TECHNISCHE UNIVERSITÄT
CHEMNITZ

Faculty of Computer Science

Computer Graphics Group

Final Diploma Examination

Communication Mechanisms for Parallel, Adaptive Level-of-Detail in VR Simulations

Author: Tino Schwarze

Advisors: Prof. Dr. G. Brunnett / Dipl.-Inf. M. Lorenz

Chemnitz, 6th May 2003

Introduction

- VR simulations require very complex scenes to allow high degree of interaction and realism

- Erikson 1996[1]:

  > For every computer graphics system, there exists a model complex enough to bring its performance to a crawl.

- speeding up rendering is researched a lot

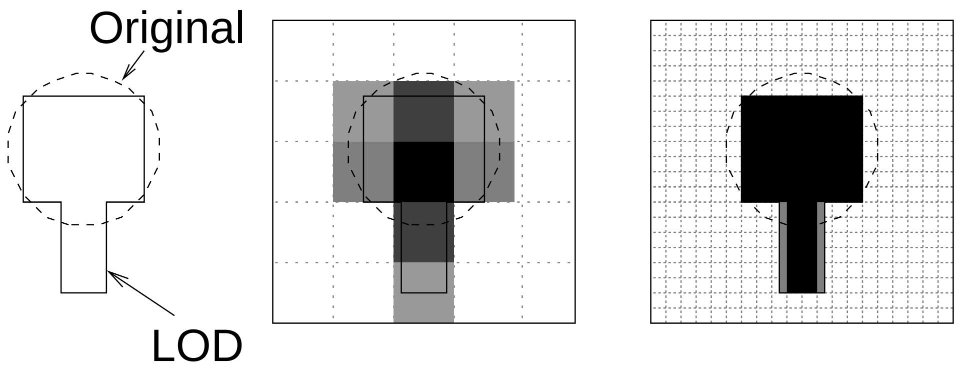
- assumption here: scene consists of many complex objects
  $\Rightarrow$ skip rendering of unobservable details

- common approach: *Level-of-Detail* (LOD)

[1] Erikson, Carl. 1996. *Polygonal Simplification: An Overview*. University of North Carolina, Chapel Hill. Technical Report TR96-016.

Static Level-of-Detail

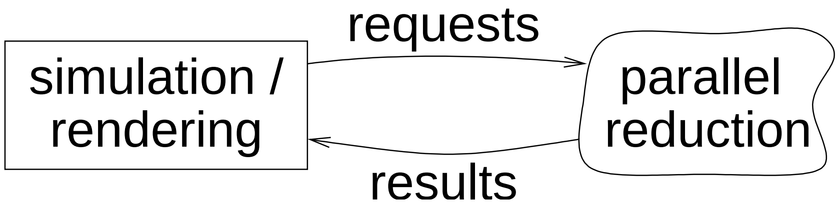
- fixed number of detail levels for different distances
- modelled manually or via preprocessing (very time-consuming)
- weaknesses
  - might contain superfluous detail levels
  - lowest level might still be too detailed for certain situations
  - independent of picture geometry (low vs. high resolution output device, e.g. mobile phone vs. Powerwall at the Visualize Center)



