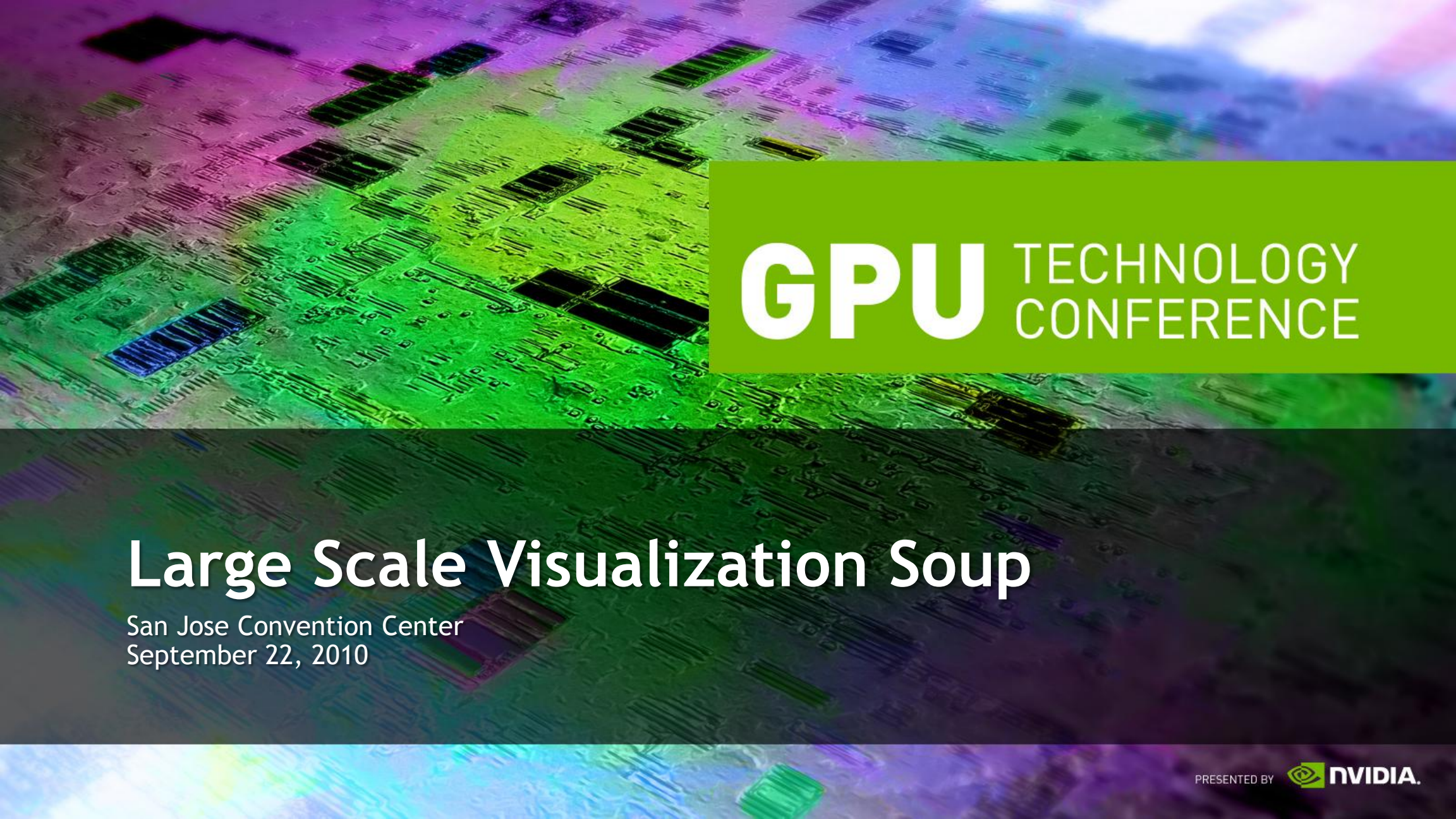


Large Scale Visualization Soup

San Jose Convention Center
September 22, 2010

PRESENTED BY NVIDIA.

Agenda

- Intro – Why Use Large Scale Visualization?
- LSV Hardware
- Adding More Realism with:
  - High Dynamic Range
  - Stereo
  - Multiple Displays
- SLI Mosaic mode
- Beyond Mosaic mode

Why use Large Scale Visualization?

- Economics
- Quality
- Detail
- Pixel real estate
- Stereo
- Immersive experience

GPUs
LSV Hardware

> 8 DVI

Quadro Graphics
Quadro G-Sync Card

4-8 DVI

2-4 DVI

1-2 DVI

Beyond 8 DVI Dual Link Requires Clustered PCs with Quadro G-Sync to synchronize displays and Multi GPU aware software.

Applications written to run on a single display just work across larger display formats.

Quadro Plex 7000
LSV Hardware

- Contains 2 Quadro 6000s
- 12 GB of total GPU memory
- 2 DVI connectors per GPU
- LED to determine primary display
- Easily rack-mountable
- Available November 2010

