Multi-Graphics

Scalable Graphics using
Commodity Graphics Systems

VIEWS PI Meeting
May 17, 2000

Pat Hanrahan
Stanford University
http://www.graphics.stanford.edu/

Multi-Graphics Project

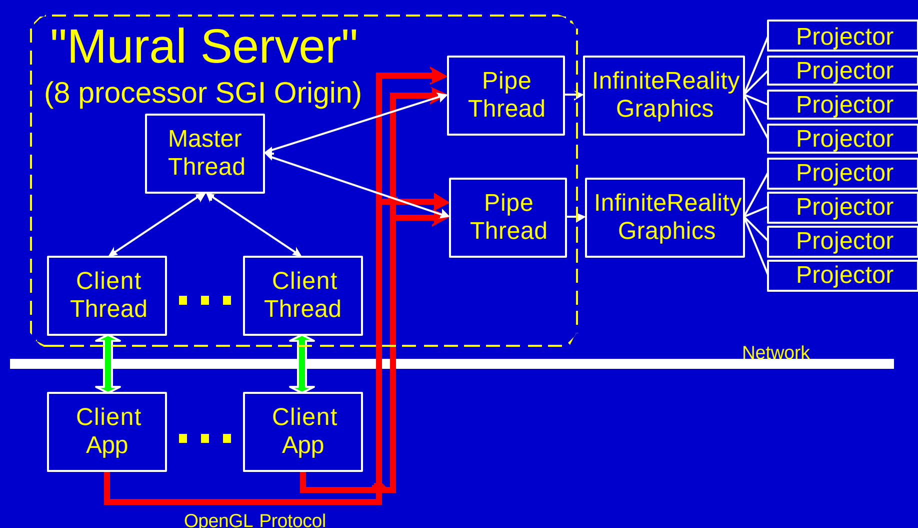
Scalable and Distributed Graphics and Visualization using Commodity Graphics Components

- Research Team
  - Greg Humphreys , 3rd yr grad, scalable cluster rendering
  - Ian Buck, 1st yr grad, remote rendering
  - Brad Johanson, 4th yr grad, interactive room
  - Matthew Everett, Sr ugrad, fast geometry compression
  - Susan Shepard, staff, Interactive Room
  - Maureen Stone, consultant, Interactive Mural
- Stanford-TriLabs Visualization Cluster
- Research funded by DVC-ARP Program
- Industrial collaboration: Intel, NVIDIA, (SGI, SUN, SONY, 3DFX)

Overview

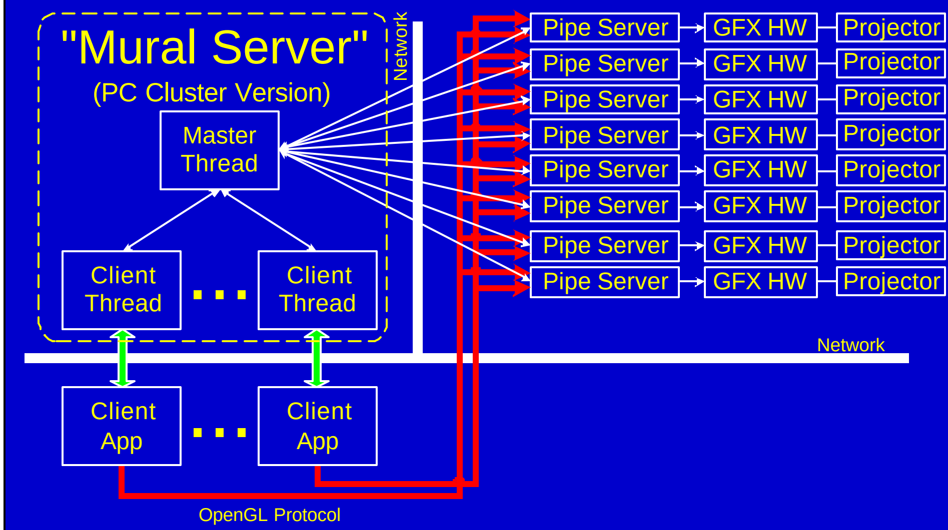
- Visualization cluster
- Lightning-2 image composition network
- Interactive room (mural and table)
- Parallel WireGL

SGI RealityMonster



G. Humphreys , P. Hanrahan, A distributed graphics system for large tiled displays, Vis99

PC “RealityMonster”