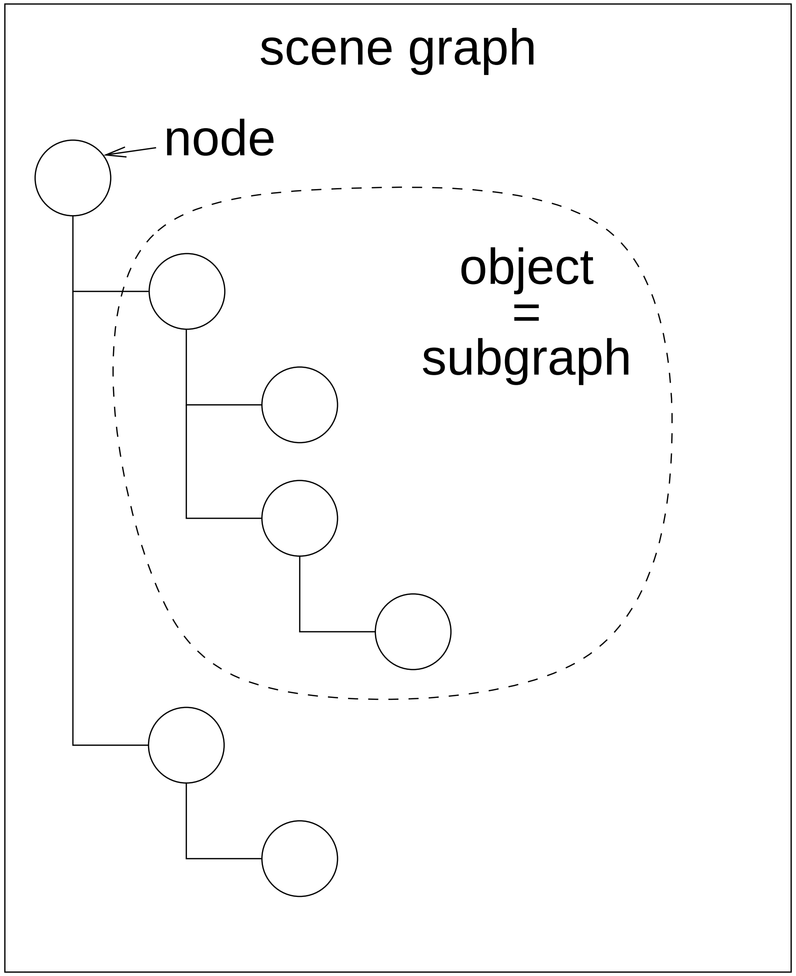
Adaptive Level-of-Detail

- simplifications of objects computed on demand
- can be fitted to current viewing situation
- geometry simplification usually computing intensive $\Rightarrow$ try to parallelize
- algorithms themselves usually not parallelizable

$\Rightarrow$ parallel reduction of disjoint parts of the scene (here: subgraphs)



Notations



Geometry Simplification

- classes of algorithms
  - decimating vs. generating (refining)
  - topology preserving vs. topology altering
- constraints on topology of original geometry (e.g. only manifold surfaces[2])
  here: polygon sets as input $\Rightarrow$ objects of arbitrary topology
- only few algorithms known which work on arbitrary topology
- examples:
  - Uniform Vertex Clustering (Rossignac and Borrel 1993)
  - Pair Contraction with Quadric Error Metrics (Garland and Heckbert 1997)

[2] A manifold is a surface for which the infinitesimal neighborhood of every point is topologically equivalent to a disk.

Reuse of Reductions

- computing a simplification is too slow for realtime $\Rightarrow$ latency too high

$\Rightarrow$ computed reductions should be cached

- validity of cached reductions *may* depend on
  - picture geometry, camera parameters
  - object geometry (includes transformation of parts of it)
  - distance from camera
  - viewing angle
  - lighting[3]