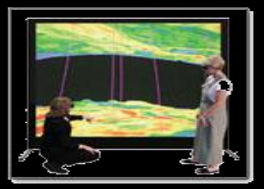
Display Technologies
LSV Hardware

- Panels

- Projectors

- Multiple Panels

- Multiple Projectors



Adding More Realism with HDR

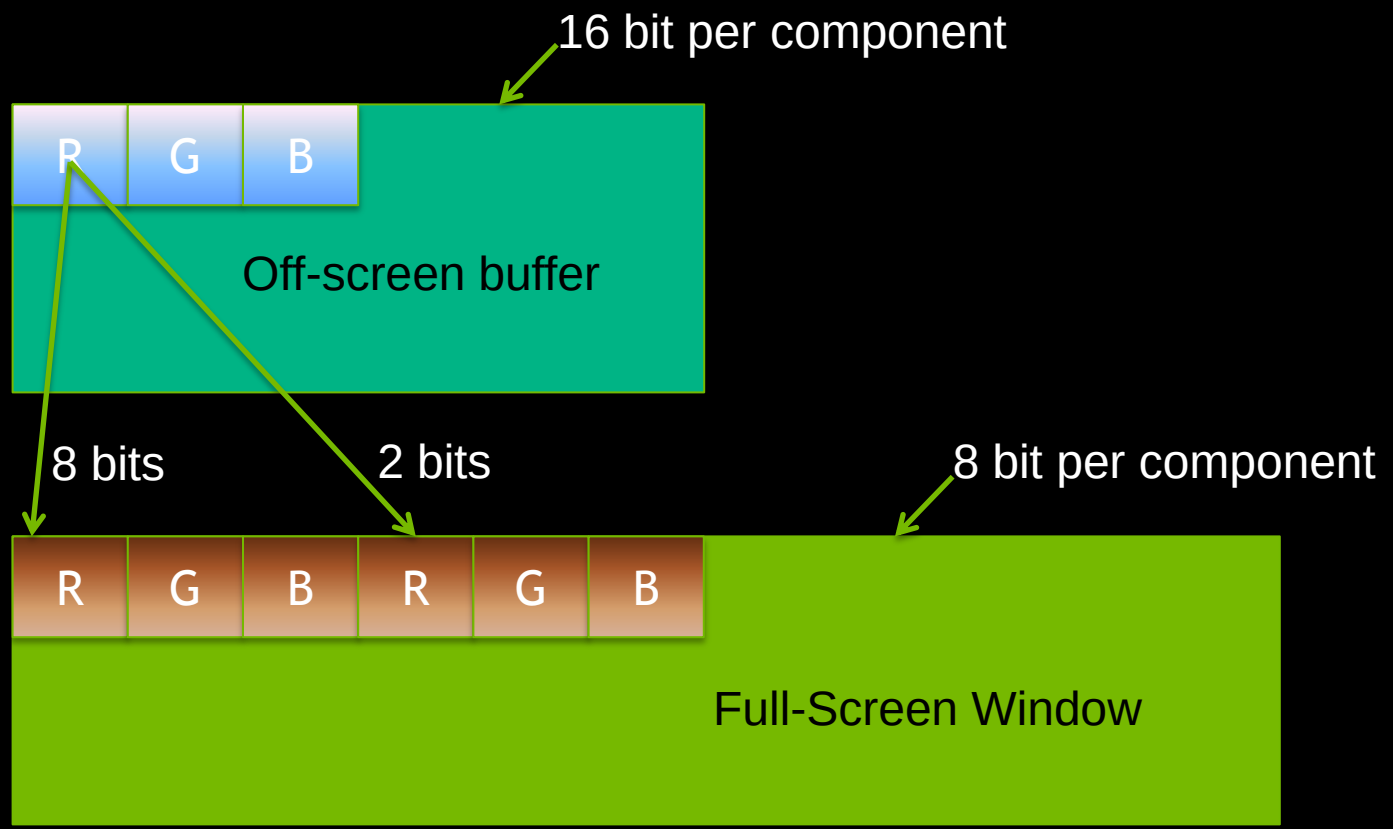
Mechanics of >8bit per component

- Possible using both DVI or Display Port
  - Display Port much easier
- Textures etc. need to be >8bit per component
  - FP16, I16
  - RGBA, LA, L

