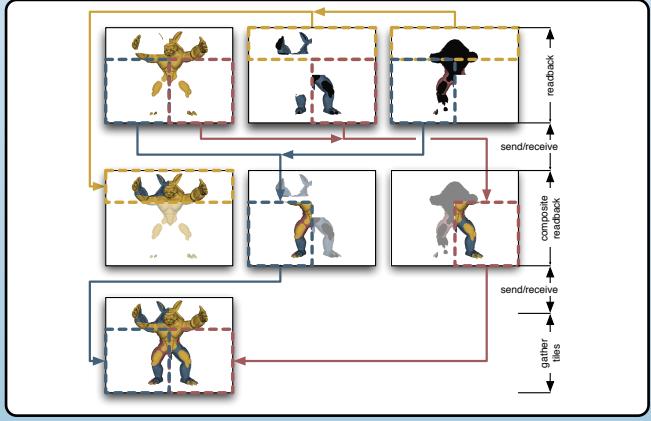


Sort-last (DB), sort-first(2D) and multi-level compounds

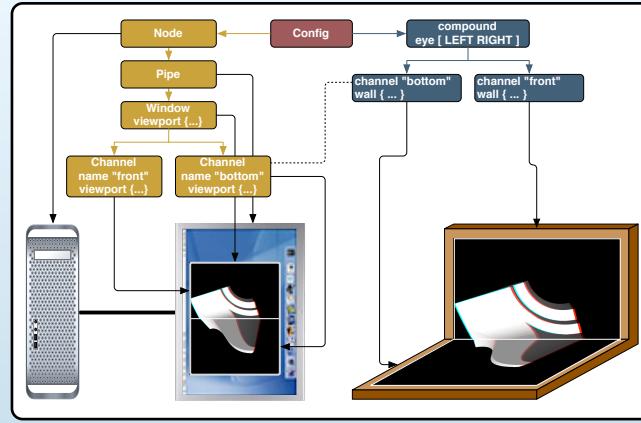


## Scalability

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.

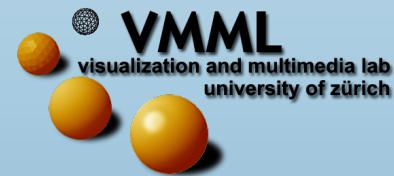
Equalizer supports virtually any parallel compositing algorithm, for example binary swap or direct send for sort-last, and tile gathering for sort-first rendering.



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:



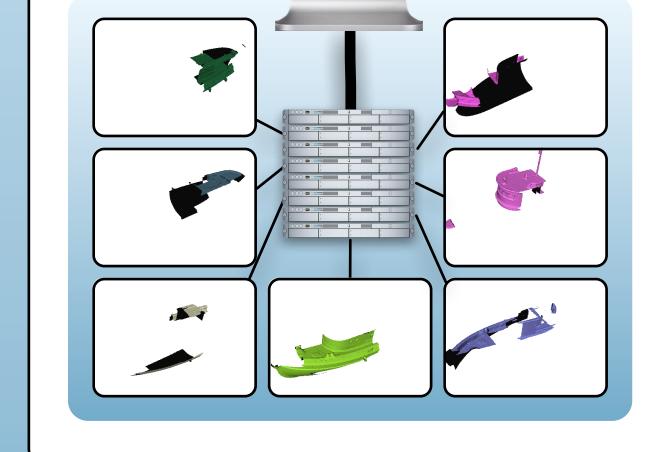
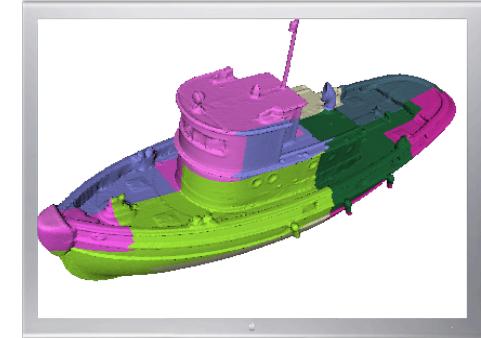
[www.eyescale.ch](http://www.eyescale.ch)  
[info@eyescale.ch](mailto:info@eyescale.ch)

Tue Jun 26 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

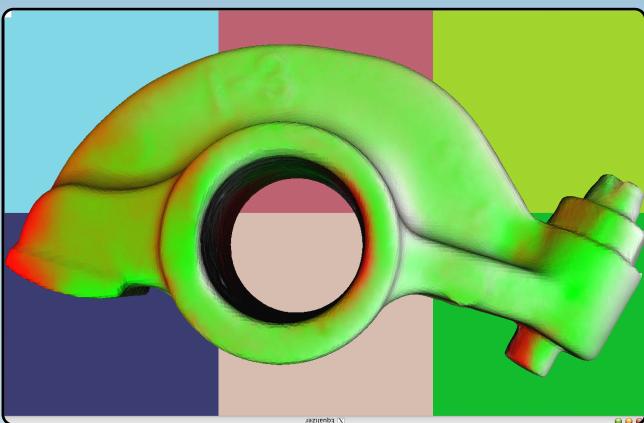
Portable: 32 and 64 bit, little- and big-endian

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 framework, and its open

algorithmic abstractions the runtime

Fast Path for Scalable OpenGL Applications:

GPU workstations and graphic clusters.

model to exploit the parallel execution

Equalizer provides the natural parallel

rendering, easily combined with your application.

10+ years of experience in parallel and scalable

parallel rendering Know-How: Equalizer contains

## Major Benefits

and graphic cards by running multiple processors

Equilizer applications can exploit multiple processors

scale the rendering performance and display size.

Multi-GPU Workstations are an affordable way to

and potentially computers to render more data faster.

Scalable Rendering parallelizes the rendering of a

single view across multiple graphic cards, processors

and potentially computers to render more data faster.

Virtual Reality installations use passive or active

stereo rendering with head tracking, both supported

by Equalizer.

Virtual Reality installations use passive or active

node for each display.

applications rendering code is executed locally on a

lization clusters today. Typically one instance of the

Display Walls are one of the common uses for visualiza-

without any modification, for example:

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 visualiza-

without any modification, for example:

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

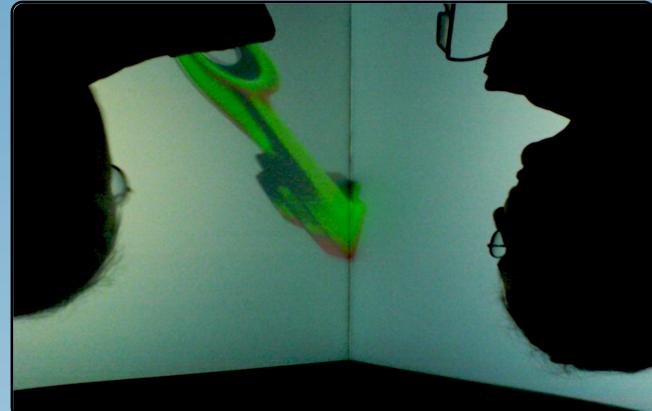
Virtual Reality installations use passive or active

node for each display.

Virtual Reality installations use passive or active

node for each display.

## Major Features



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.

- **Channel** - a viewport within a window
- **Window** - a graphics card and rendering thread
- **Pipe** - a single computer in the cluster
- **Node** - a single computer in the cluster
- **Entity** - an entity abstracted by C++ classes, for example:

## Parallel Programming Interface

A display wall running an Equalizer-based terrain rendering application

