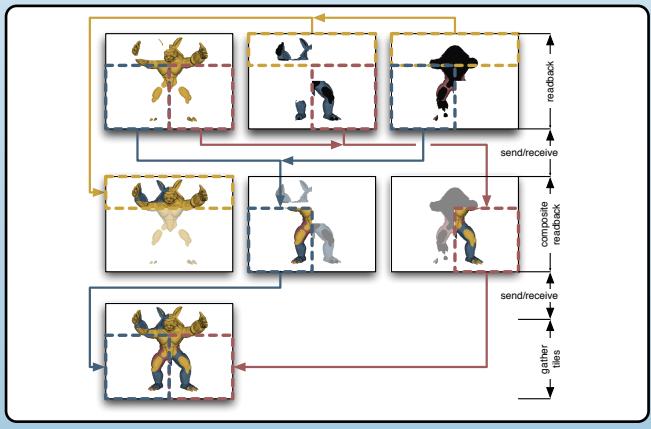


Sort-last (DB), sort-first(2D) and multi-level compounds

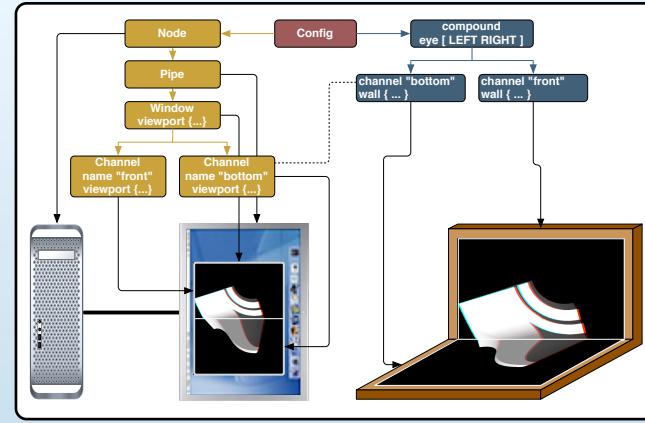


## Scalability

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.

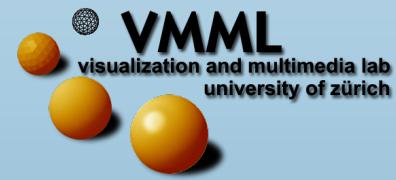
Equalizer supports virtually any parallel compositing algorithm, for example binary swap or direct send for sort-last, and tile gathering for sort-first rendering.



Example configuration for a TAN Holobench™

[www.equalizergraphics.com](http://www.equalizergraphics.com)  
[info@equalizergraphics.com](mailto:info@equalizergraphics.com)  
+41 76 33 77 247

## Contributors:



## Support and Development Services:



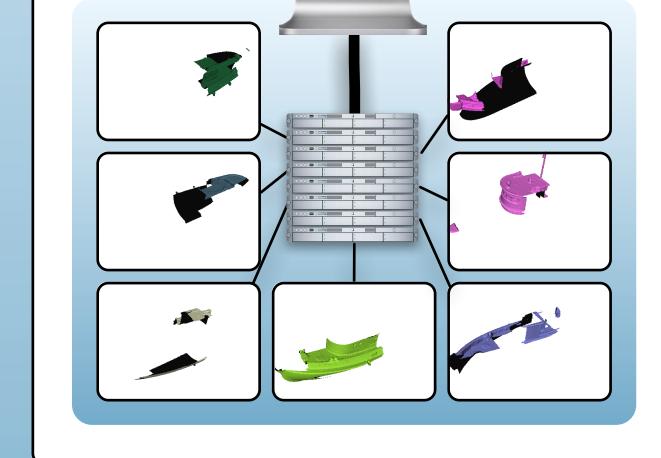
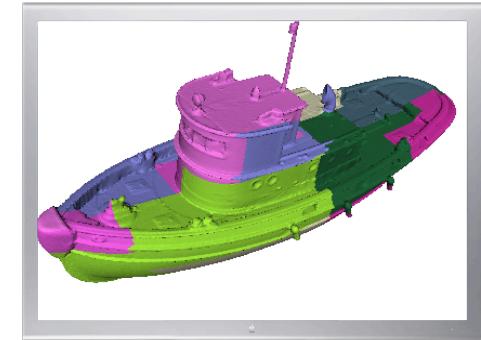
[www.eyescale.ch](http://www.eyescale.ch)  
[info@eyescale.ch](mailto:info@eyescale.ch)

Wed Jun 27 2007

© 2006-2007 Stefan Eilemann. All information provided is subject to change without notice.