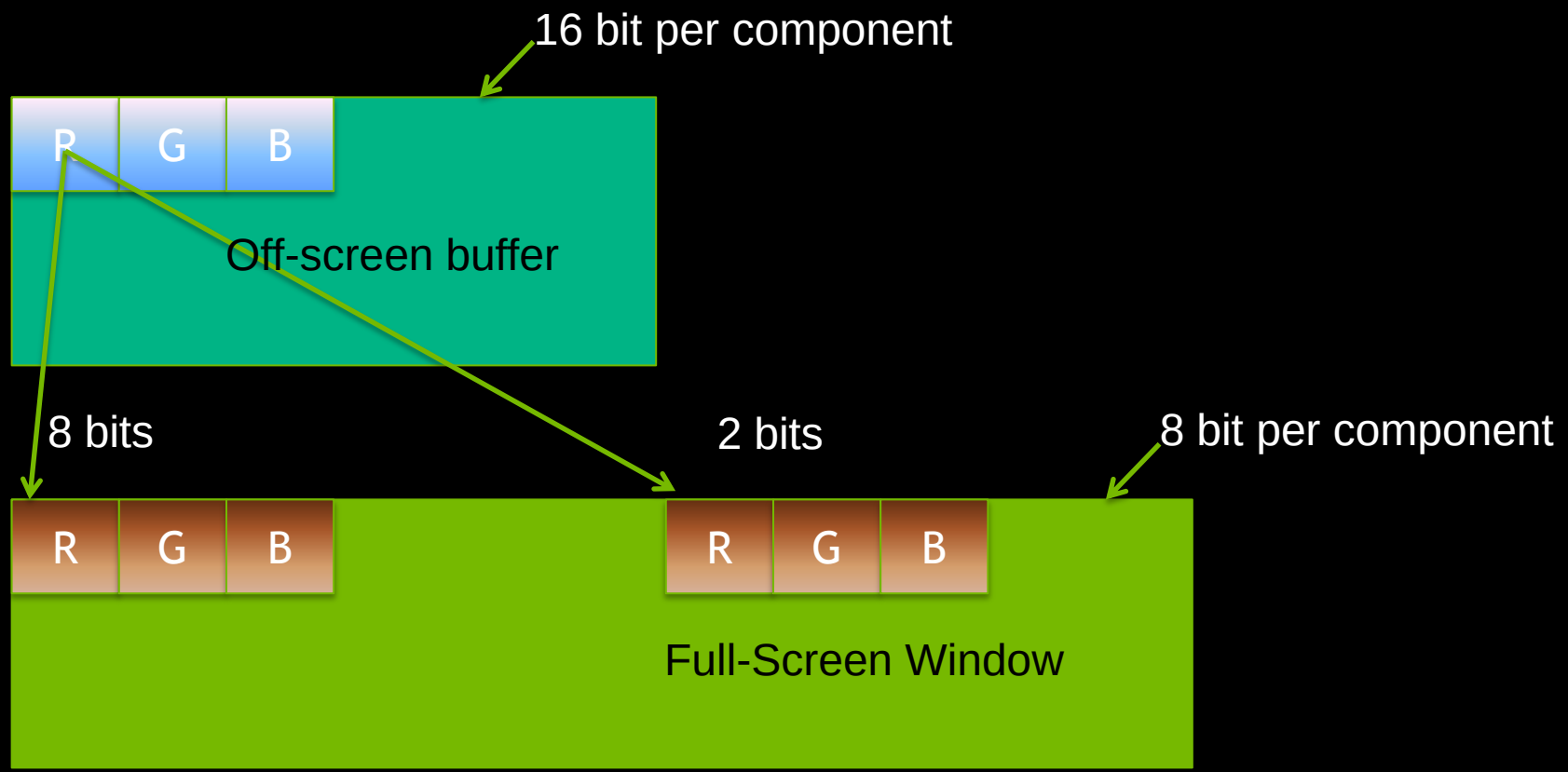
Implementing HDR over DVI

- Full screen only
  - Desktop, GUI, etc will not be correctly displayed
- Format specific to display device
- Outline:
  - Configure double-wide desktop
    - Significantly easier if exported by the EDID
  - Create full-screen window
  - Render to off-screen context
    - E.g. OpenGL FBO
  - Draw a textured quad
    - Use fragment program to pack pixels - display device specific

Implementing HDR over DVI
Using 1 Dual Link DVI



Implementing HDR over DVI
Using 2 single link DVIs



Implementing HDR over Display Port

- Requires native Display Port GPU
- Desktop will be displayed correctly (in 8bit)
- Easy!
  - Open 10bit per component Pixel Format/Visual
  - Render

Adding More Realism with Stereo

Stereo
NVIDIA 3D Vision

Software
3D Vision SW automatically converts mono games to Stereo
Direct X only

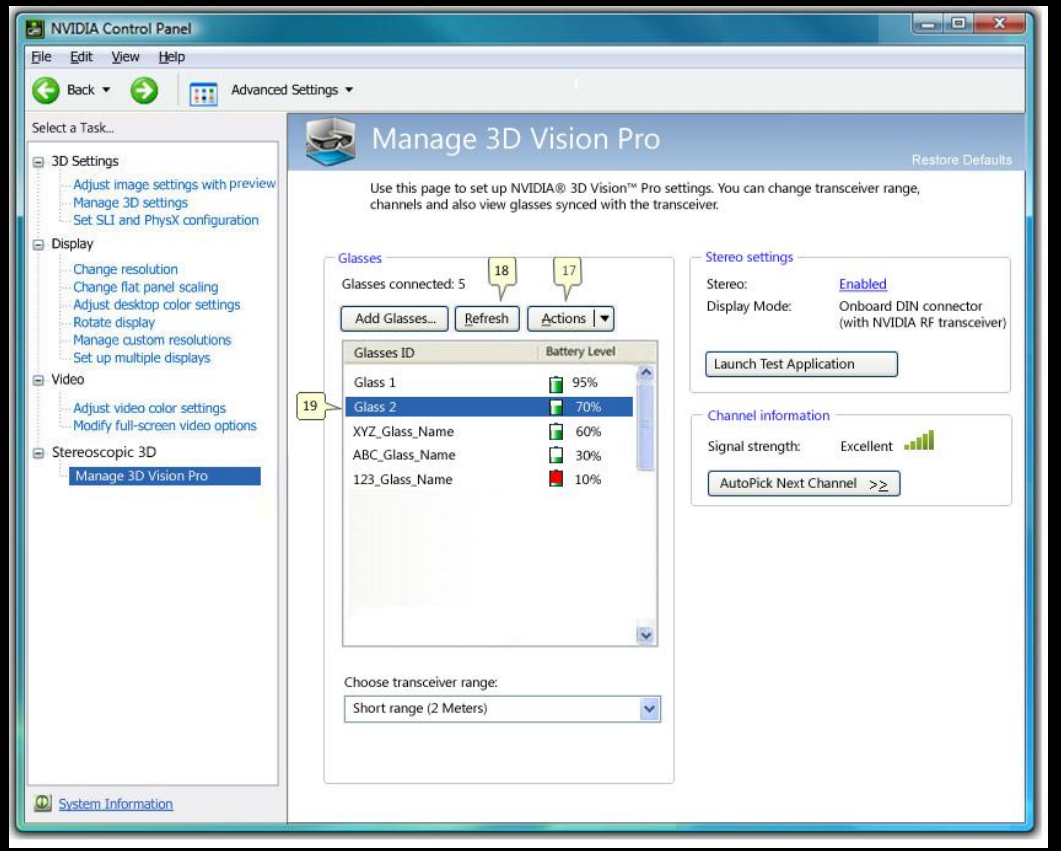
Hardware
IR communication
3D Vision certified displays
Support for single screen or 1x3 configurations