G. Humphreys, I. Buck, P. Hanrahan, Distributed rendering for large tiled displays, Submitted to Supercomputing 2000

IR vs. GeForce256 Benchmarks

Millions of triangles per second*

	InfiniteReality	GeForce DDR
Unlit	4.858	6.908
Lit	5.163	7.270
Unlit textured	5.255	7.270
Lit textured	3.651	5.715

*Performance measured by Stanford OpenGL benchmarking tool

IR Vs. GeForce256 Benchmarks

Millions of pixels per second*

	InfiniteReality	GeForce DDR
Pixels	425.638	320.418
Bilerp textured	297.753	245.612
Trilerp textured	247.945	154.127
Trilerp textured blended	247.977	106.165

*Performance measured by Stanford OpenGL benchmarking tool

IR Vs. GeForce256 Benchmarks

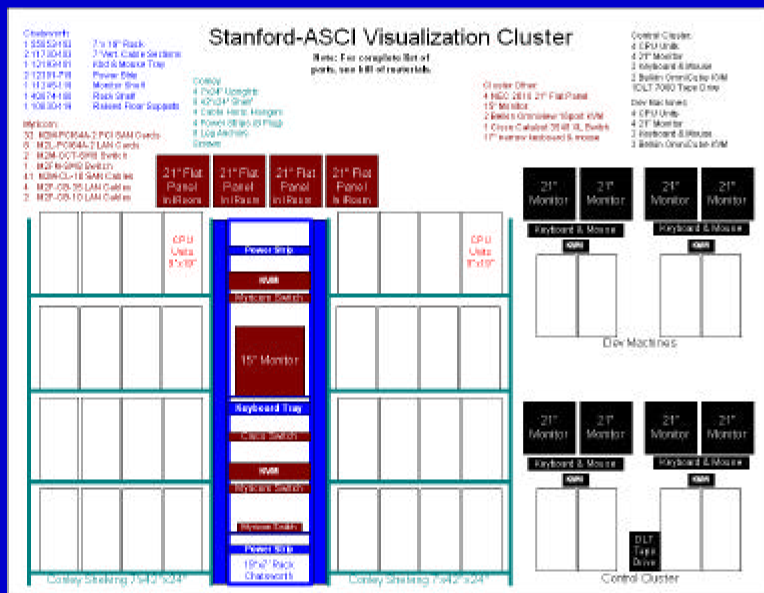
Cost

	InfiniteReality	GeForce DDR
Price	$106,000	$3000($250)

Next gen. PC graphics systems will be much faster

- X-Box 300 million triangles per second, 1 tera-op/s

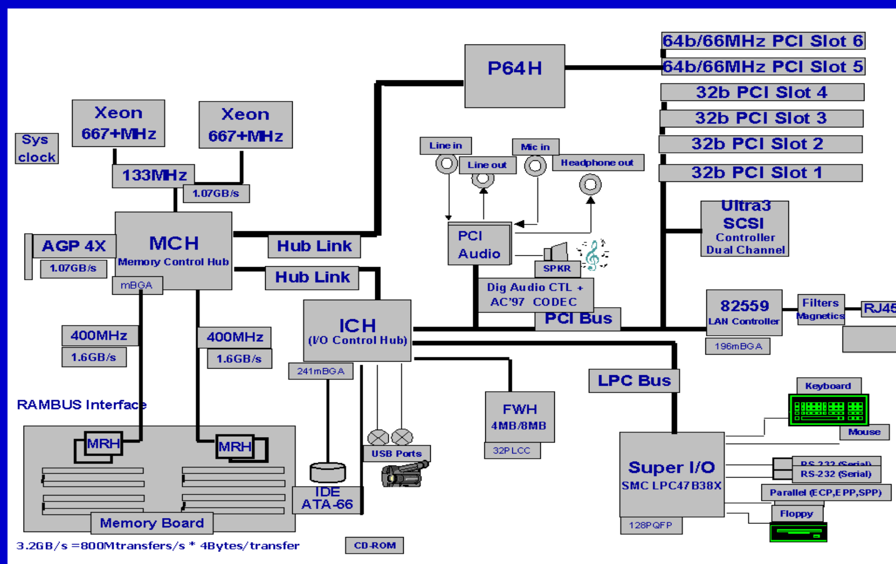
Chromium – **Cr** (Cluster Renderer)



