

Direct Send Compositing for Parallel Sort-Last Rendering

Stefan Eilemann, Renato Pajarola
Visualization and MultiMedia Lab
University of Zurich

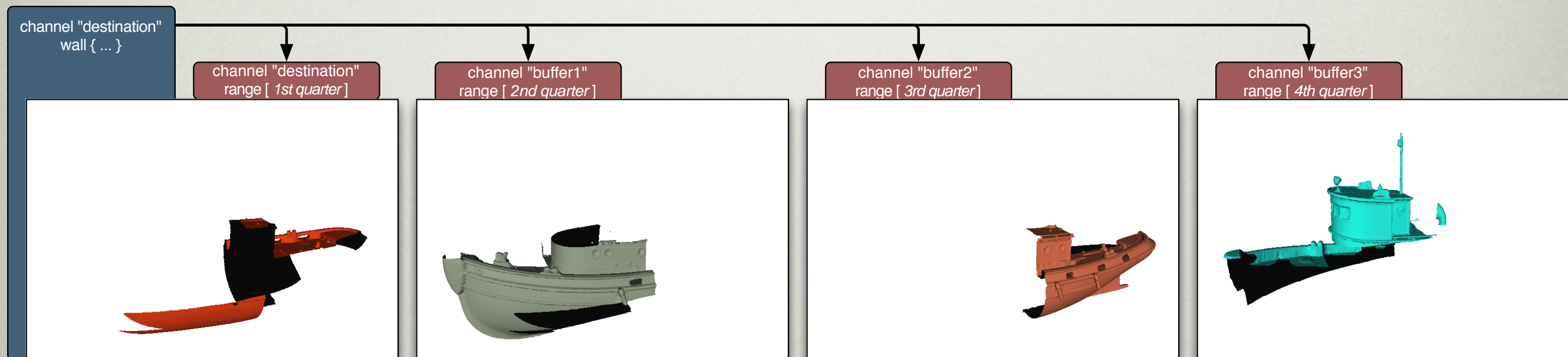
20. May 2007

Outline

- Parallel Sort-Last Rendering
- Compositing
- Direct-Send Compositing
- Binary-Swap Compositing
- Implementation
- Results

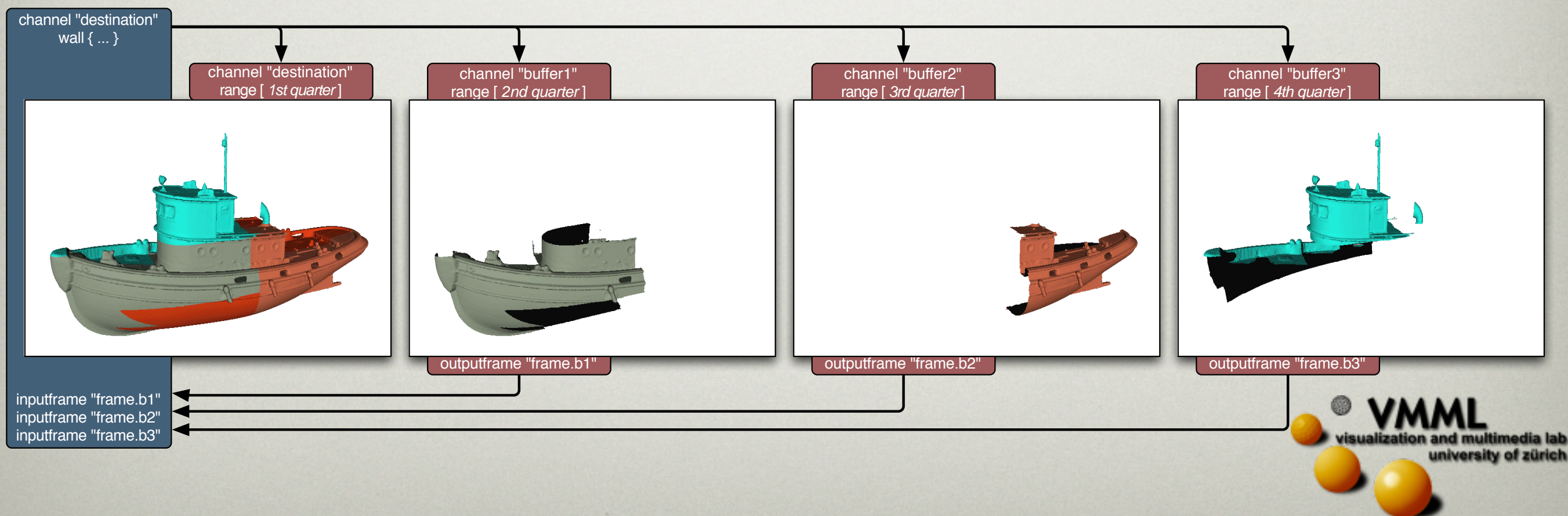
Parallel Sort-Last Rendering

- Run n rendering threads
- Each thread renders $1/n$ of the data
- Rendering performance scales nicely



