.org/ffmpeg.html
[9] FFMPEG homepage, "FFplay", [Online]. Disponible en: http://ffmpeg.org/ffplay.html
[10] FFMPEG homepage, "FFserver", [Online]. Disponible en: http://ffmpeg.org/ffserver.html
[11] GRAPHCOMP, "Grafman's FFMPEG Developer Info", [Online]. Disponible en: http://graphcomp.com/ffmpeg/
[12] UNIVERSIDAD DE ORIENTE, Venezuela, "Metodologías para el desarrollo de software", [Online]. Disponible en:http://wiki.monagas.udo.edu.ve/index.php/Metodologías_para_el_desarrollo_de_software