Chromium

- Cluster: 32 graphics nodes, 4 server nodes
- Computer: Compaq SP750
  - 2 processors (800 Mhz, 133Mhz FSB)
  - i840 core logic (*biggest issue for vis-clusters*)
  - 256 MB memory
  - 18GB disk (+ 3*36GB on servers)
  - 64-bit, 66 Mhz PCI
  - AGP-4X Pro
- Graphics
  - NVIDIA GeForce2 GTS (NVIDIA Geforce256 DDR)
  - NVIDIA NV20 with DVI this summer
- Network
  - Myrinet 64-bit, 66 Mhz

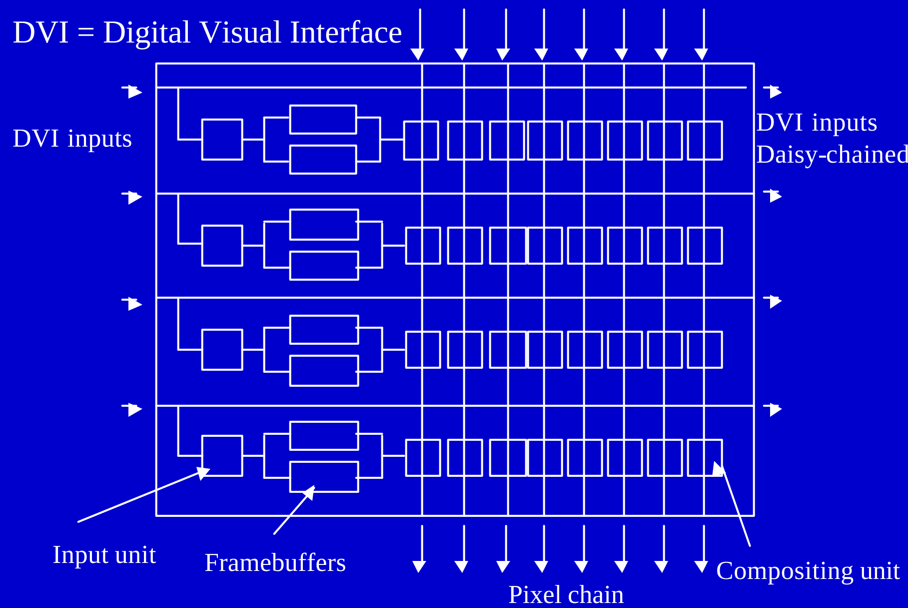
SP750: i840 implementation



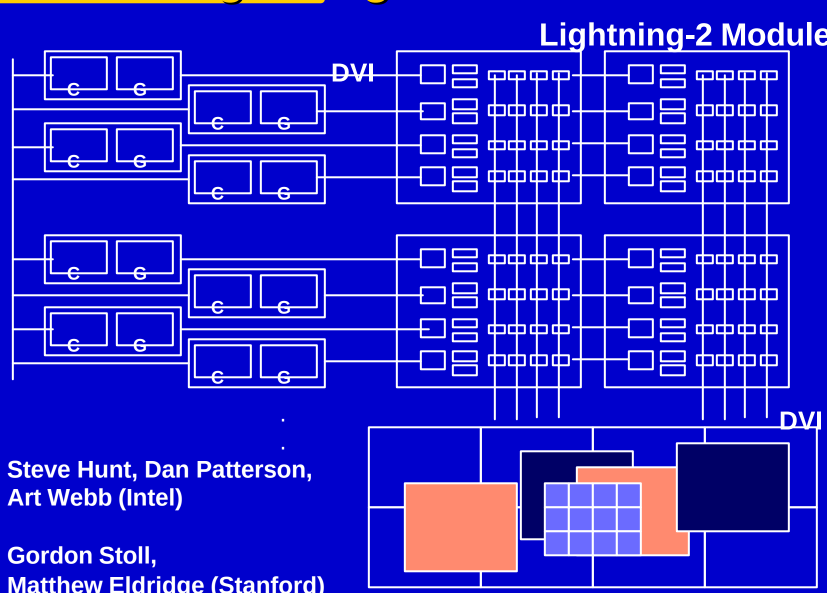
Linux Graphics

- Linux has high quality fast network drivers
  - Servernet II measured at 162 MB/sec
  - Myrinet measured at 142 MB/sec
- Windows fast network drivers disappointing
  - Servernet II only in Windows 2000
  - Myrinet crashes Windows NT in 64/66 slot
- Linux graphics drivers are *rapidly improving* (thanks to DOE!)
  - "98% shared code"
  - Just-released drivers very usable
  - Linux AGP support is the current bottleneck
  - Direct rendering architecture still contentious
  - Ultimately more efficient?
- Our software runs identically on NT, Linux, Irix

