PROSOPON OY

R o v a n i e m i - D u b l i n

THE CAVE PROGRAMMING WORKSHOP
2003/8/28

T a m p e r e

Established 1998

University of Lapland / faculty of art & design / spinoff

concepts – content – 3D visualization

History

1994 Sense8 World toolkit

1996

Multigen, Vega, Oxygen –SGO Onyx2&Infinity Reality & Panoram Technologies

1997 EON

2001 PC Cluster Visualization – WildCad 5110G

2003 nvidia FX2000 (PNY)

Clustered visualization ~ REAL-TIME 3-D GRID
~ VISUALIZATION GRID ?
(**http://www.gridtoday.com**)

some examples of PC visualization with
polarization (cyvix converter, 2 lcd's;
AMD k2 1800 Mhz, FX2000).

2 years co-operation with YIT (the
biggest construction company in
Finland; turnover 3 bill. euros) – covers
the whole life cycle of construction
business.

---

3 random examples of PC cluster
visualization

http://www.modviz.com/products/renderizer.htm

Renderizer™ Visualization Cluster Software enables
scalable real-time 3-D rendering on PC-clusters

manages distributed real-time rendering on PC clusters

with scenegraph software, Renderizer software-managed
clusters rapidly generate real-time, high resolution,
stereoscopic

Linux and Windows; APIs OpenGL Performer and Open
Inventor

**Add as many rendering processors (PCs) as needed without
increasing synchronization overhead or impacting individual
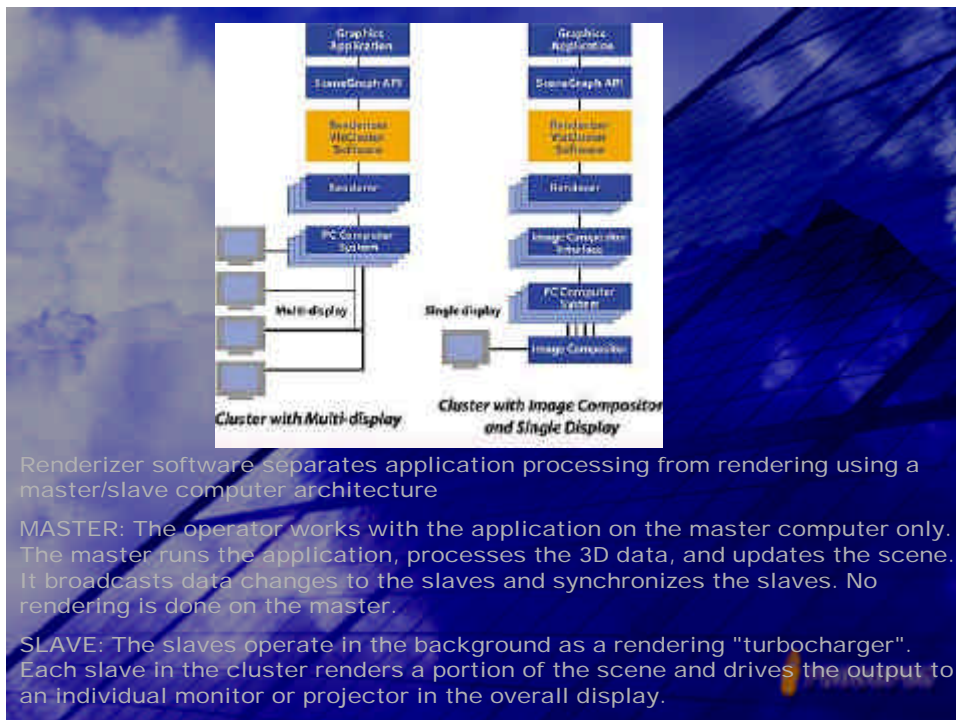channel performance (really ?)**

Renderizer software manages the visualization cluster as a single system and is configured from a single configuration file.
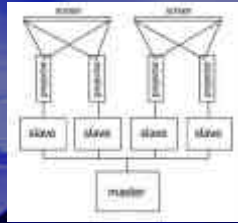
To add a channel connect a PC into the network, add an imaging device (monitor, projector) entry to the configuration file.