- simplification algorithm delivers validity constraints with reduced geometry

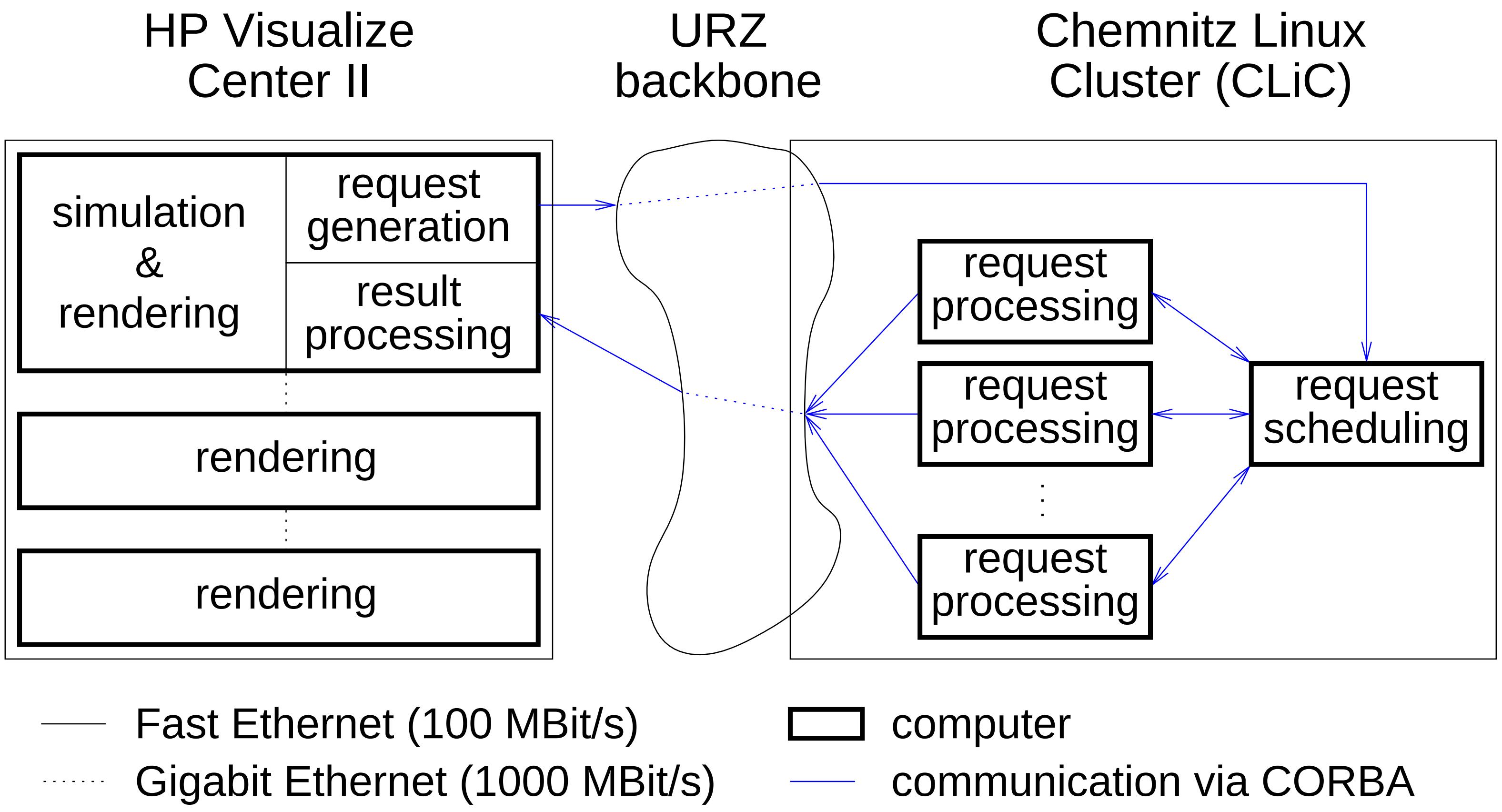
---

[3] Xia, Julie C. / Varshney, Amitabh. 1996. *Dynamic View-Dependent Simplification for Polygonal Models*. In: Proceedings of the IEEE Visualization 96.