Compositing

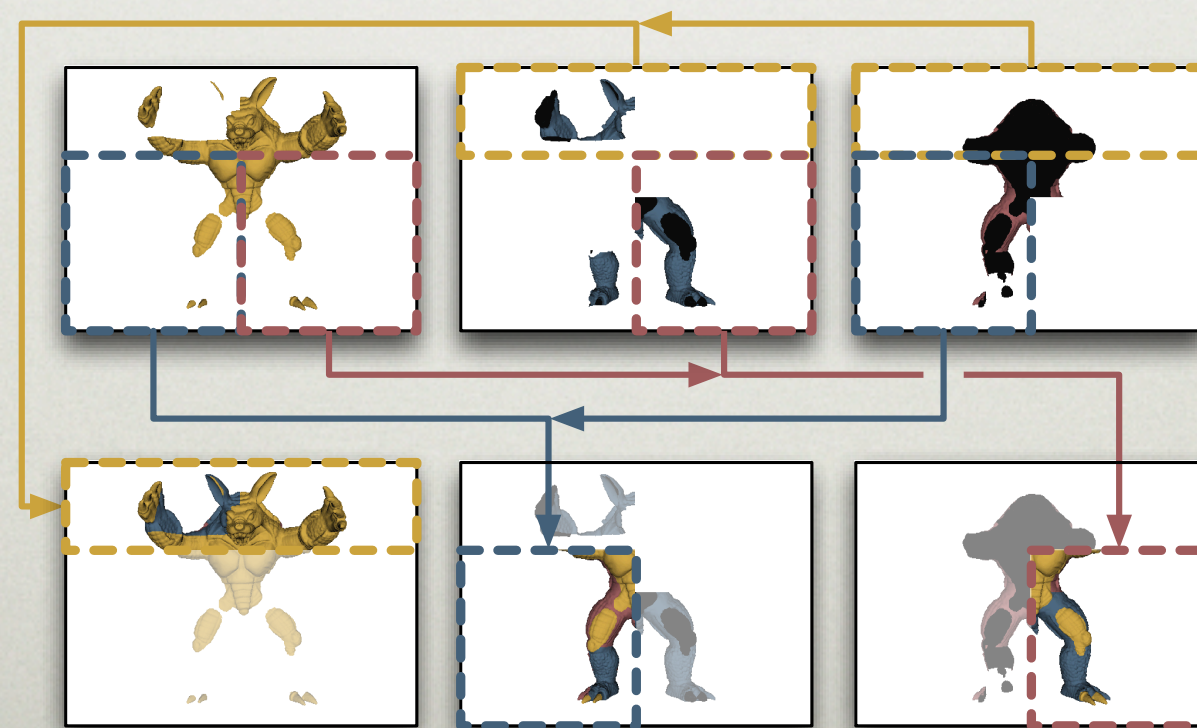
- Amount of pixel data is $O(n)$
- Polygonal data: color and depth
- Volume data: color and alpha



Direct-Send Compositing

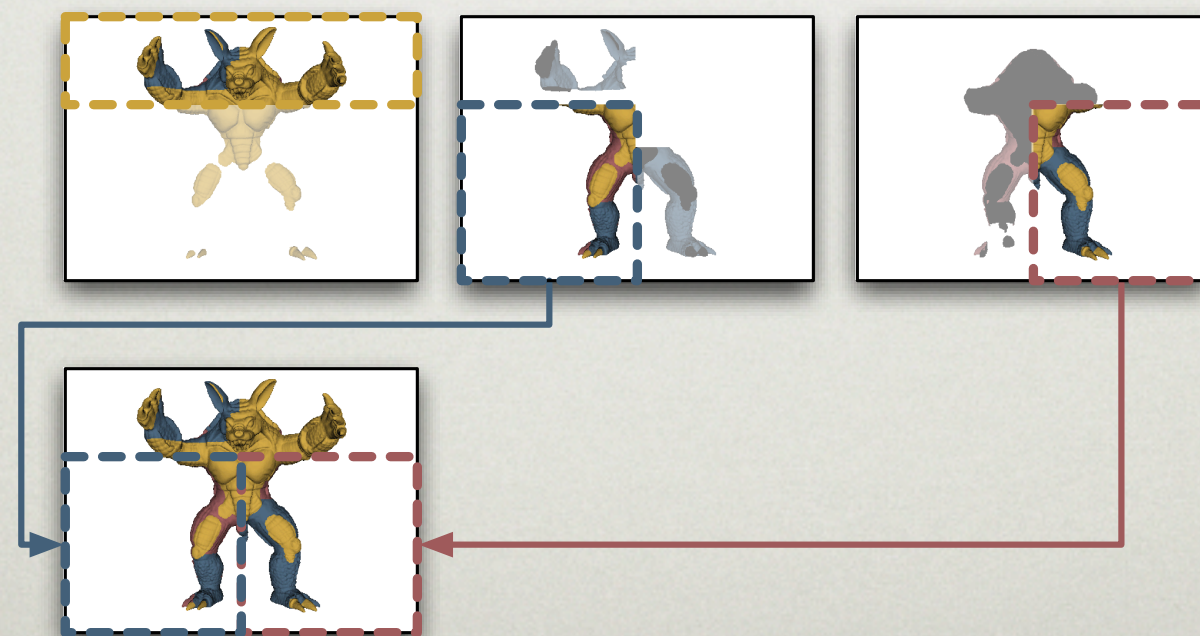
- Use all nodes for compositing
- Amount of data per node is $O(1)$