Renderizer Visualization Cluster Software decomposes digital models and distributes their elements among the clustered PCs.

PCs render their assigned objects and process changes in synchronized data streams to screens of a few linked desktop computers.

 With the addition of image composition technology, the multiple outputs from the PCs can be recombined into a seamless, single image.

Renderizer software separates application processing from rendering using a master/slave computer architecture

MASTER: The operator works with the application on the master computer only. The master runs the application, processes the 3D data, and updates the scene. It broadcasts data changes to the slaves and synchronizes the slaves. No rendering is done on the master.

SLAVE: The slaves operate in the background as a rendering "turbocharger". Each slave in the cluster renders a portion of the scene and drives the output to an individual monitor or projector in the overall display.

The images that the PC clusters calculate are "drawn" by scengraph software.
Currently Renderizer supports OpenGL Performer and Open Inventor from TGS

EON Studio is a GUI based tool for developing real-time 3D application
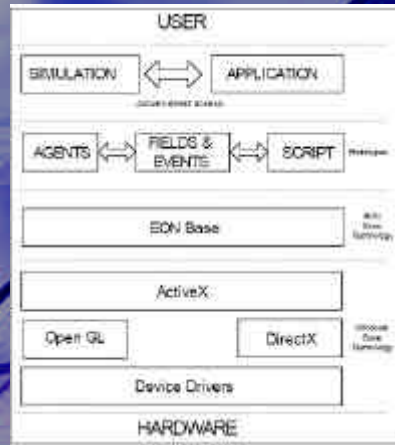
Adding behaviors through EON's graphical programming interface, scripting or compiled C++ code.

Applications can also be integrated in other tools that supports ActiveX components
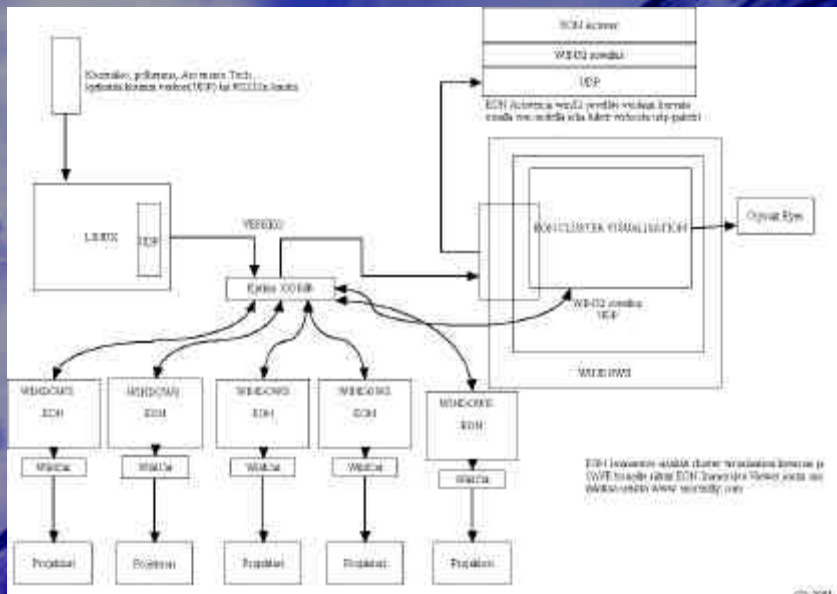
(Macromedia, Shockwave, etc.)

EON System Architecture

EON Cluster Visualization system

- synchronized multi-channel simulations, using a cluster of PCs linked together in a local area network
- Each computer is responsible for one display channel.
- uses a master/slave model, where one computer in the cluster is elected to be the master machine, responsible for maintaining scene synchronization, administration issues and input from the user.
- The remaining computers in the cluster are slaves.
- The synchronization of image updates on all displays is done on a per frame basis
- the computers are essentially working parallel during the preparation of a frame, working synchronously only during the update of the display buffer, signaled by a heartbeat from the master computer.
- the system's overall performance is determined by machine with the lowest level of performance.