Stereo
NVIDIA 3D Vision Pro

Software
Supports Consumer 3D Vision SW or Quad Buffered Stereo
QBS: OpenGL or DirectX For DX QBS, e-mail 3DVisionPro_apps@nvidia.com for help

Hardware
RF communication
3D Vision certified displays, Passive Displays, CRTs and projectors
Up to 8 displays
Mix Stereo and Regular Displays
G-Sync support for multiple displays and systems
Direct connection to GPU mini-DIN

Stereo
NVIDIA 3D Vision Pro



Hardware - cont'd
Designed for multi-user professional installations
No line of sight requirement, no dead spots, no cross talk
RF bi-directional communication with UI
50m range
Easily deploy in office no matter what the floor plan

Adding More Realism with Multiple Displays

Multiple Display Channels

Why multiple display channels?

- More pixels!
- Resolutions becoming larger than DVI/DisplayPort bandwidths can sustain
  - Sony, JVC 4K projectors
  - Barco and Mitsubishi 4K panels

Implications of Multiple Display Channels

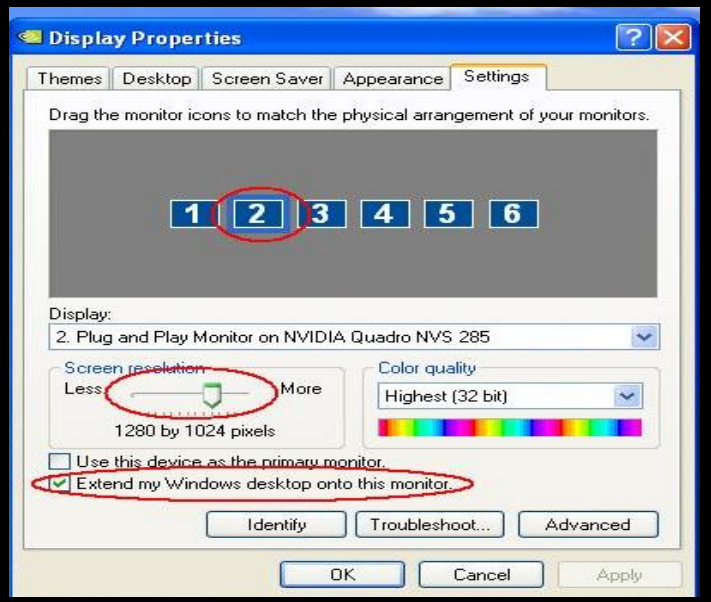
First a couple of questions:

- Which OS - Windows or Linux?
- Level of application transparency:
  - Driver does everything?
  - Application willing is multi display aware?

Multiple Displays - Windows

- Attach Multiple Monitors using Display Properties
- Extend the Desktop to each GPU
- Ensure ordering is correct for desired layout
- Adjust Resolutions and Refresh Rates
- Displays using Refresh Rates <48Hz can be problematic
- Synchronizing displays requires G-sync card

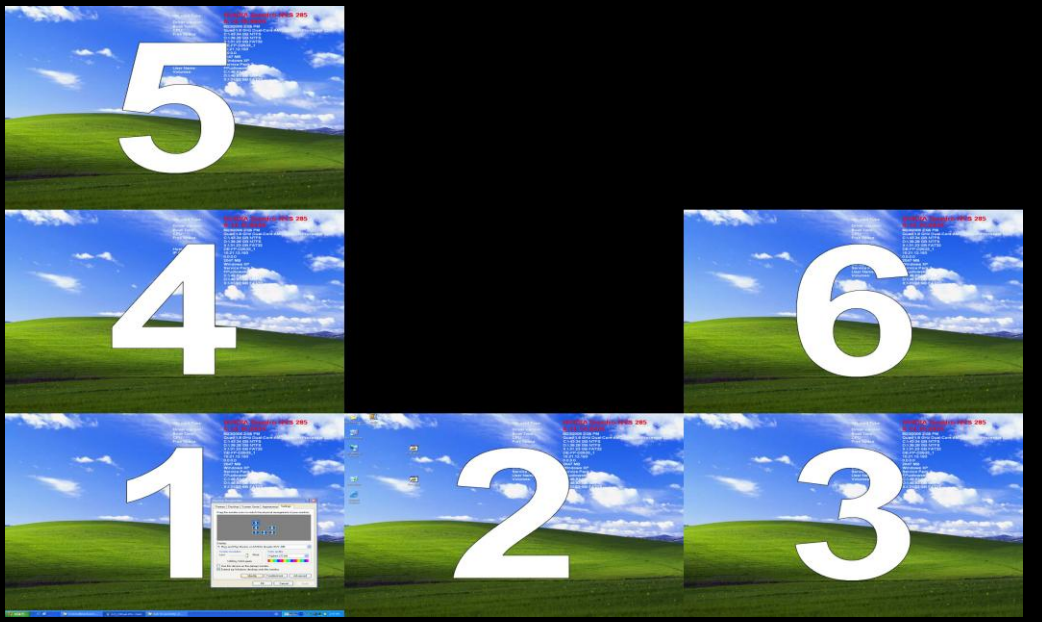
Multiple Displays - Windows



Note the task bar

Multiple Displays - Windows

Things you don't intend are also possible



Multiple Displays - Windows

Things to note:

- Windows can be opened anywhere on (and off) the complete desktop
- Windows can span display boundaries
- However maximizing will lock to one display
  - Where the window centroid is located
- Likewise full screen windows
- WGL Desktop size is considered outer rectangle spanning all displays
- Driver will typically send data to all GPUs (in case window is moved, etc.)
  - GPU Affinity OpenGL extension solves this (coming up in a few slides)

Multiple Displays - Windows
Part 1

```
DISPLAY_DEVICE lDispDev;

DEVMODE lDevMode;

lDispDev.cb = sizeof(DISPLAY_DEVICE);
```

Verify first display exists and get display settings

```
if (EnumDisplayDevices(NULL, 0, &lDispDev, NULL)) {
   EnumDisplaySettings(lDispDev.DeviceName, ENUM_CURRENT_SETTINGS,
      lDevMode);
}
```

Create Window on first display

```
g_hWnd1 = createWindow(hInstance, lDevMode.dmPosition.x,
  lDevMode.dmPosition.y, X0, Y0);


if (!g_hWnd1) {
   MessageBox(NULL, "Unable to create first window(s).", "Error",
  MB_OK); return E_FAIL;
}
```

Multiple Displays - Windows
Part 2

Verify second display exists and get display settings

```
if (EnumDisplayDevices(NULL, 1, &lDispDev, NULL)) {
        EnumDisplaySettings(lDispDev.DeviceName, ENUM_CURRENT_SETTINGS,
  &lDevMode);
}
```

Create Window on second display

```
g_hWnd2 = createWindow(hInstance, lDevMode.dmPosition.x,
  lDevMode.dmPosition.y, X1, y1);
if (!g_hWnd2) {
   MessageBox(NULL, "Unable to create second window(s).", "Error",
  MB_OK);  return E_FAIL;
}
```


Multiple Displays - Performance

Application didn’t specify which GPU to use!

We can use GPU Affinity to control this

Multiple Displays - Performance
GPU Affinity

- WGL extension (WGL_NV_gpu_affinity), core OpenGL not touched
- Application creates affinity-DC
  - `HDC wglCreateAffinityDCNV(const HGPUNV *phGpuList);`
  - Special DC that contain list of valid GPUs -> affinity mask
  - Affinity mask is immutable
- Application creates affinity context from affinity-DC
  - As usual with `RC = wglCreateContext(affinityDC);`
  - Context inherits affinity-mask from affinity-DC
- Application makes affinity context current
  - As usual using `wglMakeCurrent()`
  - Context will allow rendering only to GPU(s) in its affinity-mask

Multiple Displays - Linux

- Two traditional approaches depending on desired level of application transparency:
  - Separate X screens
    - 3D Windows can’t span X screen boundaries
    - Location of context on GPU allows driver to send data to only that GPU
    - Application needs to be multi-screen aware
  - Xinerama
    - One large virtual desktop
    - 3D Windows can span X screen boundaries
    - Will typically result in driver sending all data to all GPUs (in case window moves)

Multiple Displays - Linux

- Use `nvidia-xconfig` to create customized xorg.conf
  - Many command line options for extra control (e.g. stereo)
- `nvidia-settings` provides full featured control panel for Linux
- Drivers can capture EDID, and force it from a file
  - Useful when display device hidden behind KVM or optical cable

Multiple Displays - Performance

- GPU memory consumption
  - 3840x2160 desktop at 16x FSAA ~400MB of framebuffer.
- Performance
  - Fill-rate impact
  - Limited by the slowest renderer in the configuration
  - If using stereo, twice the frames being rendered

Multiple Displays - Synchronization

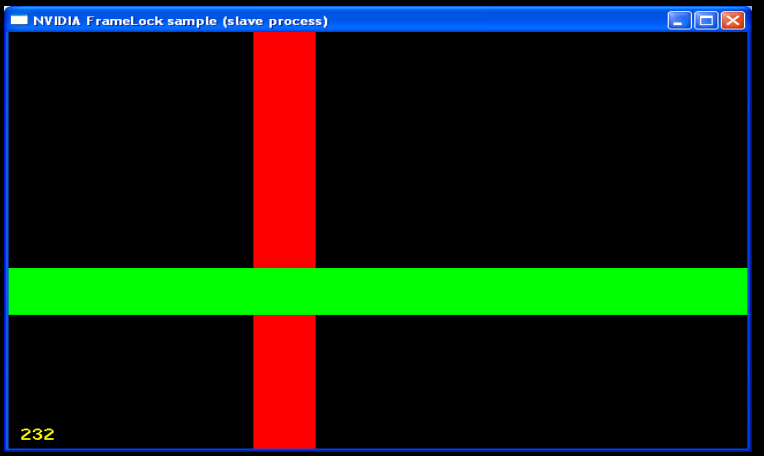
- NVIDIA G-sync
  - Synchronize vertical retrace
  - Synchronize stereo field
  - Enables swap barrier
  - Daisy-chain multiple G-sync cards

- OpenGL Extensions
  - Windows: WGL_NV_Swap_Group
  - Linux: GLX_NV_Swap_Group

Multiple Displays - Synchronization

NV_Swap_Group OpenGL Extension

- Synchronize the buffer swaps of a group of OpenGL windows
- Buffer swaps of members of the swap group will take place concurrently
- Synchronize the buffer swaps of different swap groups, from distributed systems
- Can be bound to a swap barrier
- Extends set of conditions that must be met before a buffer swap can occur
- Windows and Linux implementations



Multiple Displays - Synchronization
NV_Swap_Group OpenGL Extension

BOOL **wglJoinSwapGroupNV**(HDC hDC, GLuint group);
Adds drawable to swap group

BOOL **wglBindSwapBarrierNV**(GLuint group, GLuint barrier);
Binds group to the swap barrier