1. Each node composites one tile



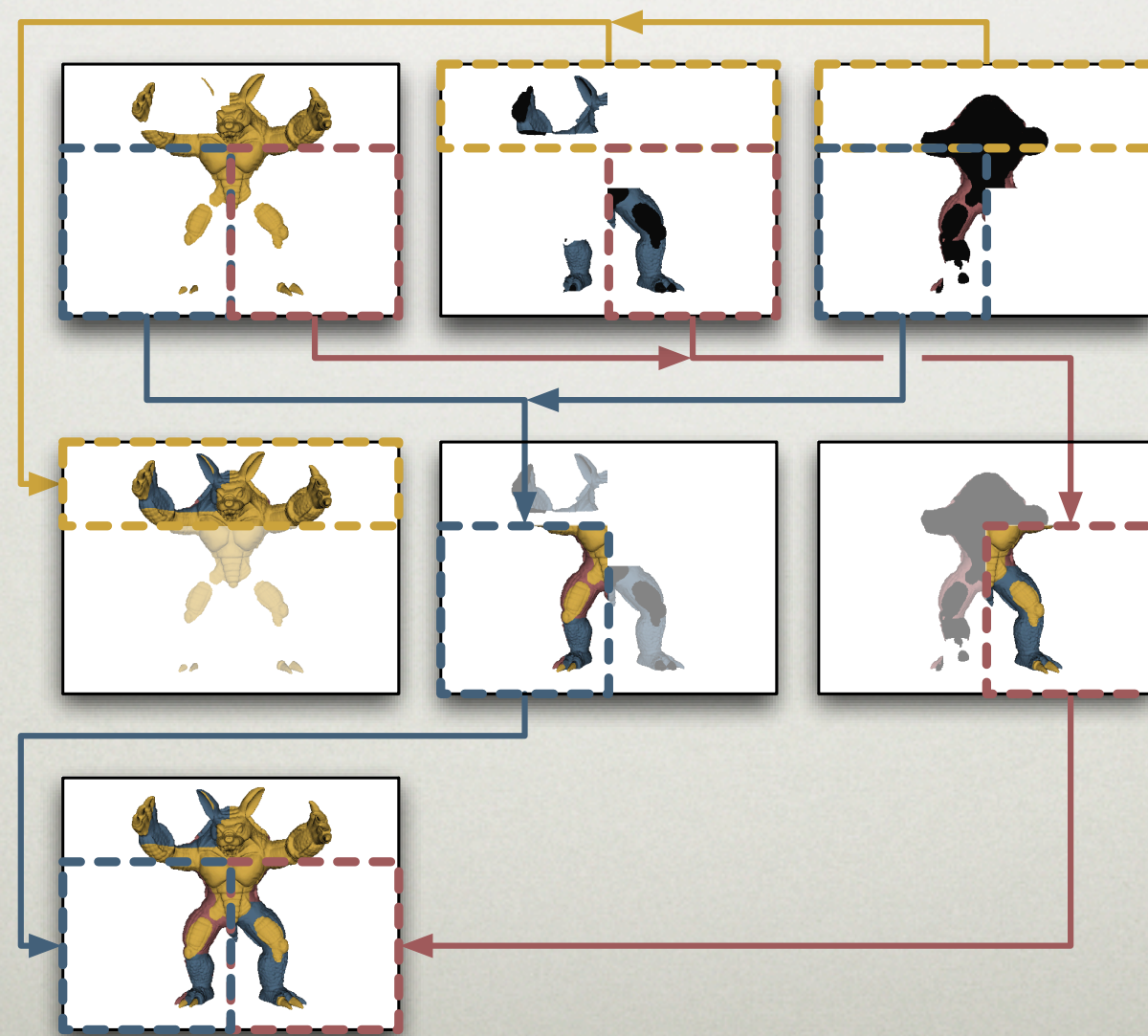
Direct-Send Compositing (2)

2. Assembled tile is send to destination
- Color information only - like sort-first
 - Destination channel's tile is in-place



Direct-Send Compositing (3)