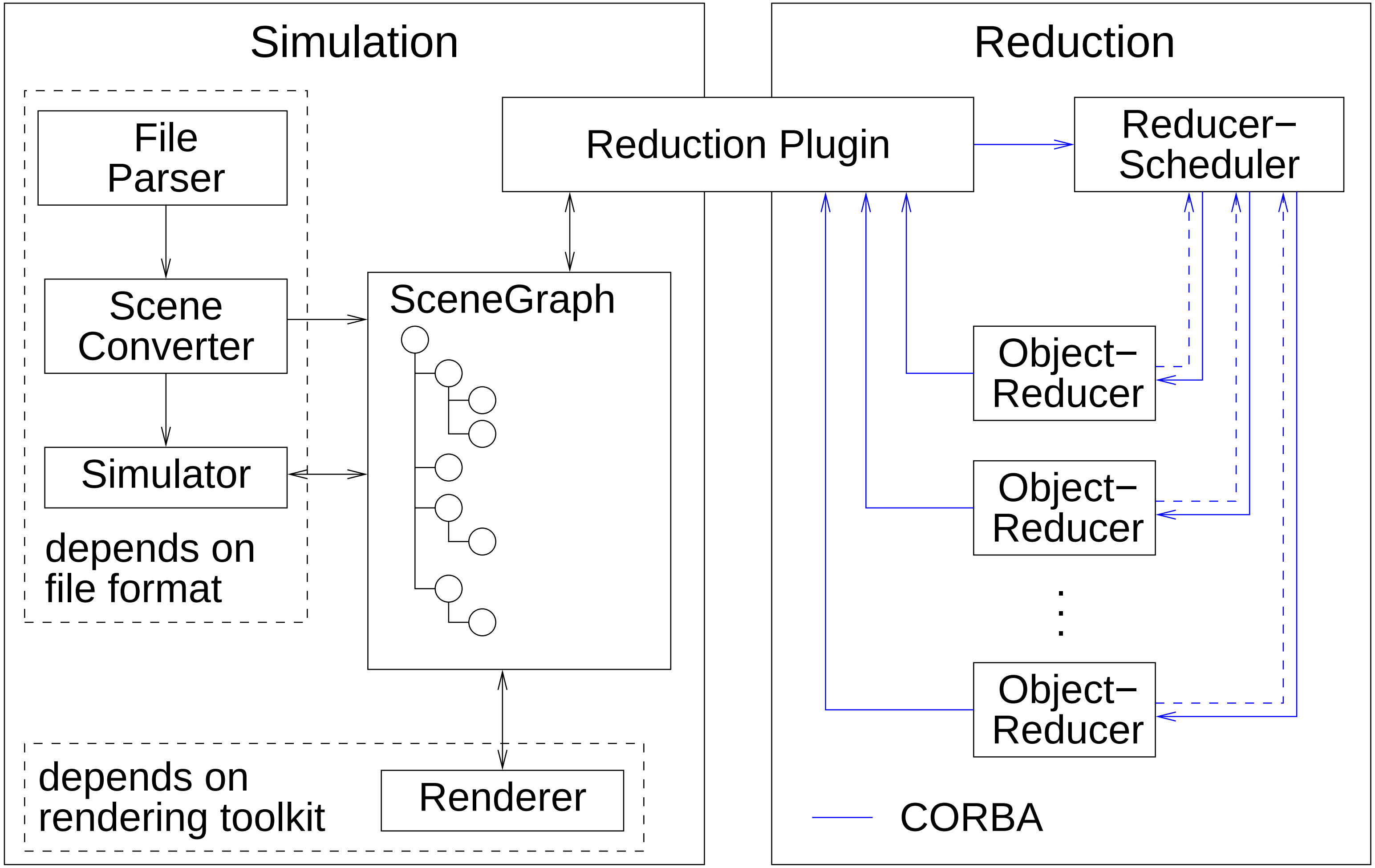
System Architecture

- required system components
  - simulation
  - rendering
  - simplification
- architecture depends on available resources:
  - HP Visualize Center II at the VR lab
  - Chemnitz Linux Cluster (CLiC)
- simulation and rendering performed by Visualize Center
- simplification performed by CLiC
- CORBA used for communication

Physical Architecture



Software Architecture



Scheduler Purpose

- central component of distributed application
- distributes scene graph
- accumulates changes in transformation and propagates to reducers on demand (no further changes in scene supported yet)
- distributes per-frame information on demand
  - picture geometry (width and height)
  - modelview and projection matrix

Results

- evaluated algorithms:

  PropSlim (Michael Garland): based on pair-contraction, uses quadrics for accumulating error

  vtkDecimatePro improved version of Schroeder's algorithm described in "Decimation of Triangle Meshes" (1992)

  vtkQuadricDecimation implemented after Hughues Hoppe's Vis '99 paper "New Quadric Metric for Simplifying Meshes with Appearance Attributes"

