

# Introducing a New Client-Server Framework for Large CFD Models

Andrea Paroni  
Paolo Geremia  
Stefano Valeri

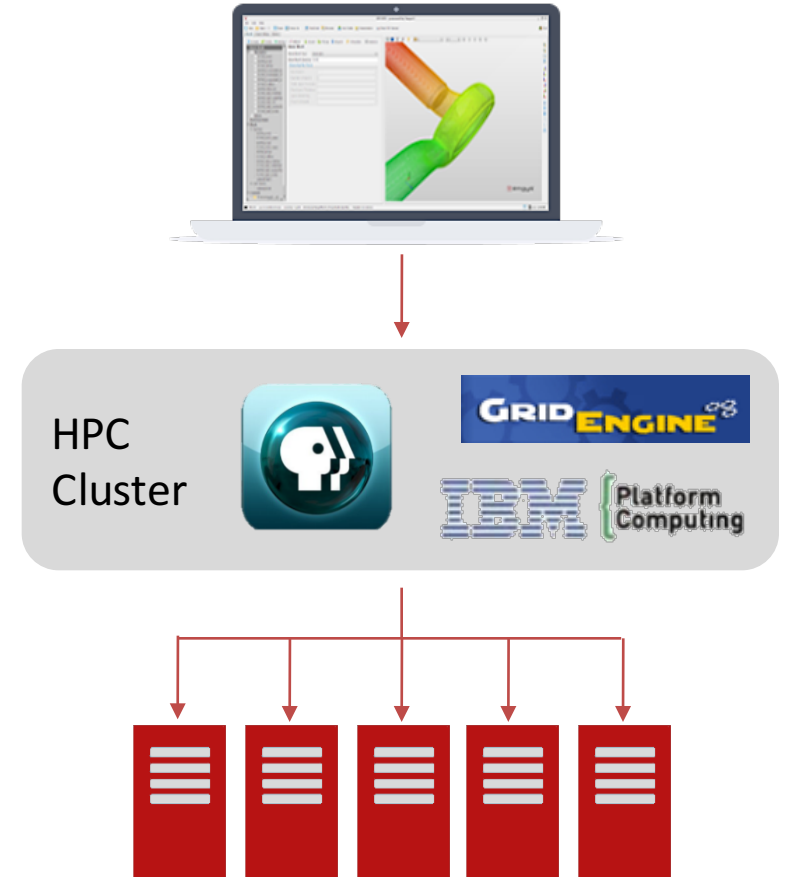


11<sup>th</sup> OpenFOAM® Workshop  
26<sup>th</sup> - 30<sup>th</sup> of June, 2016  
Guimarães, Portugal

- Motivation
- HELYX<sup>®</sup> Client-Server Network Architecture
- HELYX<sup>®</sup> HPC Usage
- Demo
- Future Developments

## Current interface available in HELYX®

- Control parallel and remote executions
- Shared or Distributed memory
- Connect to any machine with a network/Internet connection
- Asynchronous client-server
- Supports queue system for HPC clusters



## **Current interface available in HELYX<sup>®</sup>**

- Large dataset not suitable for transfer
  - Network communication bottleneck
- Asynchronous operations are not feasible
  - Missing fully synchronous client/server architecture
  - Missing remote visualisation of CFD results data