- Any number of nodes
- Two sync points



Direct-Send Compositing

Demo

Binary-Swap Compositing

- Need power-of-two number of nodes

1. Swap half of framebuffer with partner

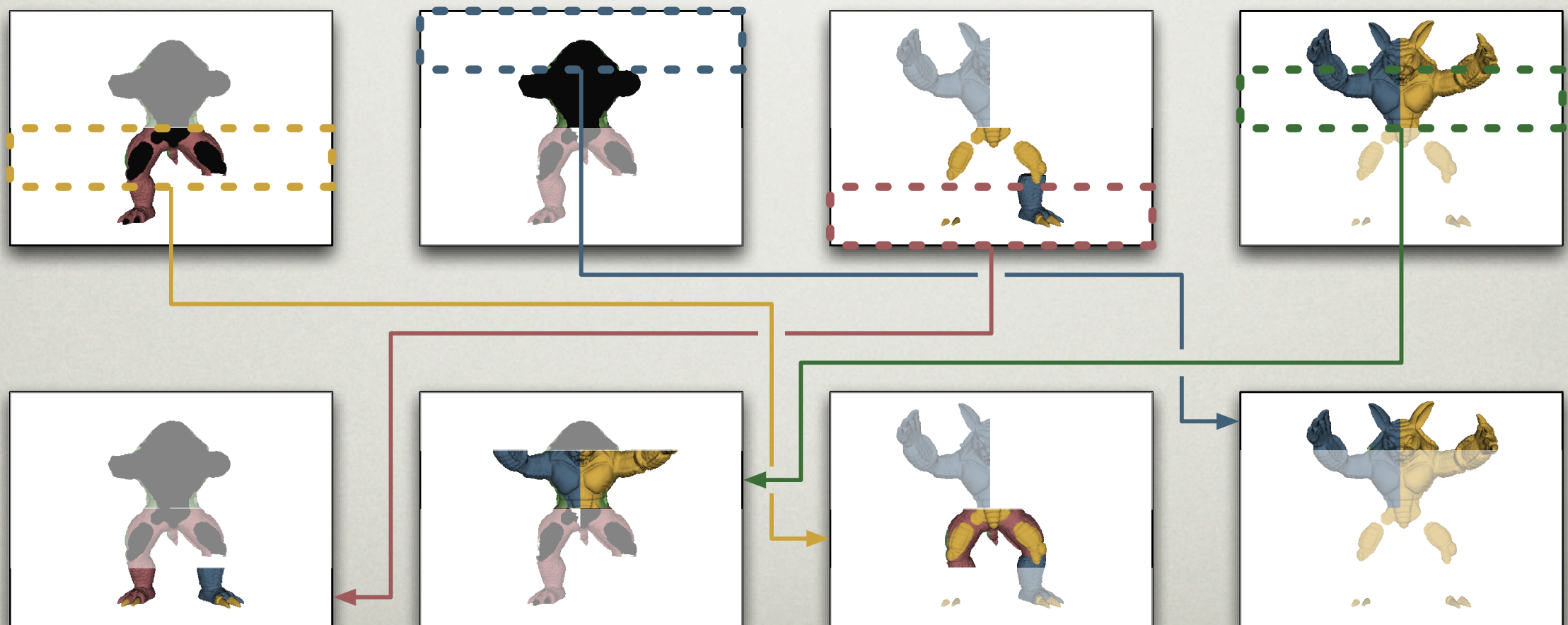
2. Composite



Binary-Swap Compositing (2)

3. Swap half of composited framebuffer with other partner

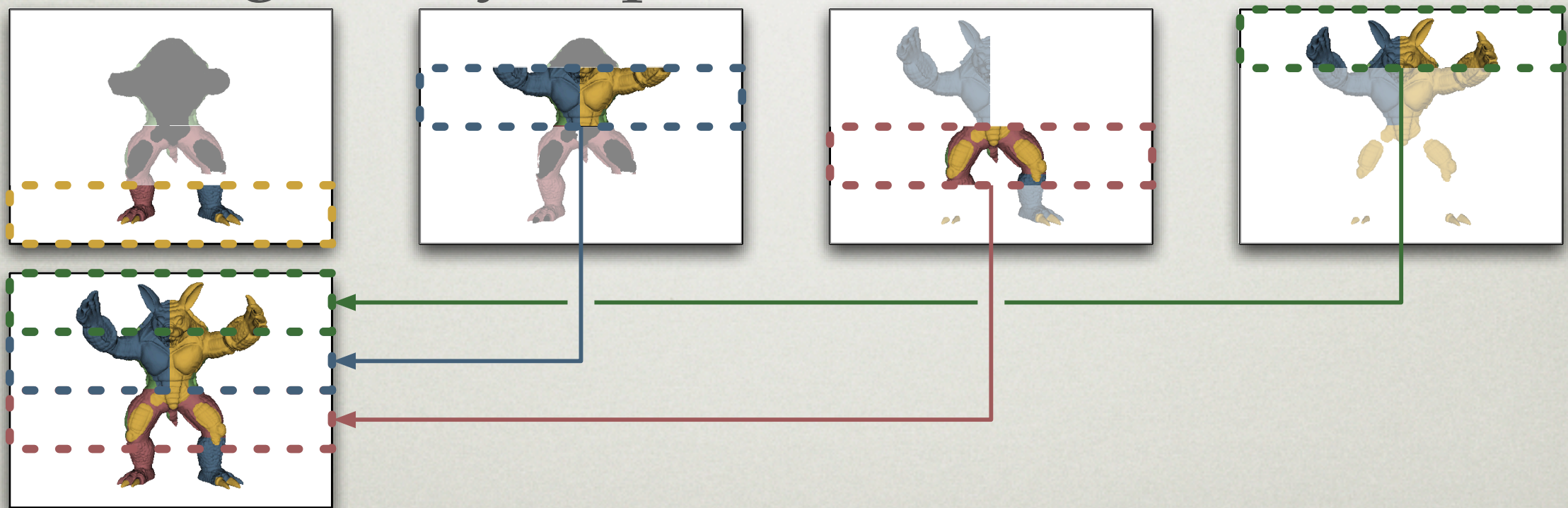
4. Repeat until own tile is complete



Binary-Swap Compositing (3)