Lightning-2 Module



Scalable Lightning-2



Steve Hunt, Dan Patterson, Art Webb (Intel)

Gordon Stoll, Matthew Eldridge (Stanford)

Scanline Switching

- Placement information is embedded in the image
- Unit of mapping is a one-pixel-high strip

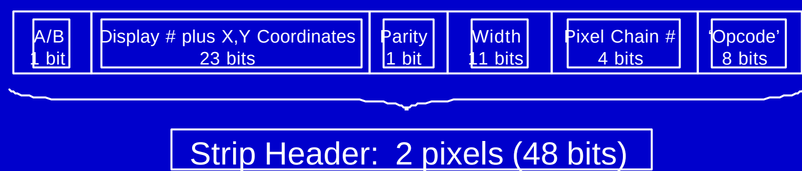


Image Assembly and Output

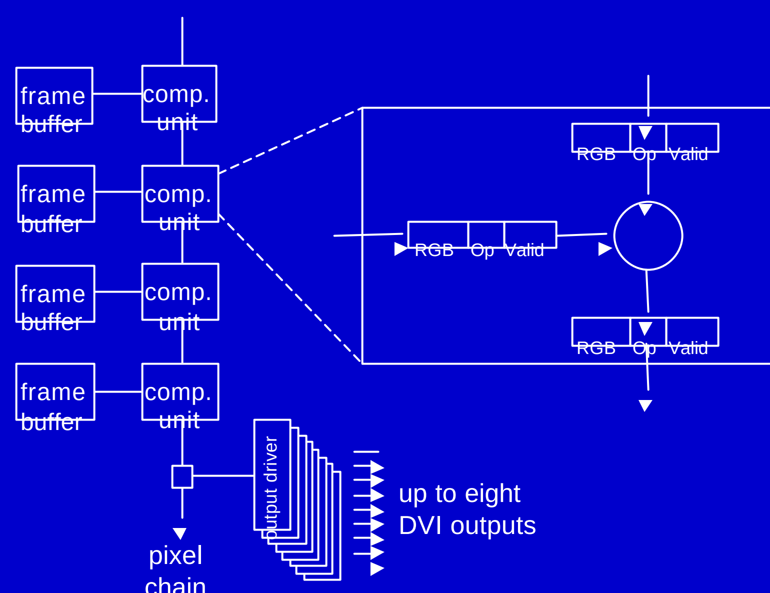


Image Composition

- Layers or windows
- Chroma-keying
  - Treat local pixels of a specific color as transparent
- Depth compositing
  - Copy depth values into the PC frame buffer
  - Scan depth on a second pixel chain
- Advanced operation
  - Program additional functions in the compositing FPGA
  - Take advantage of the 8-bit opcode
  - Combine data from multiple streams

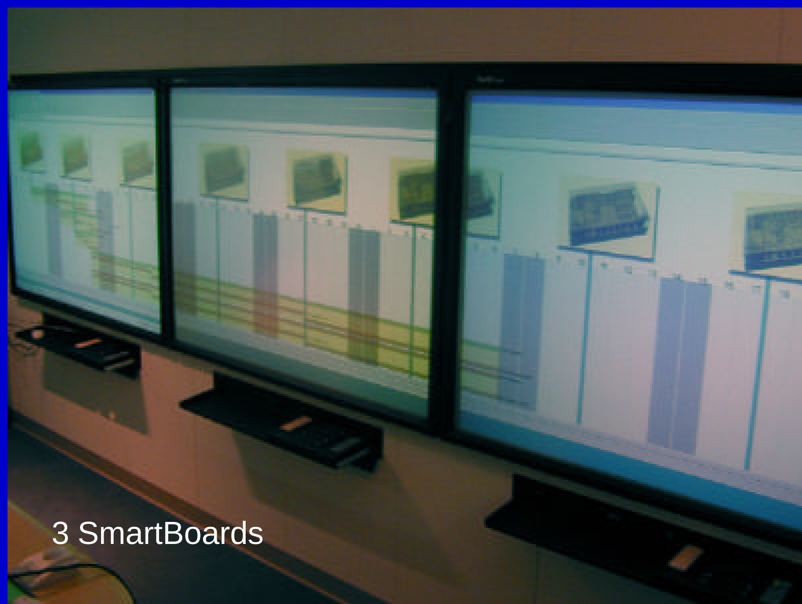
Interactive Mural

Image and virtual colonoscopy concept courtesy Sandy Napel, Stanford Radiology Department.

