About CineGrid

CineGrid’s[1] mission is to build an interdisciplinary community focused on the research, development and demonstration of networked collaborative tools, enabling the production, use and exchange of very high-quality digital media over high-speed photonic networks.

High quality digital media?

High quality digital media can be very beneficial in a scientific environment where people need very detailed images. Certain compression formats would for example throw away crucial information from the image, which is why uncompressed formats or at least lossless formats are very important. 4K media (4096 x 2160 pixels) is currently used by CineGrid, which allows for very detailed images even on big screens.

Image below: People enjoying jpeg2k compressed 4K. When you stand close to the screen like that, you can really see for yourself just how sharp the image is.

Obviously, this kind of media wouldn’t look bad in cinemas either...

Image below: The Amsterdam CineGrid node web portal

Contact Info: Sander Knopper <sknopper@science.uva.nl>, Ralph Koning <raiph@science.uva.nl>, Paola Grosso <p.grosso@uva.nl>, Cees de Laat <delaat@uva.nl>

References

Content Exchange Architecture

These are the main components of our CineGrid Context Exchange solution:

- **Portal**
  The portal is a web-based user interface for the end-user to start streaming content. The user can choose what content should be streamed and can also choose a format. The last thing the user should provide is a display where the content can be actually shown.

- **Metadata server**
  The metadata server serves as a database of metadata. It currently collects this using simple XML descriptions which are coupled to content files. But due to its modular design anything can be used to obtain metadata such as RDF or even iRODS[2]. By using systems such as notify (available under GNU/Linux) the metadata server automatically updates when new descriptions are added or existing ones are modified.

- **Stream server**
  The stream server takes requests from the end-user and tries to start the streams. In its current state it holds a queue per destination address and just plays the content in the queue in first come, first-serve order. It also has a modular design, which in this case means that it potentially can support a large number of media formats and will choose the right media player for the right format.

- **Display**
  The display will most of the time be something that the end-user has the provide, since it’s most probably located at a location the end-user controls. Concerning the display itself there are several possibilities, it can be a tiled-panel display, a projector or a single display. In terms of software there are also several options, we currently support SAGE[3] and VLC[4] setups, but it’s relatively easy to extend this.

The portal itself is written using the Django[5] Python-based web framework. It uses a database system which actually serves as a cache. The database pulls information from one or more metadata servers using XML-RPC and then stores it so the portal can access the information in a fast way. The user is then able to choose a specific film from the portal and start it. At that point an XML-RPC is being used to instruct the stream server what it should stream and where to.

Eventually the goal is to interact with the network as well, making it possible to dynamically allocate light paths for streaming content with high bandwidth needs. Though, at this moment static light paths are already a critical part of the infrastructure to be able to stream high quality media like this without hick-ups and other annoyances that could negatively influence the user experience.