5. Gather color tiles

- Power-of-two number of nodes
- $\log_2(n)$ sync points



Binary-Swap Compositing

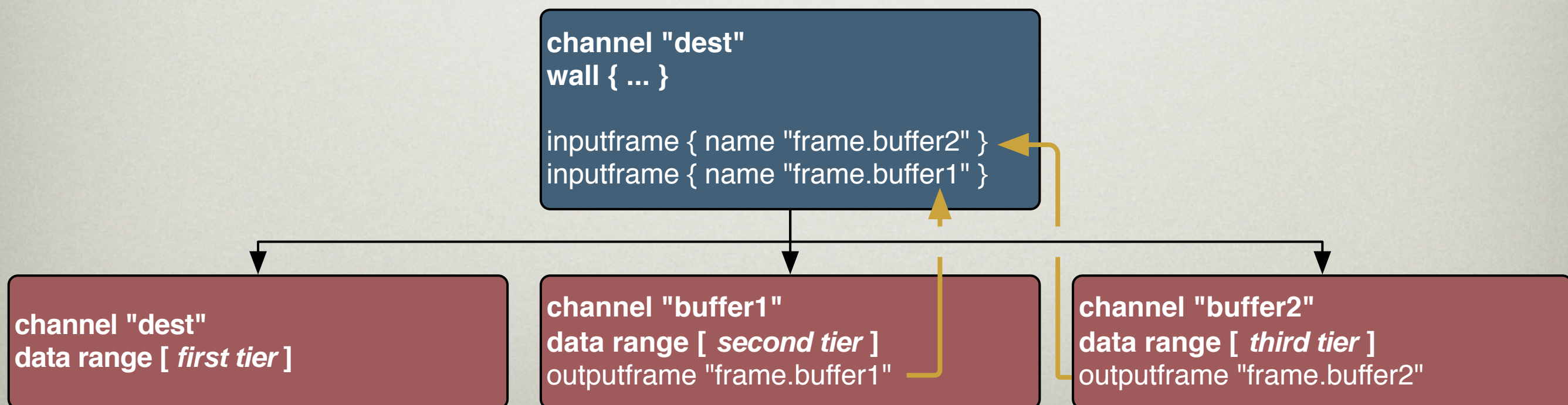
Demo

Implementation

- Part of Equalizer, an open source parallel OpenGL rendering framework
- Application provides OpenGL code
- Configured using compound tree
- Compositing based on output and input frames
- No hardcoded compositing algorithm!

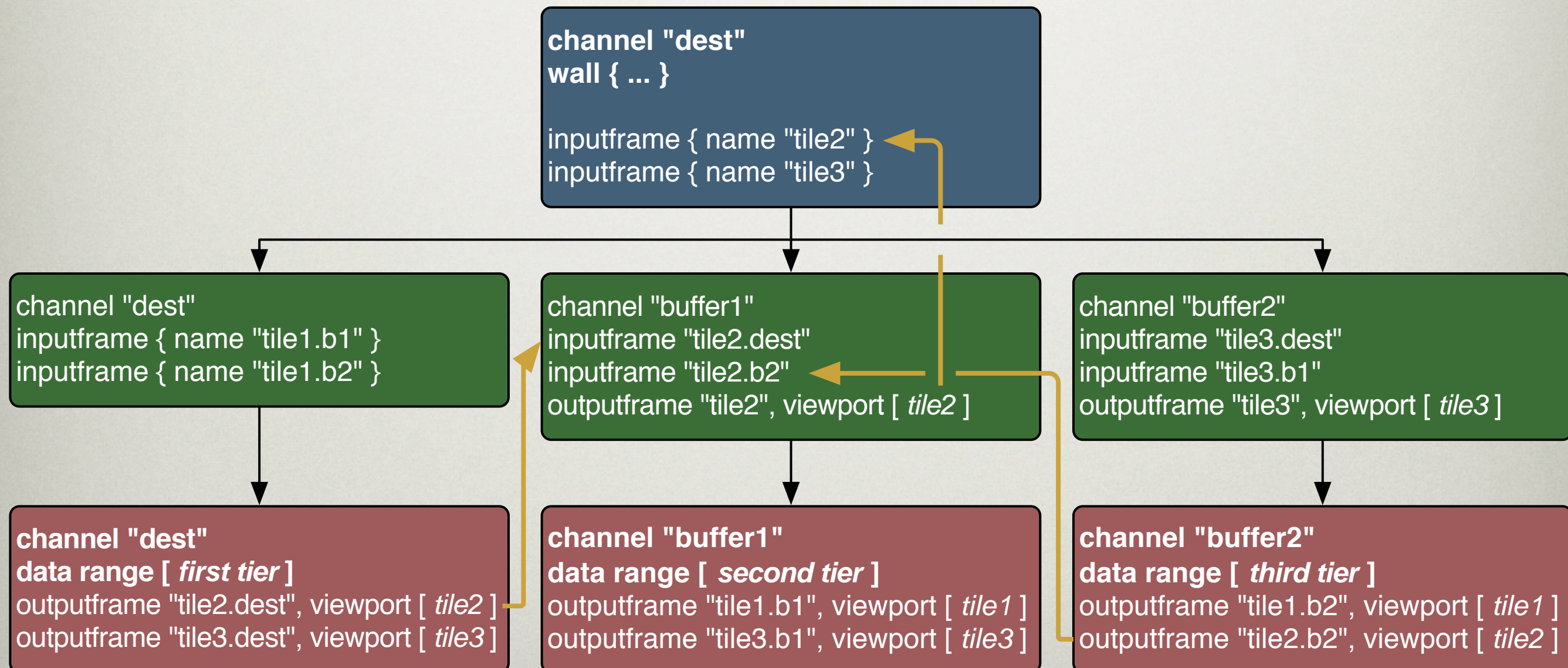
Implementation (2)

- Each compound has a channel
- Top-most channel has destination view
- Leaf compounds render for destination
- Sort-Last serial assembly tree:



Implementation (3)