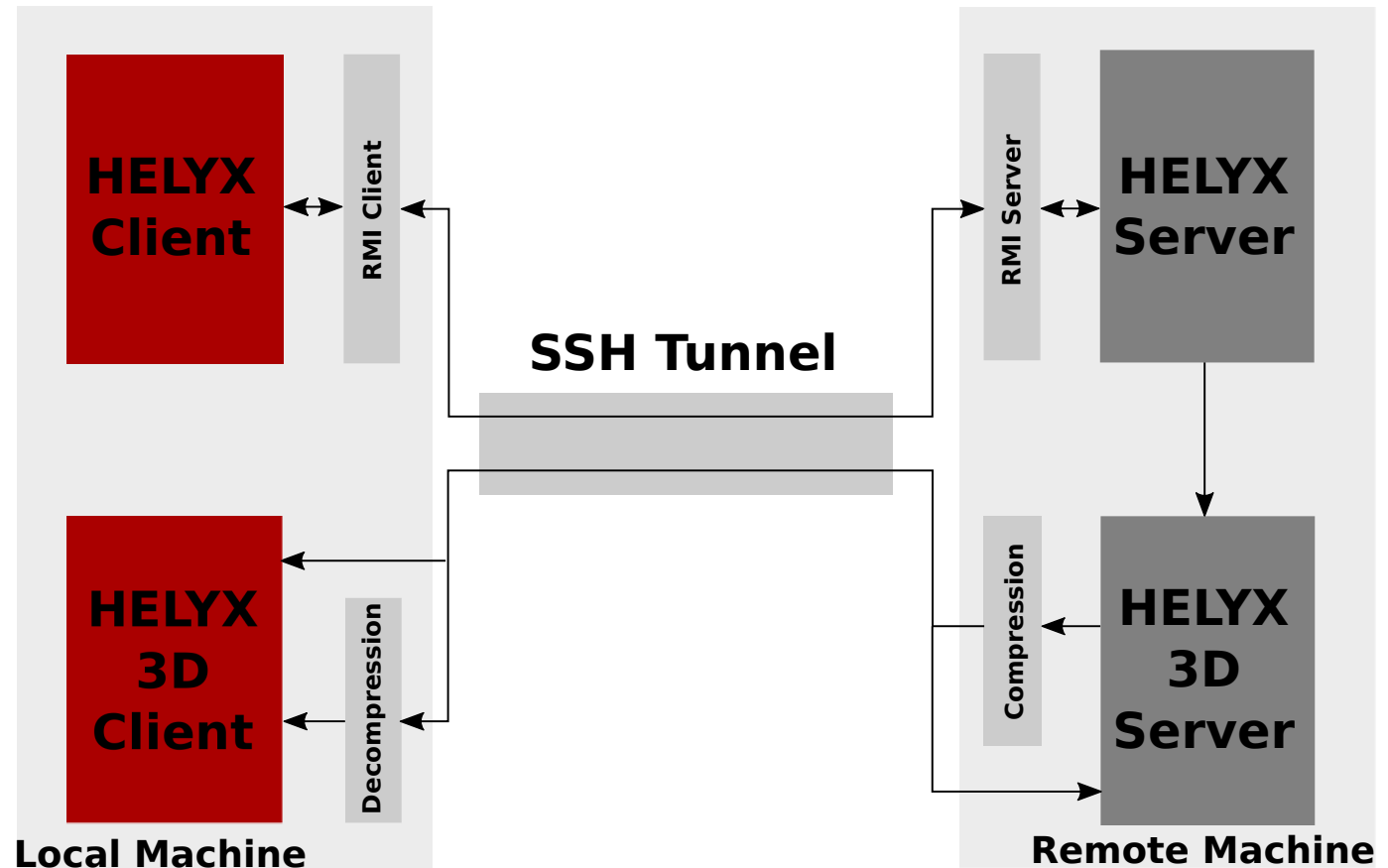
**Solution** → New HELYX<sup>®</sup> Client-Server

- Motivation
- **HELYX<sup>®</sup> Client-Server Network Architecture**
- HELYX<sup>®</sup> HPC Usage
- Demo
- Future Developments

