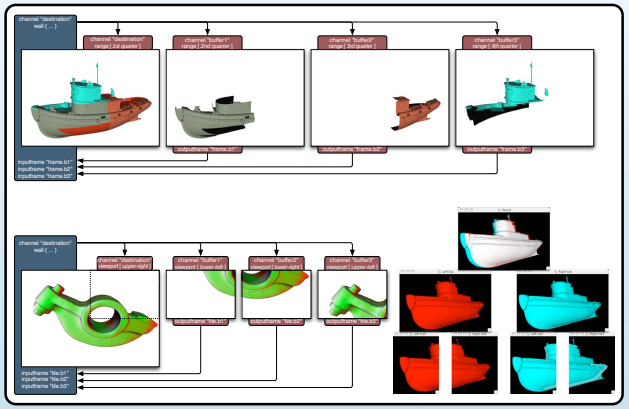
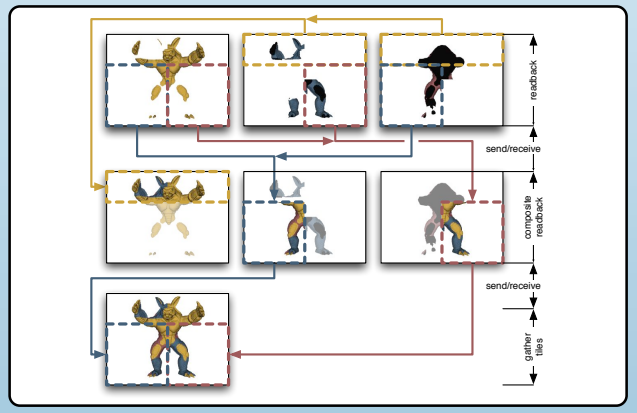


Sort-last (DB), sort-first(2D) and multi-level compounds



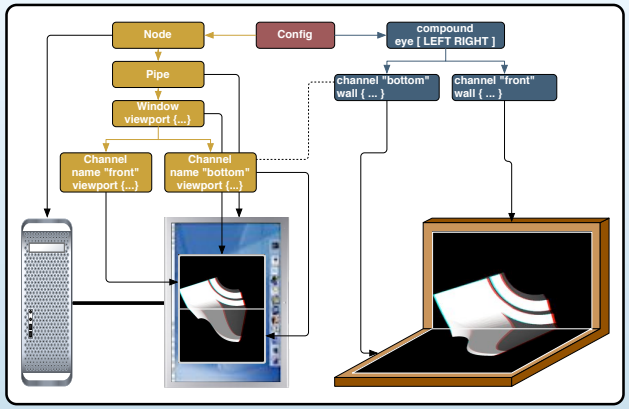
Direct-send parallel compositing

Scalability

Equalizer implements a wide range of algorithms to parallelize the rendering of large data sets. Multiple graphic cards, processors and computers can be combined to render a single view. Equalizer distributes the rendering task across the available resources (decomposition) and assembles the results on the final view (recomposition).

For the task decomposition, Equalizer currently supports sort-first (2D), sort-last (DB) and stereo (Eye) compounds. Time-multiplex (DPlex) is planned.

Equalizer supports virtually any parallel compositing algorithm, for example binary swap or direct send for sort-last, and tile gathering for sort-first rendering.



Example configuration for a TAN Holobench™

www.equalizergraphics.com
info@equalizergraphics.com
+41 76 33 77 247

Contributors:

VMML
visualization and multimedia lab
university of zürich

Support and Development Services:

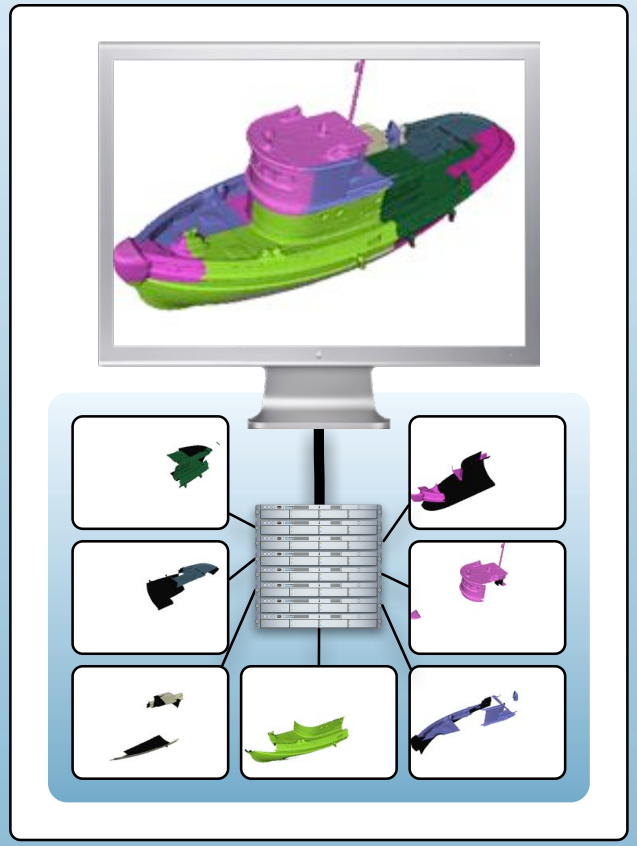
Tungsten Graphics

www.tungstengraphics.com
sales@tungstengraphics.com

Wed Apr 18 2007

Equalizer

Scalable Rendering



Equalizer is an open source programming interface and resource management system for *parallel, scalable* OpenGL® applications. An Equalizer application can run unmodified on any visualization system, from a singlepipe workstation to large scale graphics clusters and multi-GPU workstations. The foundation of Equalizer is a *minimally invasive* programming interface which addresses the problems common to any multipipe application.