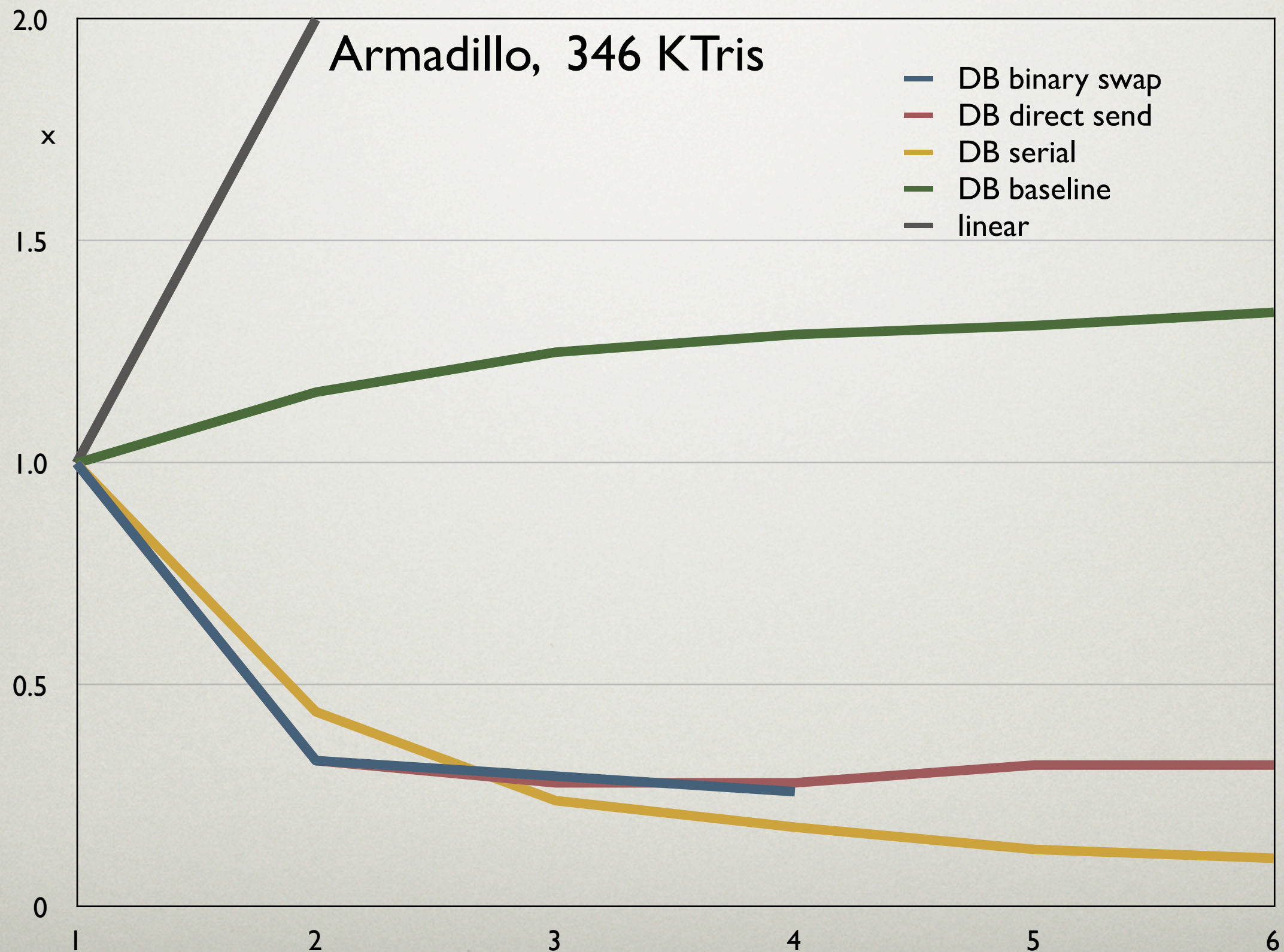
- Direct-Send compound tree:



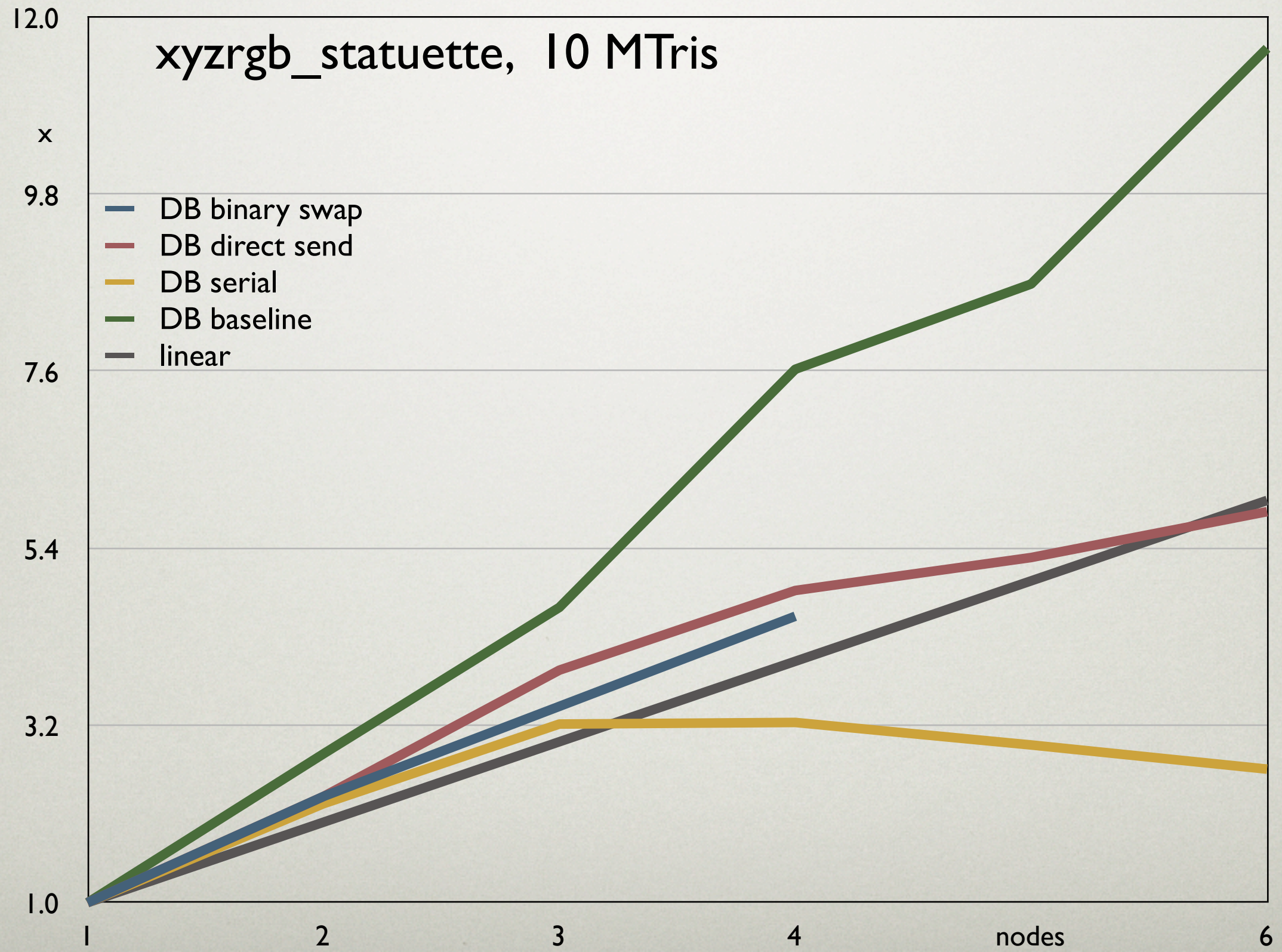
Results

- Hactar:
 - Six dual-Opteron nodes, 4 GB RAM
 - One Geforce 7800GTX per node
 - Gigabit Ethernet
- Polygonal Data: eqPly example
- Window size: 1280x800