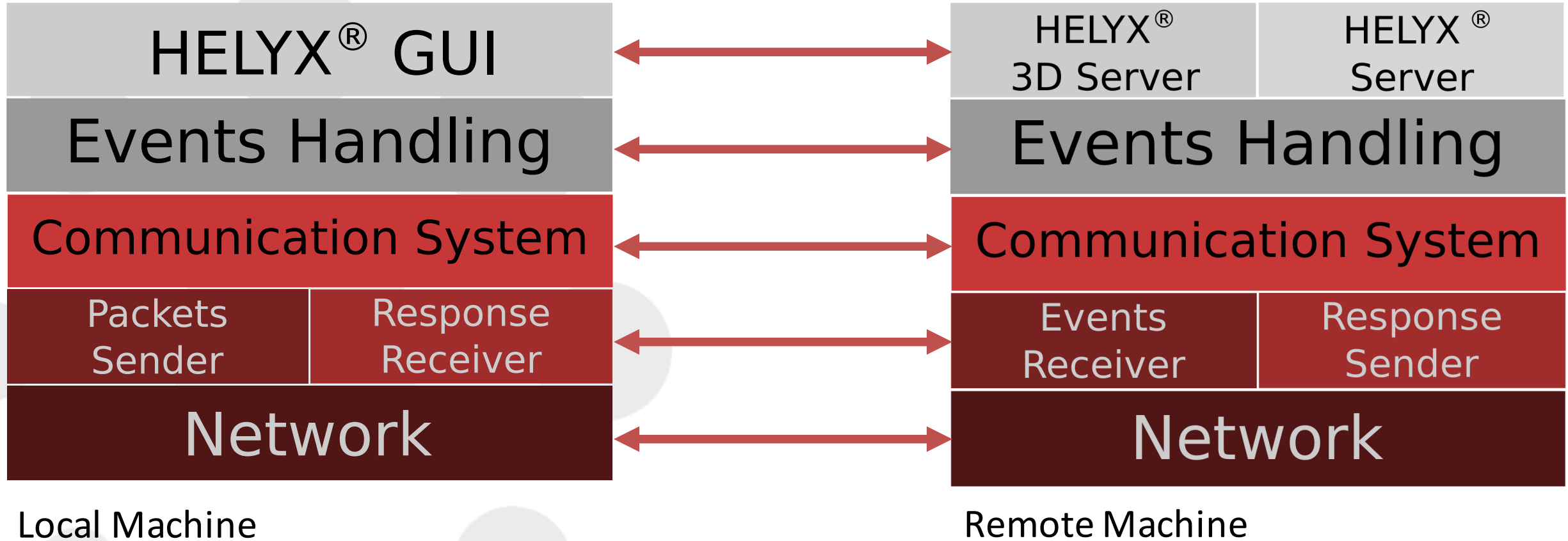
# Helyx | HELYX<sup>®</sup> Client-Server Network Architecture

## Main Features:

- Only rendered images are sent back
- Data transfer minimisation
- Security assurance → SSH tunnel