# 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.

**Open standard** for parallel rendering

GPL license: commercial and open source use

**Clusters and Multi-GPU** workstations

Minimally invasive: easy porting

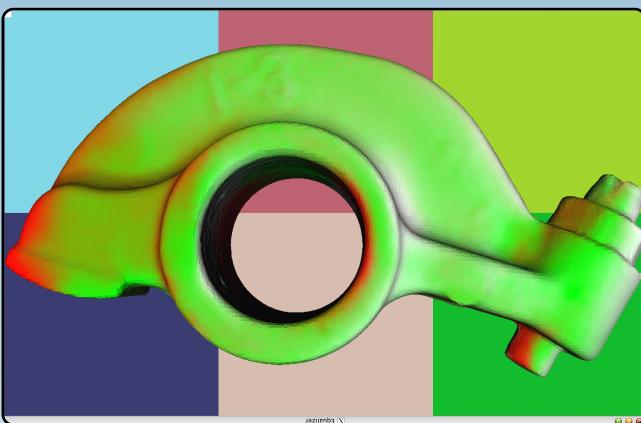
Portable: 32 and 64 bit, little- and big-endian

Interconnects: Ethernet, InfiniBand

OS Support: Linux, Windows XP and Mac OS X

## Compatibility

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



Applications in many, often unforseen environments.

The configurability allows to deploy Equalizer

development model ensures constant improvement.

of-the-art scalable algorithms, and its open

Equalizer contains state-

**Feature-Rich Framework:** Equalizer

renders in parallel multiple GPUs and graphic clusters.

Fast Path for Scalable OpenGL Applications:

Equalizer provides the natural parallel execution

rendering, easily combined with your application.

10+ years of experience in parallel and scalable

rendering, Equalizer application

Parallel Rendering Know-How: Equalizer

## Major Benefits

**Open standard** for parallel rendering

GPL license: commercial and open source use

Minimally invasive: easy porting

Portable: 32 and 64 bit, little- and big-endian

Interconnects: Ethernet, InfiniBand

OS Support: Linux, Windows XP and Mac OS X

## Major Features

**Scalable Rendering** parallelizes the rendering of a

single view across multiple graphics cards, both supported

by Equalizer.

**Virtual Reality Installations** use passive or active

stereo rendering with head tracking, both supported

by Equalizer.

**Scalable Renderings** parallelizes the rendering of a

single view across multiple graphics cards, both supported

by Equalizer.

**Virtual Reality Installations** use passive or active

stereo rendering with head tracking, both supported

by Equalizer.

**Scalable Renderings** parallelizes the rendering of a

single view across multiple graphics cards, both supported

by Equalizer.

**Scalable Renderings** parallelizes the rendering of a

single view across multiple graphics cards, both supported

by Equalizer.

**Scalable Renderings** parallelizes the rendering of a

single view across multiple graphics cards, both supported

by Equalizer.

## Use Cases

**Virtual Reality Installations:** Equalizer

supports both active and passive stereo rendering, as

well as head tracking, which is required for immersive

virtual reality installations.

**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.

**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.

**Runtime Scalability:** An Equalizer application can be

use multiple CPUs, GPUs and computers to scale

the rendering performance of a single view.

**Runtime Configuration:** An Equalizer application can

run time configuration is externalized from the

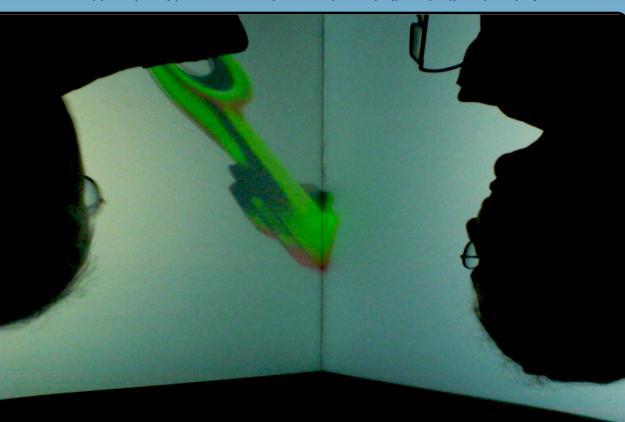
application to a systemwide resource server.

**Runtime Configuration:** An Equalizer application can

run on any configuration, without recompilation. The

scale visualization clusters, without recompilation. The

can run on any configuration, from laptops to large



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.

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

Entities are abstracted by C++ classes, for example:

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:

Equalizer is implemented into the application is unmodified, only the rendering is separated and plugged into the Equalizer API for parallelization. Only the rendering is modified, the rest of the application is unmodified.

## Parallel Programming Interface

A display wall running an Equalizer-based terrain rendering application