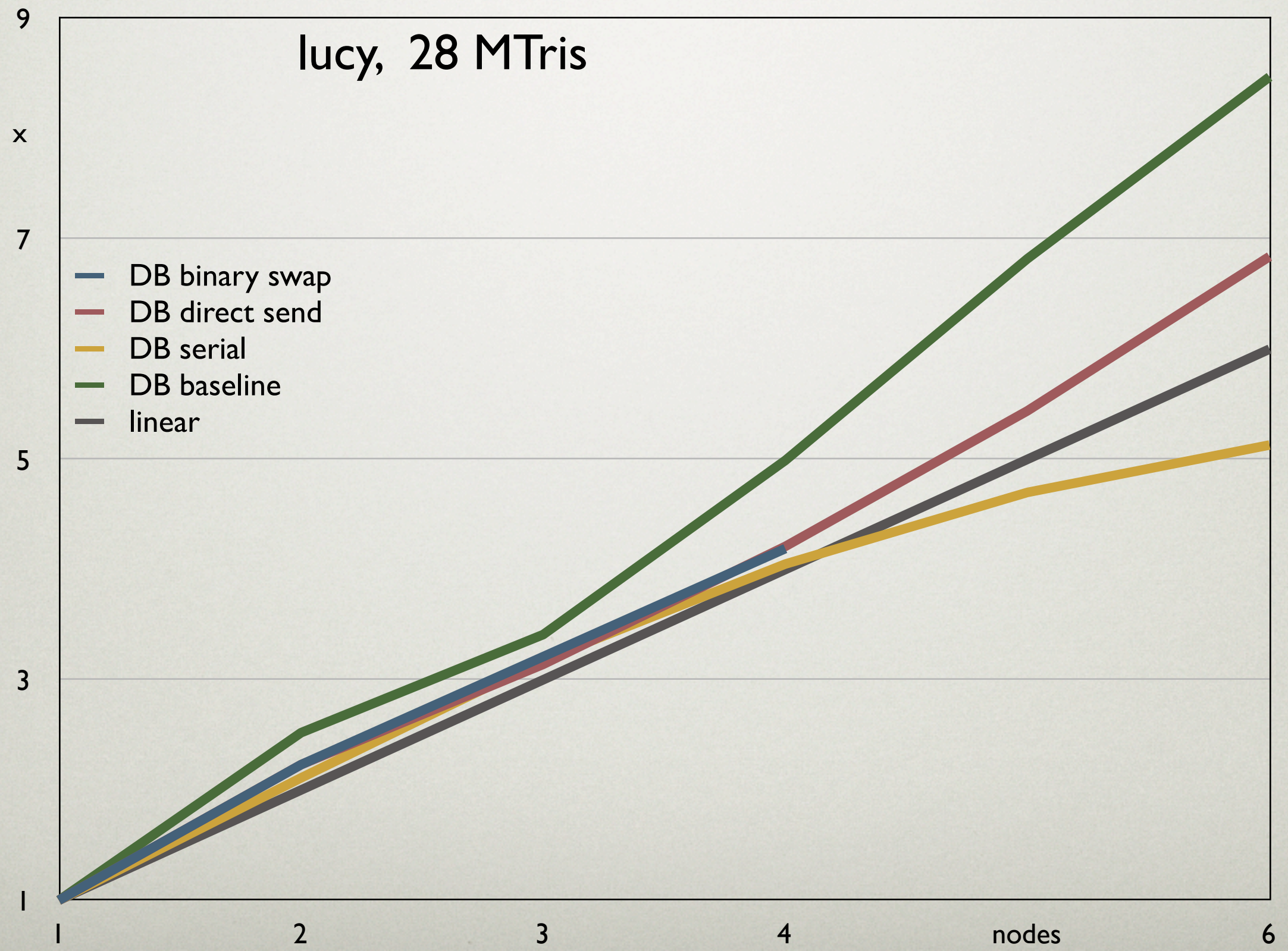
Small Model



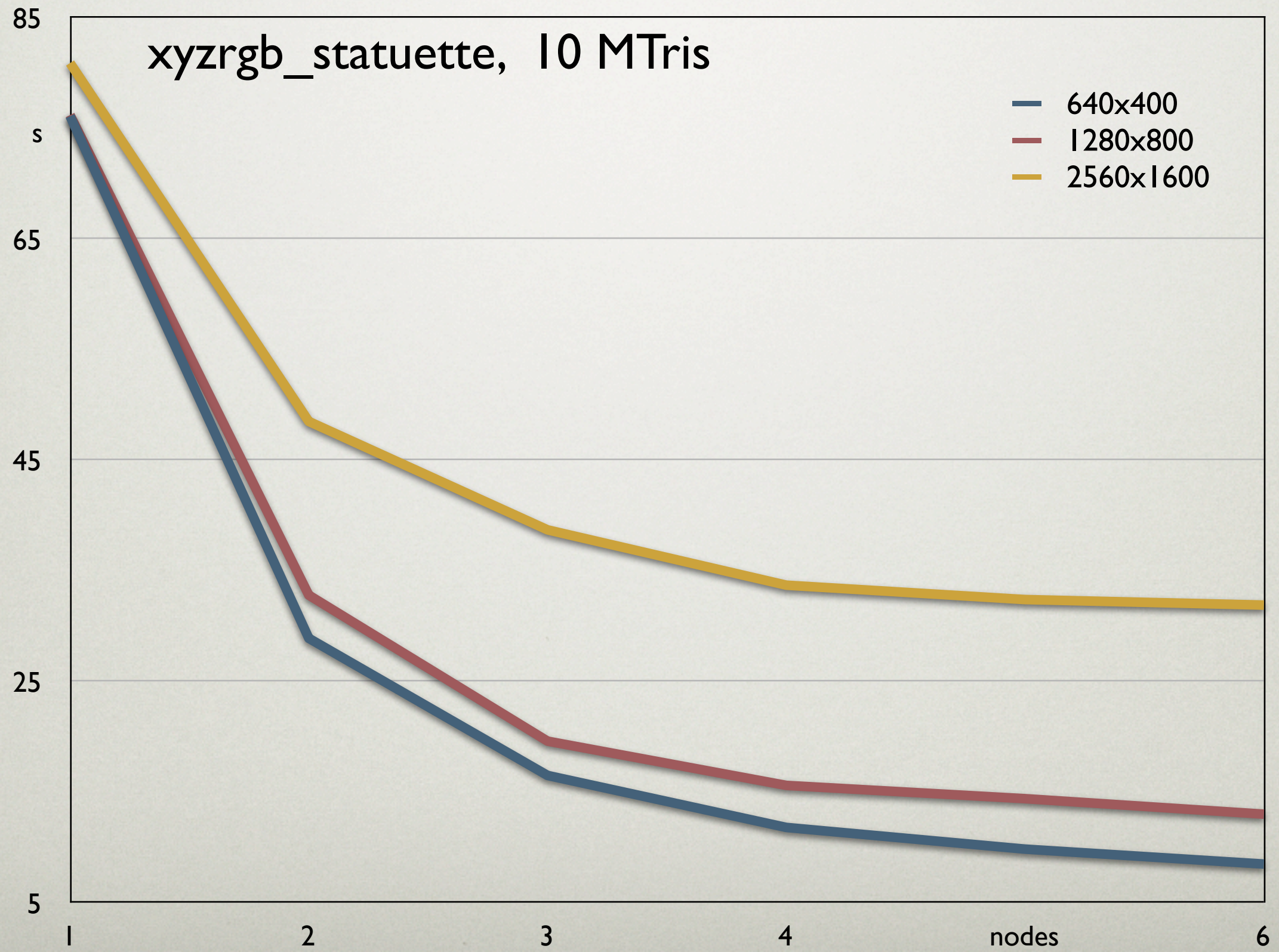
Medium Model



Large Model



Viewport Size



Results

- Direct-Send performs better or equal to binary-swap
- Direct-Send is more flexible
- Implementation behaves as expected
- Implementation is flexible for future algorithms and extensions

Future Work

- Full Equalizer Performance Evaluation
- Optimizations (ROI, SDP)
- 3D kd-Tree, VBO's for eqPly
- Volume Renderer

Last Words

- Website: www.equalizergraphics.com
- Linux, Windows, Mac OS X
- LGPL license
- Open standard for scalable graphics
- **Demo: tomorrow 5pm at CSCS**
- Questions?