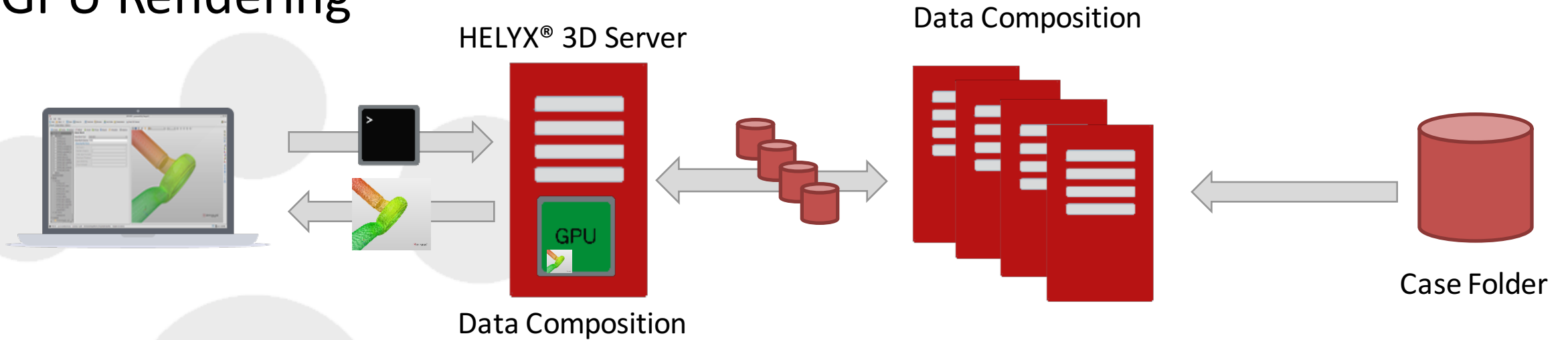


# **Helyx** | HELYX<sup>®</sup> Client-Server Network Architecture

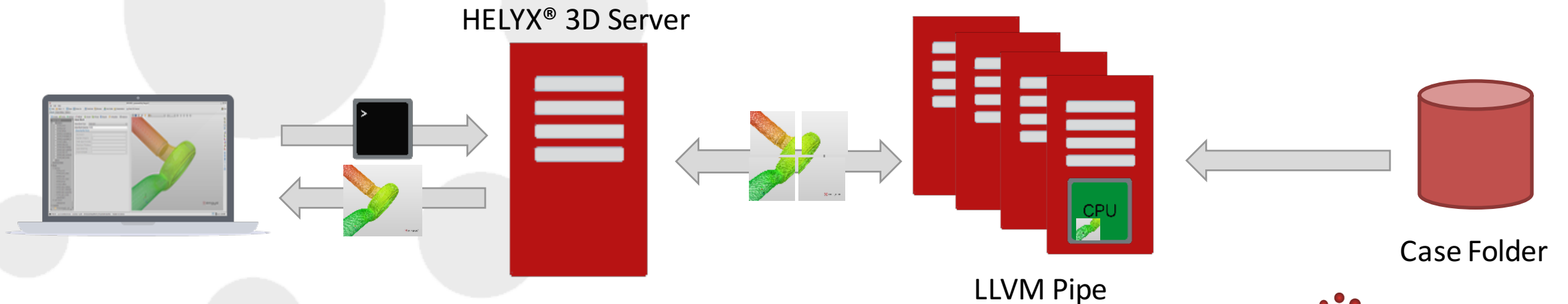


# Helyx | Application Scenarios

- GPU Rendering



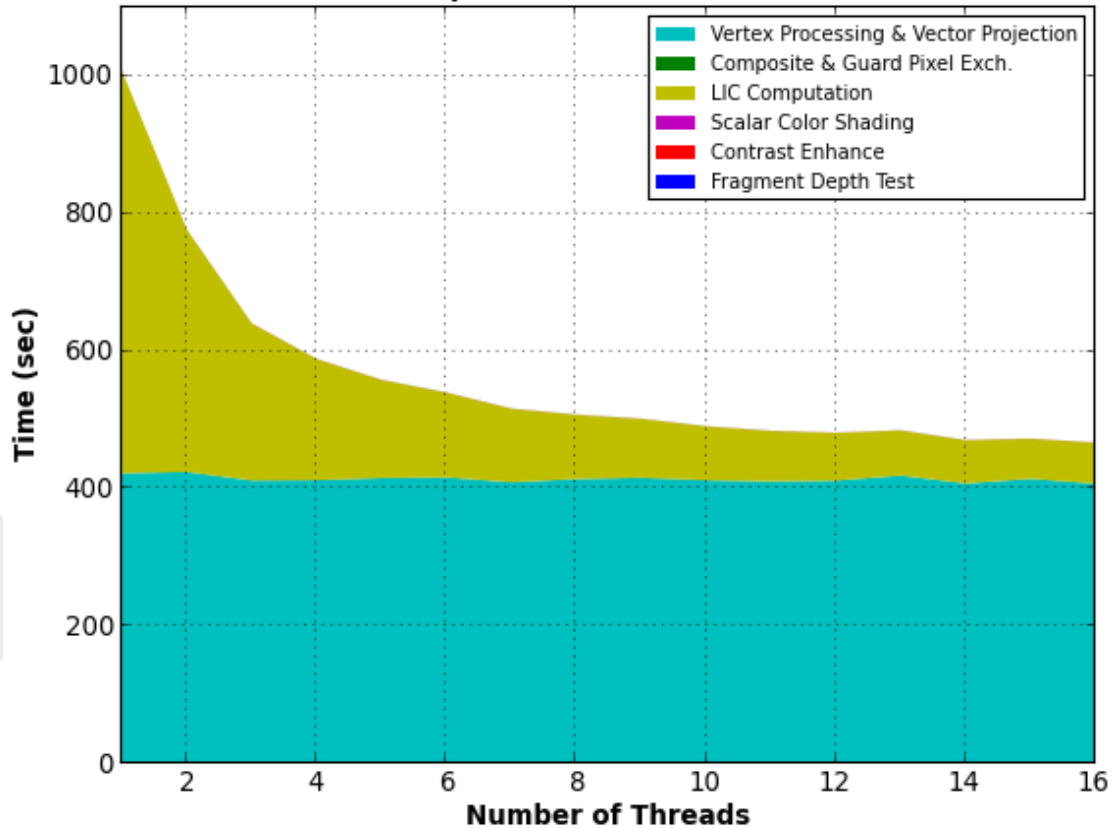
- Software Rendering