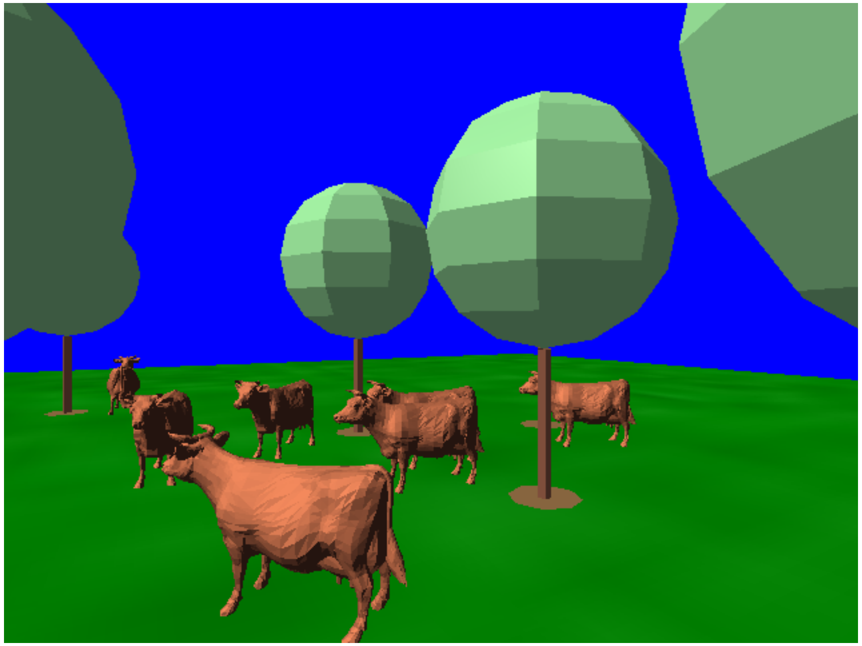
  vtkQuadricClustering implemented after Peter Lindstrom's Siggraph 2000 paper "Out-of-Core Simplification of Large Polygonal Models"

- algorithms sub-optimal for reduction of VR objects: only PropSlim and vtkQuadricDecimation support vertex attributes

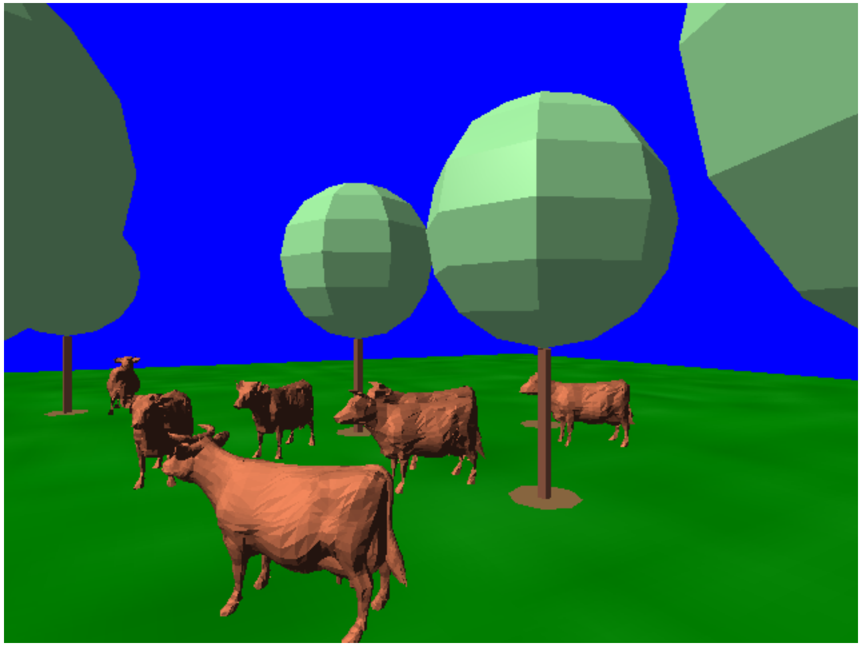
$\Rightarrow$ only suitable for special scenes