Projectors: 1024x768, 900 ANSI Lumens

Mural: 6’ x 2’, 4096 x 1536, ~60dpi,

Interactive Mural



Interactive Conference Room Table

Interactive Workspaces

Advanced display technology meets ubiquitous computing

Planning and Evaluating

Disney Theme Park Model

Software Goals

1. Efficient "as-possible" remote rendering
2. Scalable to support high resolution tiled displays
3. Parallel graphics API for ultimate scalability
4. Immediate mode for dynamic, time-varying datasets
   (Does not "require" data replication)
5. Support heterogeneous clusters/components/os'es
   (Eventually "network-aware")
6. Drop-in, dynamically linked OpenGL library

WireGL

- Similar in spirit to GLX OpenGL 1.1 protocol
- Novel features
  - Virtual OpenGL
  - Protocol
    - More efficient handling of **glvertex** commands
    - Optimized packer
  - State tracking
  - Parallel API

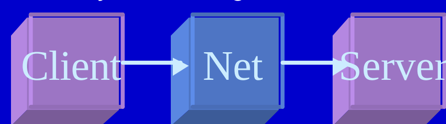
High Performance Remote Graphics

GLUT Atlantis Demo
Vertices per second (Frames per second)

Direct	**2.8M (1015 fps)**	

Remote	100mbit	1000mbit
GLX	.108M (39)	
WireGL	.22M (78)	**.98M (350 fps)**

SP750 w/ GeForce w/ Myrinet (32/32) under WindowsNT
Have measured remote rendering at >2M vertices per second
Roughly 35% efficiency – current goal: drive much higher



State Tracking

Technique

- Intercept OpenGL commands on client and track state
- Mirror OpenGL state for each connection on server

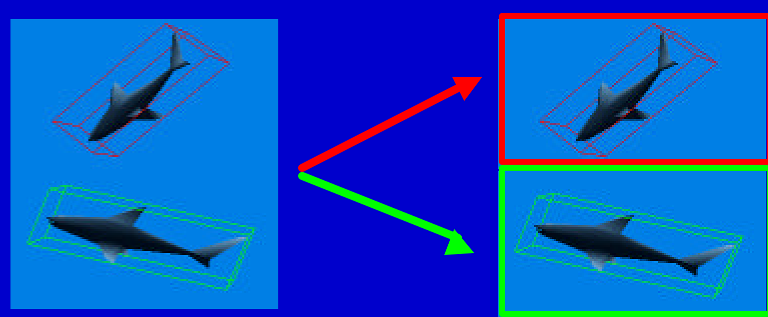
Enables

- Efficient remote rendering (culling, compression)
- Lazy state update (no state broadcast)
- Fast context switching
- Tiled displays (bucket-sort)
- CAVES and stereo rendering (create multiple viewing transforms)
- Parallel API (reorder streams)
- Mobile contexts and vis-servers

I. Buck, G. Humphreys , P. Hanrahan, Tracking graphics state for networked rendering, to appear Graphics Hardware 2000

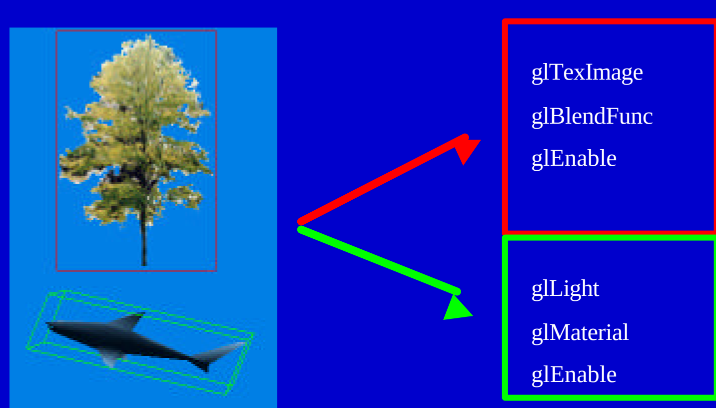
Interactive Mural Graphics

- Sort (first) geometry into tiles using bounding box
  - Incrementally update bounds with each vertex
  - Transform only primitive blocks
- Separate rendering stream for each server