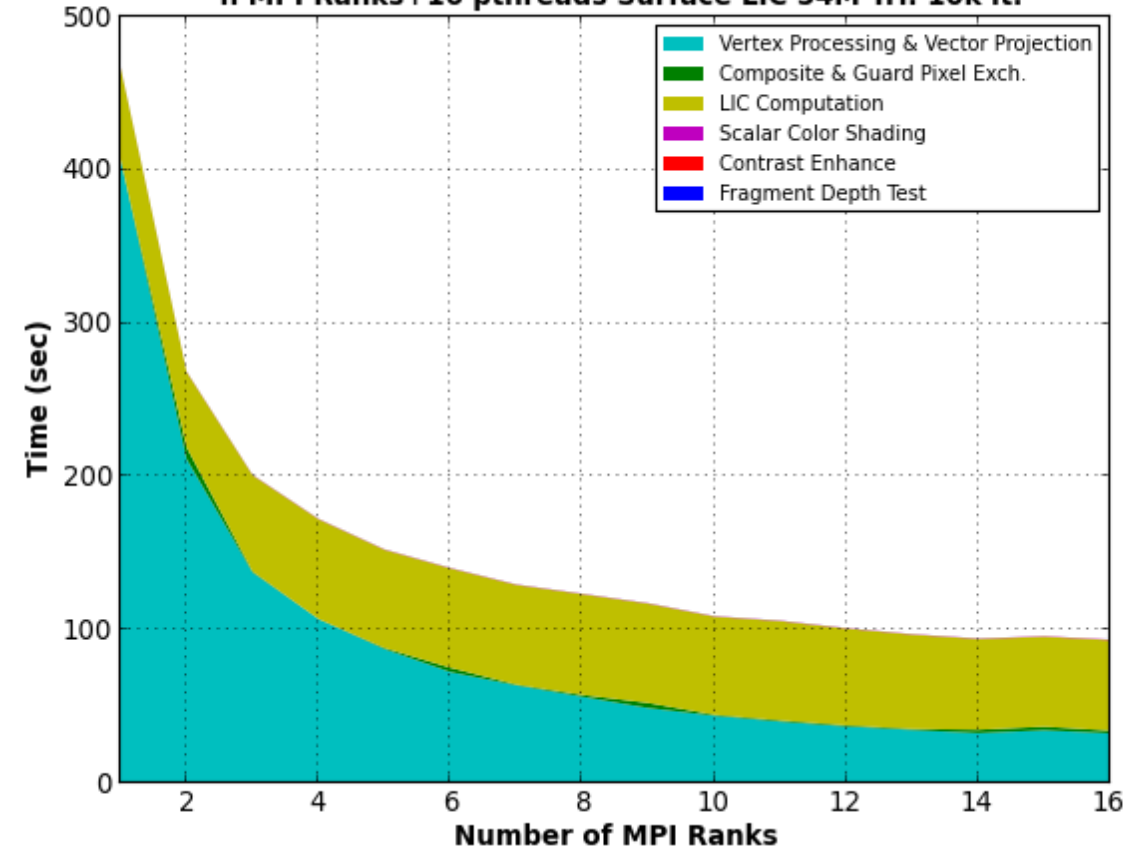


# LLVM-Pipe Algorithm Performances

OS Mesa Ilvmpipe Edison Single Node Performance  
1 MPI Rank+n pthreads Surface LIC 54M Tri. 10k it.



OS Mesa Ilvmpipe Edison Single Node Performance  
n MPI Ranks+16 pthreads Surface LIC 54M Tri. 10k it.

