Grid File System Scenario

A Grid File System provides a simple, hierarchically structured view of diverse data resources, along with service interfaces to access or transfer the data to clients. A client may be a Grid-enabled browser or an interface to the Posix I/O API, or even another service.

Summary

This scenario extends the functionality and workflow presented in Scenario 3, Data Pipelining. That scenario presents a workflow associated with a digital imaging production process, noting that it is common now for 3D animations to be very complex and include numerous files, and also for the image rendering process to happen on rendering(compute) clusters separate from the desktop machine used by the scene designer. In this scenario, an equally common but even more complex configuration is envisioned, in which the animators workstation and associated data resource, the rendering compute and data resources, a technical/creative directors workstation, a post-processing compute resources, and the corporate clients presentation video and data resources, are all involved in a single workflow. This apparent single point-of-access must be achieved even though each human, compute, and data resource may be located in a different geographical region, may be based on different hardware with different capabilities, and may be in separate administrative domains.

Scenarios

1. A Grid File System service is created and configured for a specific project, commissioned by a corporate client of a digital imaging firm. The GFS service, utilizing the Resource Namespace Service, provides a unified directory hierarchy containing subdirectories for all the commonly-accessed components of the digital-imaging process, from 3D models and textures to animation sequences, test renders, renderings of “layers” of a scence, and final composited images. The GFS also provides a single mechanism of access to all the data resources referenced by it; in this case, that mechanism is generally Posix I/O.

2. An animator builds an animation definition file with a commonly-used tool such as Alias’ Maya, utilizing models created by modelers, and textures created by texture painters. The models and textures may be accessed through data services which serve data from the shared fileservers used by the modelers and texture painters, but are presented to Maya as simple files in “models” and “textures” subdirectories of the GFS hierarchy, and are accessed by Maya through the standard Posix I/O API. The final animation is to be rendered in “layers”, with separate components of the scene rendered separately (for example, backgrounds may be rendered first and then used as reflection-maps for foreground objects)

3. The animator submits an animation rendering task to her companies render/compute cluster, which divides up the layers and the frames of the animation and farms them out

to separate nodes within the cluster. The animation description itself and its data dependencies, such as textures and models, are accessed using the same GFS-managed path and access mechanism that she used when creating the scene, although each may actually be retrieved using various data transfer services associated with resources local to each employee. The results are written directly to the cluster's local high-performance parallel filesystem such as IBM's GPFS,

4. As they are completed, the high-resolution, high-quality layer files are registered with a Replica Location Service, and their RLS addresses are registered with the GFS server, which in turn creates one or more RNS virtual directories to contain references to the rendered images (junctions in RNS terminology). All modern rendering applications support this kind of post-render callout function.

5. As the layer-rendering process is completed for individual frames, a technical director examines the files for accuracy and conformance to the clients needs by launching single compositing processes on a compositing/post-processing compute cluster, which returns the completed frames to the storage server at the technical director's office using a data transfer service and then registers the files with the GFS service. The technical director may go through multiple iterations of this compositing process, tuning parameters such as transparency, layer-order, and color-correction of each layer, until she is satisfied. The creative director also examines the composited frames, from the same location within the GFS hierarchy, and determines that the animation is ready for batch-composition followed by presentation to the client.

6. The technical director then launches a batch composition process on the post-processing compute cluster, this time using data transfer services to transfer the files both to the companies main archival storage system as well as to the high-performance file systems used for presentation at the external client site. A separate RNS service is used to manage the primary archival storage area, and a new reference to the RNS directories containing the final images and component layer images is created within the project GFS service, with access controls applied through the GFS so that the corporate client can access only the required presentation materials, and none of the possibly sensitive other materials stored on the companies main archival system.

7. Using an RLS and data transfer services, duplicates of the large, high-quality finished image sequence are created on the clients high-performance file system, which is directly connected to a high-resolution projection system. The final image sequence is viewed by the corporate client during a meeting with the digital imaging company creative and technical directors, and the client is given the ability, using the GFS service, to browse the relevant final product, as well as the component layers used to construct the final images, Once this infrastructure is in place, it can remain in place as the design and implementation details for the animation are refined.

Involved Resources

Grid File System Service provides a front-end to the client, managing underlying services for directory management, service selection, and data transfer.

The Namespace Service provides mappings of entries in a directory hierarchy (names) to data services, which may be simple data transfer services or more complex replication services. It also provides properties associated with each directory entry.

Registry or replication services may be used to provide multiple locations for data transfer for some or all of the files in the "file system"

Data Transfer Services are used to provide data from data resources to the GFS service or directly to the client

Functional Design

(see above for much of the functional design – major to do is to reorganize this document to put functional design detils here instead of in the scenario section.)

Also, need to make graphs here for the functional design.