Lazy State Updates

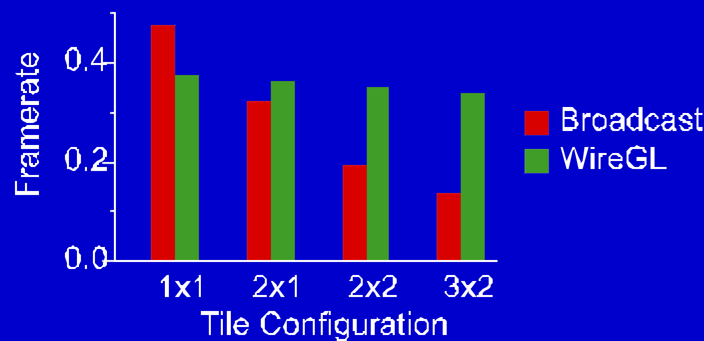
- Only send state which is required for rendering



Mural Performance

Marching Cubes Application

Prototype 8 Node PC cluster with GeForce and Myrinet
Driving an 8 projector Interactive Mural

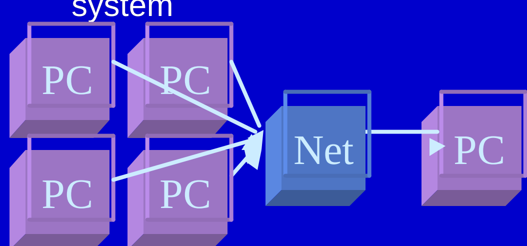


Under simulation, performance scales to 32 nodes

Parallel OpenGL API

Key to scalability

- Multiple applications issuing graphics streams
- Each stream has different graphics context
- Each stream inserts **synchronization** commands
- Graphics streams are **reordered** in the graphics system

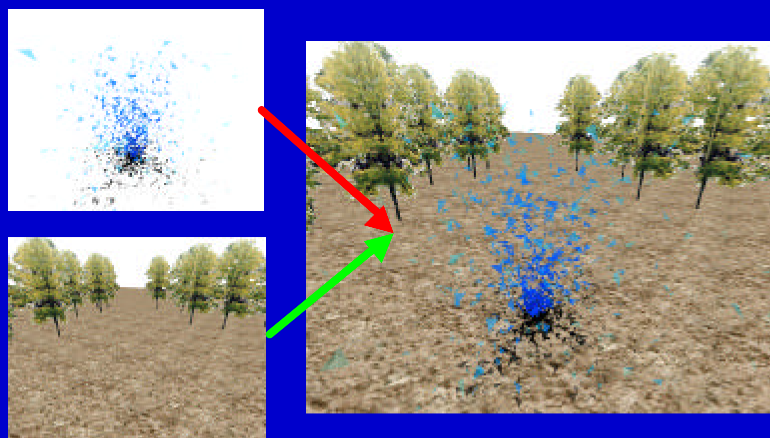


Igehy, Stoll, Hanrahan
SIGGRAPH '98

Interactive Workspaces Overview
Stanford University Computer Science Dept.

Soft Context Switch

- Compute context difference
- Generate OpenGL commands to switch contexts



Fast Context Switching

Graphics context switches per second*

Graphics System	Identical	Varying
InfiniteReality	719	697
Cobalt	2,239	2,101
GeForce DDR	11,919	5,968
WireGL	5,817,152	191,699

Identical: graphics state in old and new the same
Varying: graphics state in old and new different

Performance Goals (Summer 2000)

Graphics == Network

- Network: 6 Mtri @ 13 bytes/tri = 78 Mb/s
- Graphics: 6 Mtri/sec, 400 Mpix/sec

32 node system

- 32*5 = 160 Mtri/sec
- 32*400 = 12.8 Gpix/sec

Assumes …

- Perfect load balancing
- Lightning-2 image composition

  Should also work well with pixel readback

Status

- Accomplishments
  - Interactive room remodeled (Oct 99)
  - Conference room-style tabletop display
  - Interactive mural (3 smartboards)
  - Remote rendering functional
  - Tiled rendering functional
- Future plans
  - Release alpha version of wiregl to DOE, Brown, Princeton, …
  - Install cluster by June 1st
  - Parallel API by June 1st
  - 8 Lightning-2 modules by mid-June
  - New 12 projector mural (seamless) by end of summer
  - Port DOE app (Williams volume rendering?) by end of summer
  - SIGGRAPH Demo w/ Intel/NVIDIA