Benchmark

- scene with several objects and animated camera movement



without reduction



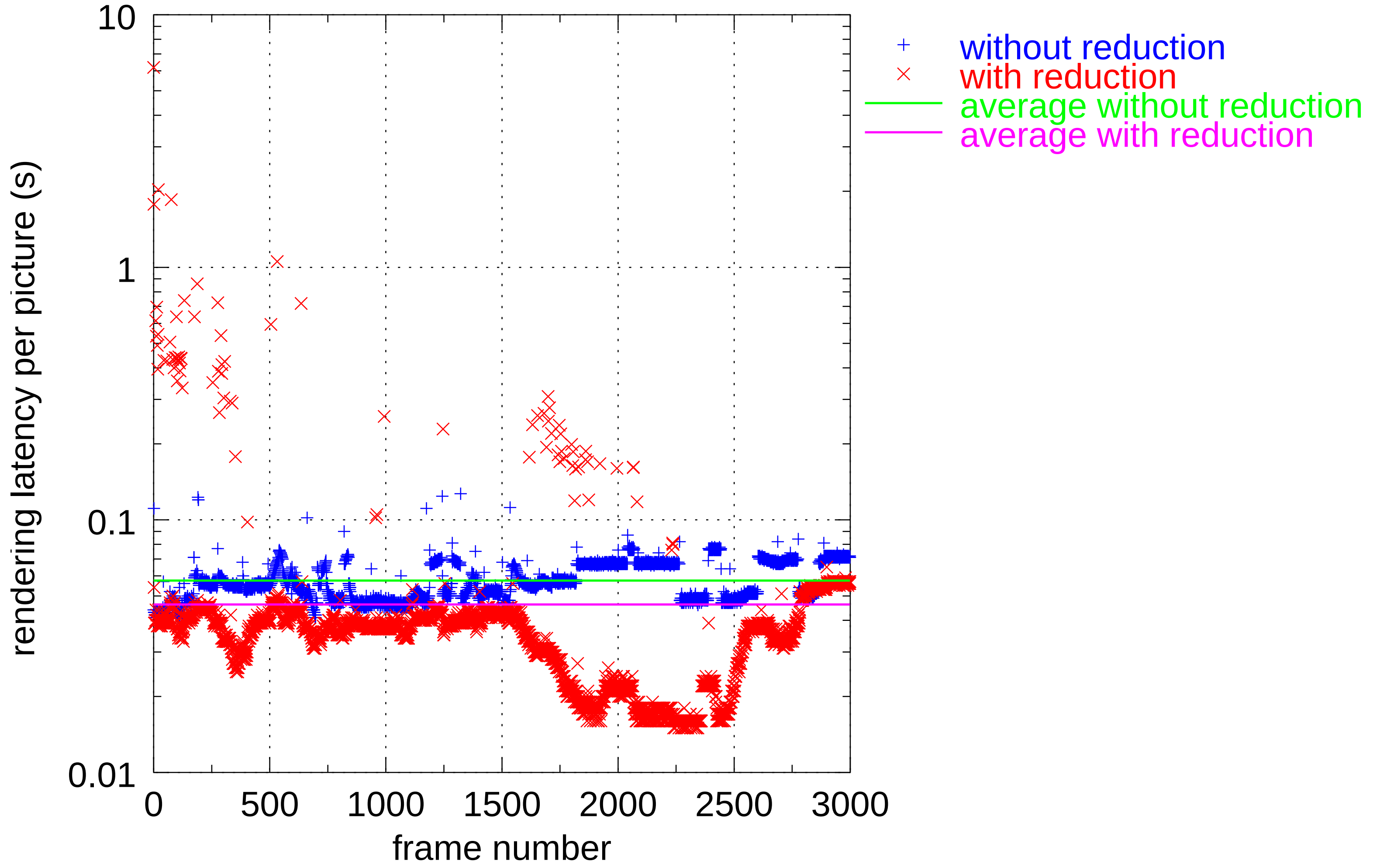
with reduction

differences hardly noticeable!

Benchmark

- duration: 300 seconds in simulation time at 0.1 s per frame = 3000 frames
- reduction by 7 nodes of CLiC (+ 1 for scheduling)
- measured latency between start and end of rendering (does not include simulation, does include request generation, receiving and processing)
- improvement in average rendering latency
  - 17 frames/s to 22 frames/s (30 % speedup)
  - 23 frames/s when not counting one-time initialization (35 % speedup)

Benchmark Result



Further Work / Suggested Improvements

- asynchronous rendering (don't wait for results to be available)
  - how many outstanding reductions?
  - worst case: reductions not useable when arrived
- introducing abstract reduction parameter might be useful (would allow speculative request of reductions)
- heuristics to figure out ”usefulness” of a reduction in current situation
  - beneficial for cache purging
  - asynchronous rendering could use most useful reduction in meantime
- more efficient transport of geometry data

Summary

- rendering can be speed up by using adaptive Level-of-Detail with parallel computation of detail levels
- scenes have to be suitable for this task – implementations of simplification algorithms not yet suitable for all scenes
- adaptive Level-of-Detail probably works best with long VR sessions where a large cache of reductions can be built up
- architecture useful for other purposes, e.g. distributed rendering (with appropiate special hardware)

Thank You for Your attention.