BOOL **wglQuerySwapGroupNV**(HDC hDC, GLuint *group, GLuint *barrier);
Returns group and barrier drawable is bound to

BOOL **wglQueryFrameCountNV**(HDC hDC, GLuint *count);
Returns current frame count for swap group

BOOL **wglResetFrameCountNV**(HDC hDC);
Resets framecounter to zero (should only be done on master)

Multiple Displays – Synchronization
Recommendations

- Eliminate extraneous GPU distractions
  - Control Panel will cause regular GPU contention when polling hardware status
- Use additional synchronization mechanisms in addition to swapbarrier
  - Broadcast frame count
- Create a G-sync topology with the Master in the middle of the daisy chain

SLI Mosaic Mode

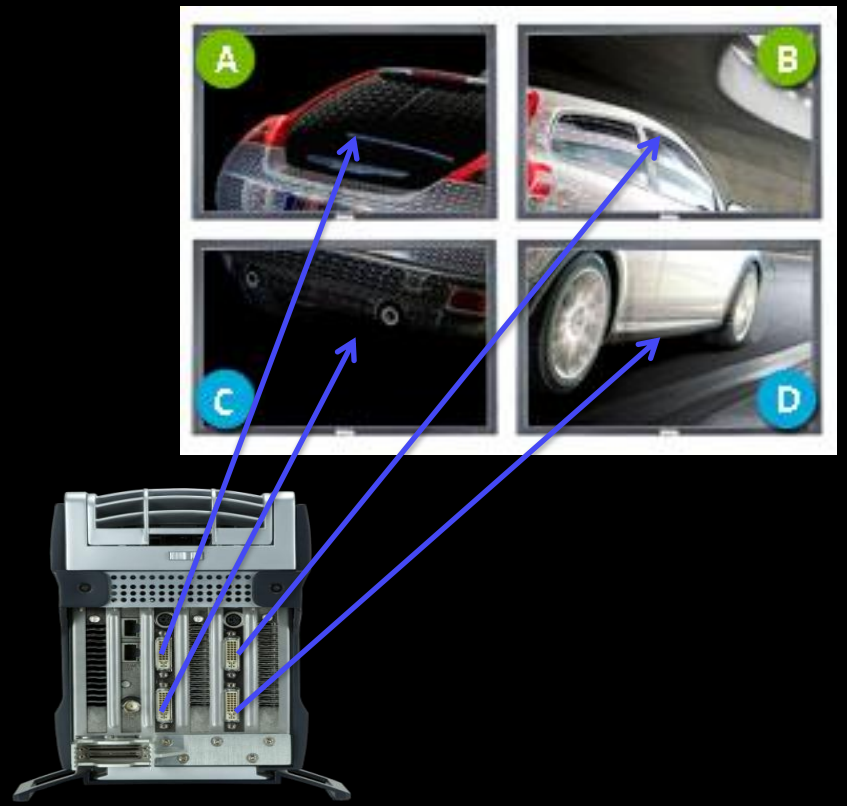
Multiple Displays made easy!

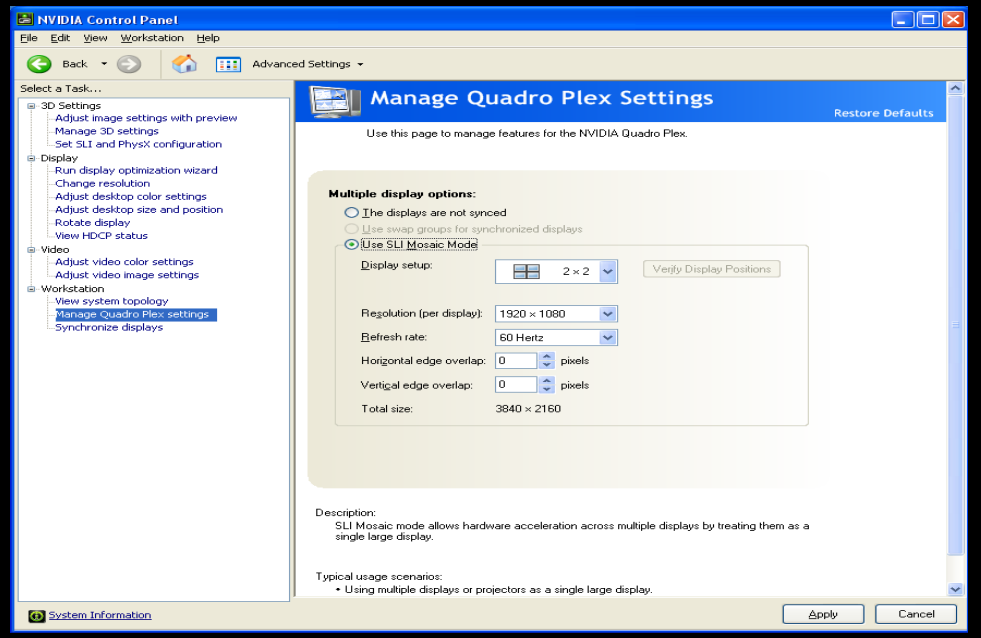
- Enables transparent use of multiple GPUs on multiple displays
  - Enables multiple GPUs to be seen as one logical GPU by the operating system
  - Applications '*just work*' across multi GPUs and multi displays
  - Works with OGL, DX, GDI etc
  - Coming soon: SLI Mosaic on SLI-capable workstations with Quadro FX 5800, Quadro 5000 and Quadro 6000 GPUs
- Zero or minimal performance impact for 2D and 3D applications compared with a single GPU per single display
- Doesn't support multiple View Frustums