- GUI makes it possible for the user to specify the rendering quality and viewport orientation of each channel centrally. These parameters are set in the property windows of the CVS nodes in the EON simulation tree. All configuration settings are saved along with the other EON nodes, forming a ready-to-use multi-channel simulation file.

- The simulation file must be stored on a network-shared drive, so that the slave machines can access it. (File and Printer sharing for Microsoft Network service)

- all slave machines load the simulation file from the master computer, (no need to distribute the simulation file to each member machine).

Cluster Visualization nodes (simulation is built from a number of nodes)

RemoteSC node
Each sub-channel is represented by a RemoteSC node.
The RemoteSc node determines the quality and resolution of the channel display.
The node intializes and updates the sub-channel (on a slave machine)
The sub-channel nodes can only be inserted under the manager node, RemoteSCM
Each RemoteSC node represents a slave machine.
(assigning the same slave identifier to two machines in the Distributed Multi-Channel Settings Properties window, the configuration will have effect on both machines.

RemoteSCM node
Responsible for the enumeration and management of the subchannels/slave machines.

EON SDK
create custom plug-ins to EON Studio:
·**Nodes – Runtime components**
EON nodes are the most commonly used objects by EON Application developers. In design time nodes are used for constructing a 3D environment, in runtime each node has it's functionality in order to add behavior to the 3D environment.

·**Modules – Design time components**
Modules are used in design time and adds functionality to EON Studio which can be accessed through general GUI components like
·Menus
·Toolbar buttons
·Windows
All windows in EON Studio are modules (Simulation tree, Routes, Component Window etc).

MS Visual C++ is used for creating the components. The EON SDK adds

·Two wizards to MS Developer Studio that create skeleton code for nodes and modules

·Some base classes for use with nodes and modules

·A number of libraries with functions for use with the rest of the EON kernel

·Knowledge of

1.MS Developer Studio

2.C++

3.COM, ATL

4.MFC (MDI, SDI)

Integration of a Commodity Cluster into an Existing 4-Wall Display System
Douglas B. Maxwell, Aaron Bryden, Greg S. Schmidt, Ian Roth, J.Edward Swan II
Virtual Reality Laboratory, Naval Research Laboratory

2002-08-18-VideoCLusterPaper.pdf

•Low cos per performance of PC commodity clusters
•Cluster Design Strategies
•VR Juggler
•The quality of the displays and stereo viewing was comparable to the same software running on three walls using an SGI Onyx 2 with IR2 graphics
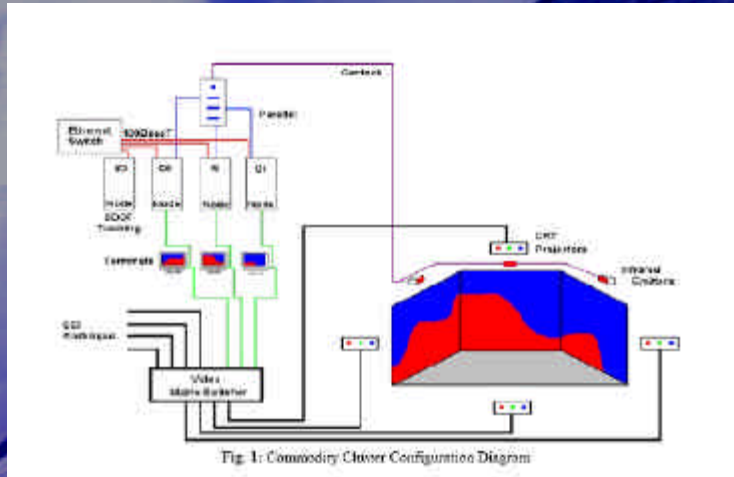•Cluster was able to visualize the data with better performance

Fig. 1: Commodity Cluster Configuration Diagram