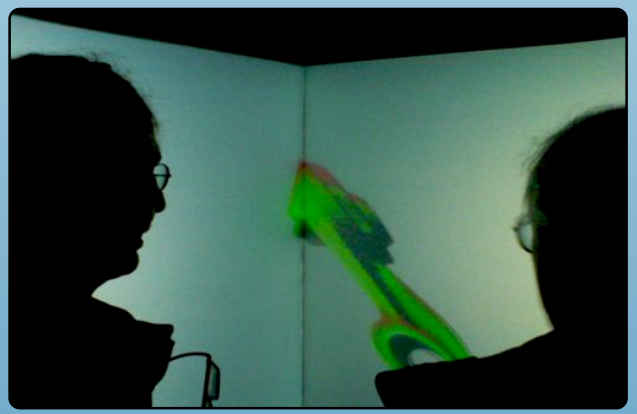
A display wall running an Equalizer-based terrain rendering application

Parallel Programming Interface

Equalizer uses a *minimally invasive* programming interface. Most of the application is unmodified, only the rendering is separated and plugged into the Equalizer API for parallelization. Common graphic entities are abstracted by C++ classes, for example:

- **Node** - a single computer in the cluster
- **Pipe** - a graphics card and rendering thread
- **Window** - an OpenGL drawable
- **Channel** - a viewport within a window

The developer amends these entities by implementing application-specific *task methods*. Equalizer facilitates application porting by providing a default implementation for each task, which implements the typical use case.



A virtual reality installation using passive stereo and head tracking

Major Features

Runtime Configurability: An Equalizer application can run on any configuration, from laptops to large scale visualization clusters, without recompilation. The runtime configuration is externalized from the application to a systemwide resource server.

Runtime Scalability: An Equalizer application can use multiple CPU's, GPU's and computers to scale the rendering performance of a single view.

Distributed Execution: Equalizer applications can be written to support cluster-based execution. The task of distributing the application data is facilitated by support for versioned, distributed objects.

Support for Immersive Environments: Equalizer supports both active and passive stereo rendering, as well as head tracking, which is required for immersive Virtual Reality installations.

Use Cases

The Equalizer framework abstracts the runtime configuration from the application code. This allows an application to be deployed in many different ways without any modification, for example:

Display Walls are one of the common uses for visualization clusters today. Typically one instance of the application's rendering code is executed locally on a node for each display.

Virtual Reality Installations use passive or active stereo rendering with head tracking, both supported by Equalizer.

Scalable Rendering parallelizes the rendering of a single view across multiple graphic cards, processors and potentially computers to render more data faster.

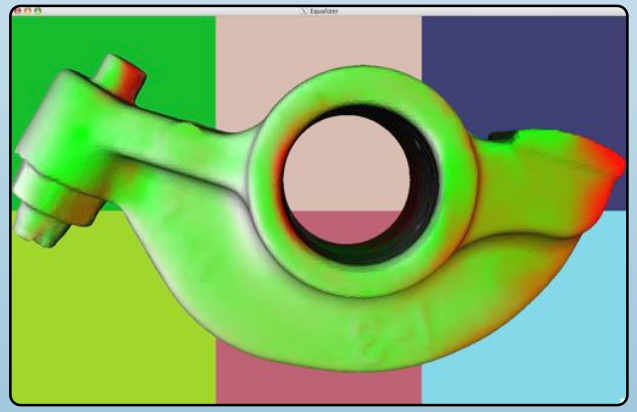
Multi-GPU Workstations are an affordable way to scale the rendering performance and display size. Equalizer applications can exploit multiple processors and graphic cards by running multiple rendering threads.

Major Benefits

Parallel Rendering Know-How: Equalizer contains 10+ years of experience in parallel and scalable rendering, easily combined with your application.

Fast Path for Scalable OpenGL Applications: Equalizer provides the natural parallel execution model to exploit the parallelism of multi-core, multi-GPU workstations and graphic clusters.

Feature-Rich Framework: Equalizer contains state-of-the-art scalable rendering algorithms, and it's open development model ensures constant improvement. The configurability allows to deploy Equalizer applications in many, often unforeseen environments.



Scalable rendering using six graphic cards to render a single view.

Compatibility

OS Support: Linux, Windows XP and Mac OS X
Interconnects: Ethernet, InfiniBand
Portable: 32 and 64 bit, little- and big-endian tested

Minimally invasive: easy porting
Clusters and **Multi-GPU** workstations

LGPL license: commercial and open source use
Open standard for parallel rendering