SLI Mosaic Mode
Details

- Available in SLI-certified workstations
- Operating System support
  - Windows XP and Windows 7 (32 and 64 bit)
  - Linux (32 and 64 bit)
- Maximum desktop size = 16k X 16k (on Fermi-class GPUs)
  - FSAA may exacerbate desktop size
- Compatible with G-sync
  - Clustering tiled displays
- Supports Stereo

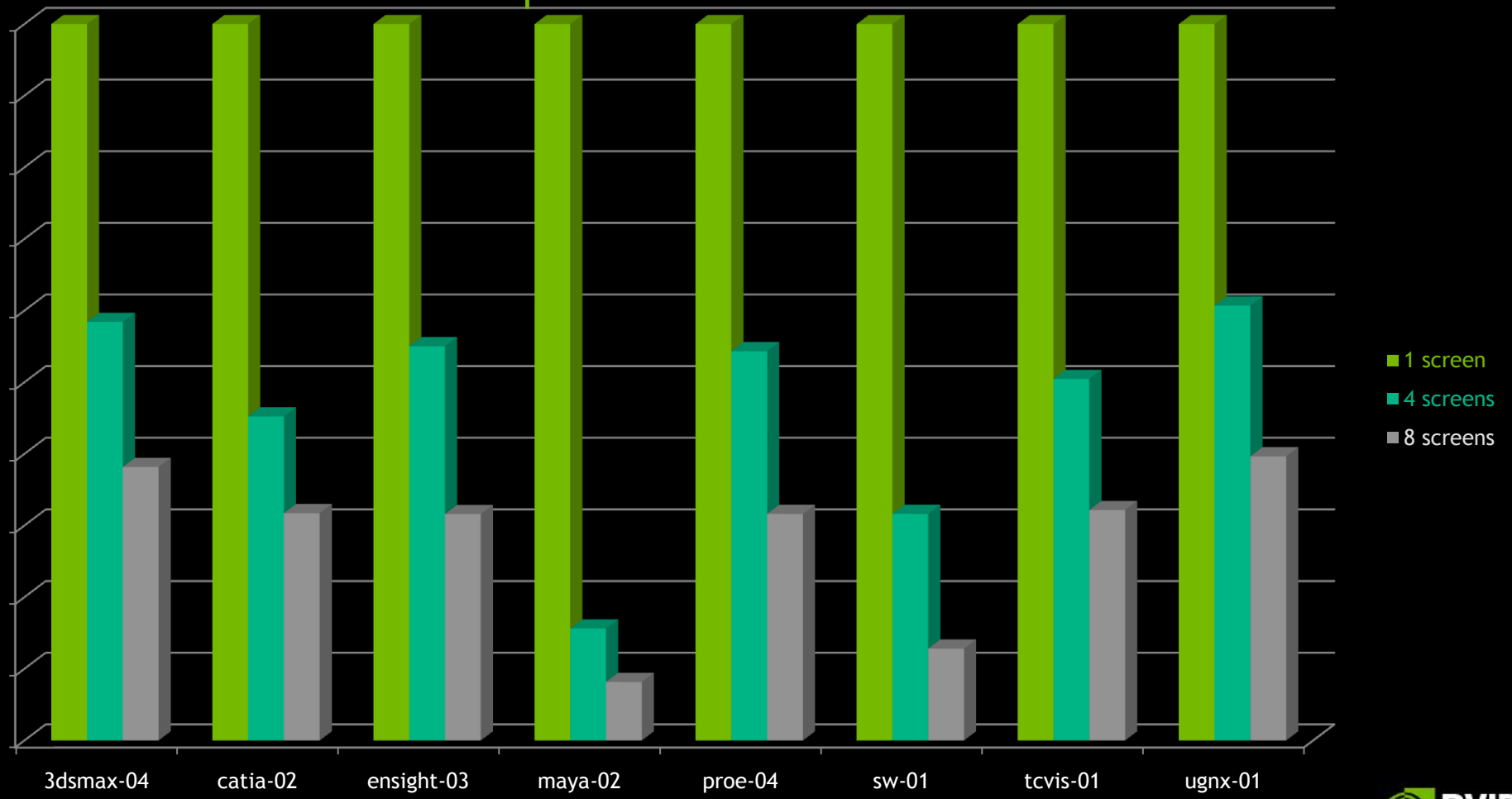
SLI Mosaic Mode
Configuration



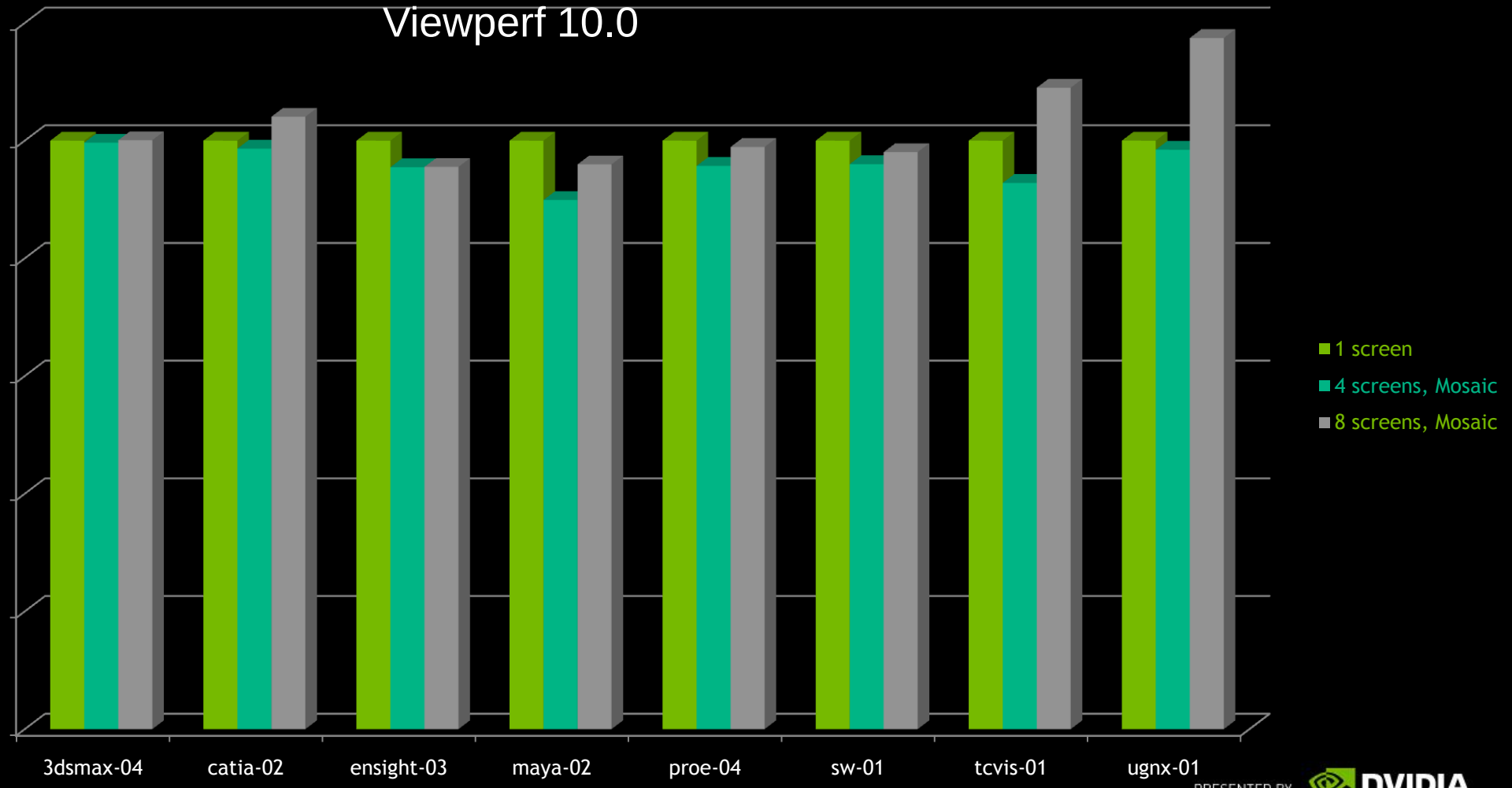


Performance Hit for Multiple Displays

Viewperf 10.0



SLI Mosaic Performance Advantage

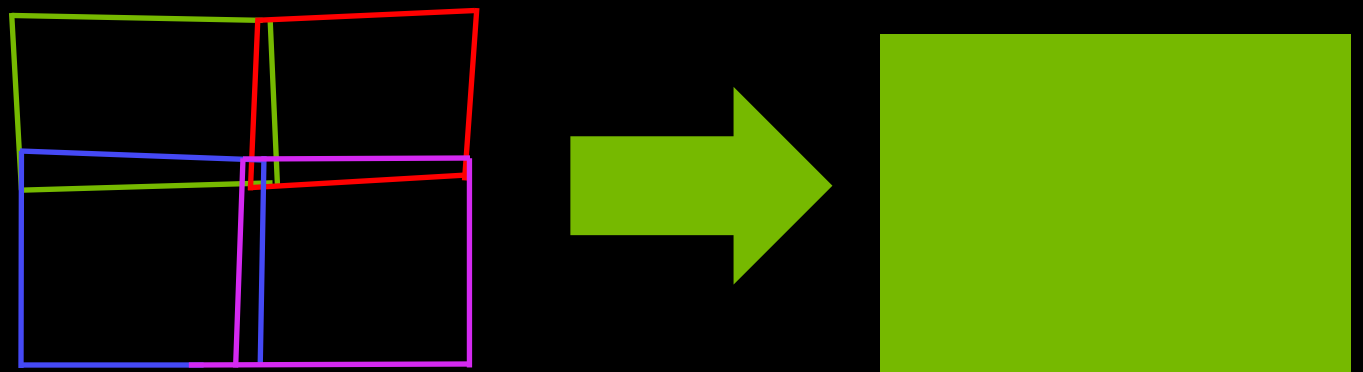


Beyond SLI Mosaic Mode

- Can combine Mosaic for partial set of all GPUs
- Use CUDA or Remote Visualization software for non-display GPUs
- May require “Manual” Configuration – contact us if you need help
- Combine Mosaic with CompleX Application Acceleration Engine

Warping and Blending

- New API enables the creation of 1 seamless display from many projected displays
- Additional hardware no longer necessary
- Available mid CY2011



Summary

- Demand for Large Scale Viz is being driven by economics
  - E.g. Digital Prototypes significantly less expensive than physical prototypes however demand high quality and realism
- Very large resolutions are de-facto standard for collaborative and large venue installations
- Pixel bandwidth requirements still require multiple channels, even with Display Port
- Some large venue displays are HDR capable

Summary - cont.

- Be aware of performance implications when using multiple GPUs
  - Use GPU affinity/Separate Xscreens
- Solutions like SLI Mosaic Mode extends the reach of Large Scale Visualization
- Combining solutions enables unprecedented realism and interactivity

Questions?