

Setting up a Video Wall with NMM

Motama GmbH, Saarbruecken, Germany
(<http://www.motama.com>)

April 2010

Copyright (C) 2005-2010
Motama GmbH, Saarbruecken, Germany
<http://www.motama.com>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being all sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found in the file COPYING.FDL.

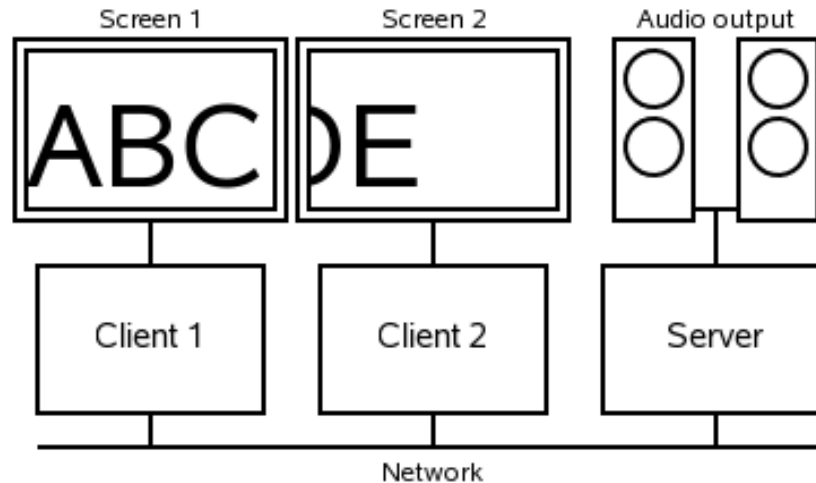
THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE DOCUMENT BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT.

A video wall consists of several video screens, such as TVs, CRT monitors, or LCDs, that are placed on top of each other or side by side. Each of the available screens is configured to only show the section of a video frame that corresponds to its position within the array of screens. Therefore, a video frame -- or each frame of a video stream -- to be shown is split into different sections. This allows to use the overall screen size of all available devices to create a large screen.

This document describes how to set up a video wall based on commodity PC technology using NMM. In particular, this demonstrates how easy it is to set up such a distributed multimedia application using the tool 'clic' provided by NMM. A video of a running video wall can be found here (ftp://graphics.cs.uni-sb.de/www/NMM/Status/Videowall/NMM_videowall_oct2004.avi) (14 MB).

1. Using NMM's clic for setting up a video wall

Figure 1. Video wall using a 'server' for rendering audio, and two 'clients' for rendering the different cropped parts of a single video stream.



The tool 'clic' allows to set up distributed multimedia applications from a textual description provided in a .gd file. The content of such a .gd file is called graph description. This graph description specifies a distributed flow graph of NMM nodes. Each node can transparently be located on a remote host. NMM also handles the inter-stream synchronization of all media processing and rendering on distributed hosts. For example, when running a video wall, all parts of a video frame have to be rendered synchronously on all used systems. Furthermore, lip-synchronization between audio and video needs to be provided.

Instead of using an application for a server and another client application running on each system that is connected to a display, NMM only needs a 'serverregistry' to be running on each system. Only a single main application is used.

1.1. Requirements

You need a running NMM installation on a least two networked systems that are both connected to some screen. You also need the clocks of the systems to be synchronized sufficiently, e.g. using NTP, the network time protocol. This step is described in a separate document provided at the NMM homepage. You also need an NMM serverregistry running -- at least on all remote systems. More precisely, you need to provide

full access to X for the serverregistry applications running on all systems. The easiest way to achieve this is to start the serverregistry in a terminal running within your window manager.

1.2. Two-screen video wall

The following shows the `dist_video_wall.gd` file to be used for a simple video wall consisting of two systems.

```
DVDReadNode      $ select(1, 1, 1)      OUTPUT_INITIALIZED
! MPEGDemuxNode
{
  { ["ac3_audio0"] ! AC3DecodeNode
                                ! PlaybackNode
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode
                                ! VideoCropNode      $ setCrop(0,0, 384, 288) INITIALIZED
                                ! XDisplayNode       $ setFullscreen(1) ACTIVATED
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode # setLocation("asterix")
                                ! VideoCropNode # setLocation("asterix")
                                                $ setCrop(384,0,384, 288) INITIALIZED
                                ! XDisplayNode # setLocation("asterix")
                                                $ setFullscreen(1) ACTIVATED
  }
}
```

In particular, data is read from a DVD. The 'select' method allows to choose the title, chapter, and angle of the DVD. The first AC3 audio stream is decoded and played out on the local system. The first MPEG video stream is decoded, cropped, and rendered on the local system and on a remote system named 'asterix'. On the local system, only the left part of the video stream is shown, namely a region of 384 pixels width and 288 pixels height starting from (0,0); on the remote system, the adjacent region is shown starting from (384,288).

To start clic using this graph description simply type

```
./clic dist_video_wall.gd
```

This graph description is only meant as starting point for further experiments. For example, try to change the cropped section of the video stream, or change the location of all nodes of the branch of the flow graph that decodes and renders the audio stream. In addition, you can employ a different source node, e.g. for

reading an audio/video file. Using NMM and NMM's clic, different configurations can be tested within seconds.

1.3. Two-screen video wall with overview on third screen

If you want to render the complete video stream on a third system, say system 'obelix', this is also very easy to achieve.

```
DVDReadNode      $ select(1, 1, 1)      OUTPUT_INITIALIZED
! MPEGDemuxNode
{
  { ["ac3_audio0"] ! AC3DecodeNode
                                ! PlaybackNode
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode
                                ! VideoCropNode      $ setCrop(0,0, 384, 288) INITIALIZED
                                ! XDisplayNode       $ setFullscreen(1) ACTIVATED
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode # setLocation("asterix")
                                ! VideoCropNode   # setLocation("asterix")
                                                $ setCrop(384,0,384, 288) INITIALIZED
                                ! XDisplayNode    # setLocation("asterix")
                                                $ setFullscreen(1) ACTIVATED
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode # setLocation("obelix")
                                ! XDisplayNode   # setLocation("obelix")
                                                $ setFullscreen(1) ACTIVATED
  }
}
```

As you can see, we only added an additional video branch that is not cropped for the system 'obelix'.

A photo showing such a setup can be seen below. Since the DVD used in this setup provides 16:9 video, different cropped regions were used. As you can see from this example, adding additional systems to your video wall is also very easy using NMM.

1.4. Dynamically adding systems using NMM's session sharing service

Instead of statically configuring such a video wall, NMM also allows to dynamically add systems by using the facilities provided by the session sharing service. In short, the session sharing service allows to 'share' parts of a running flow graph within different applications; media processing and rendering is automatically synchronized for all participating flow graphs.

To enable session sharing, you have to add two lines to the .gd file, the 'setSharingType' commands:

```
DVDReadNode      # setSharingType(EXCLUSIVE_THEN_SHARED)
                  $ select(1, 1, 1) OUTPUT_INITIALIZED
! MPEGDemuxNode  # setSharingType(EXCLUSIVE_THEN_SHARED)
{
  { ["ac3_audio0"] ! AC3DecodeNode
                    ! PlaybackNode
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode
                    ! VideoCropNode      $ setCrop(0,0, 384, 288) INITIALIZED
                    ! XDisplayNode      $ setFullscreen(1) ACTIVATED
  }
  { ["mpeg_video0"] ! MPEGVideoDecodeNode # setLocation("asterix")
                    ! VideoCropNode      # setLocation("asterix")
                                          $ setCrop(384,0,384, 288) INITIALIZED
                    ! XDisplayNode      # setLocation("asterix")
                                          $ setFullscreen(1) ACTIVATED
  }
}
```

For using the session sharing service, you need a running serverregistry on every system. Then, start clic as usual using this graph description.

```
./clic dist_video_wall_shared.gd
```

Now you can dynamically add (and remove) additional systems by using following graph description called dist_video_wall_shared2.gd

```
DVDReadNode      # setSharingType(SHARED)
! MPEGDemuxNode  # setSharingType(SHARED)
{
  { ["mpeg_video0"] ! MPEGVideoDecodeNode # setLocation("obelix")
                    ! VideoCropNode      # setLocation("obelix")
                                          $ setCrop(192,0,384, 288) INITIALIZED
                    ! XDisplayNode      # setLocation("obelix")
                                          $ setFullscreen(1) ACTIVATED
  }
}
```

Just start a second instance of clic:

```
./clic dist_video_wall_shared2.gd
```

In this case, an additional screen is opened on 'obelix' that shows a different cropped section of the video in fullscreen mode. The session sharing service automatically shares the source node for reading the DVD and the MPEG demultiplexer. Notice that you can stop clic any time and restart it again, e.g. using different cropping parameters. To this end, the session sharing service of NMM greatly helps to set up such complex scenarios without doing any programming at all.