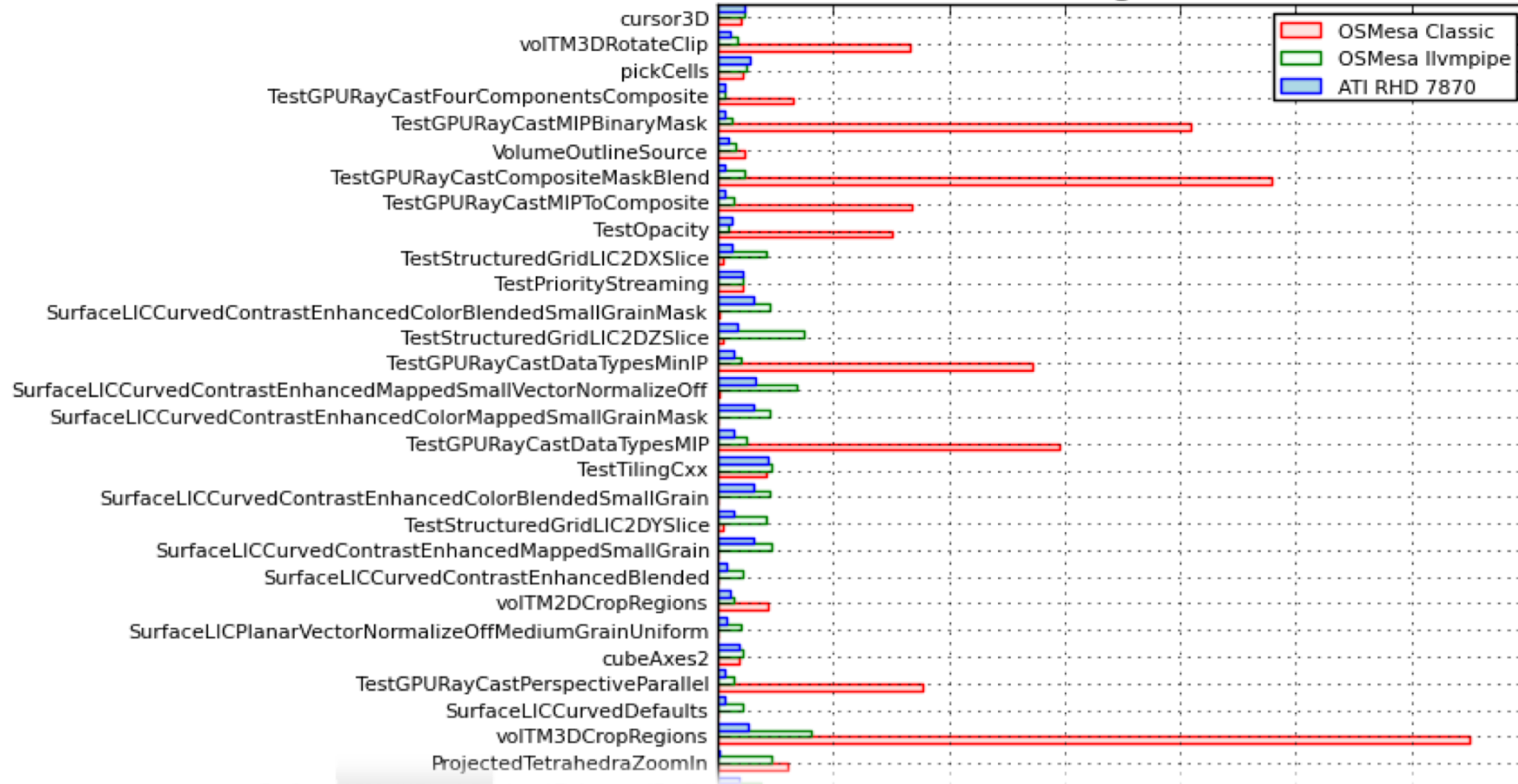


Ref: [http://www.paraview.org/Wiki/ParaView/ParaView\\_And\\_Mesa\\_3D](http://www.paraview.org/Wiki/ParaView/ParaView_And_Mesa_3D)



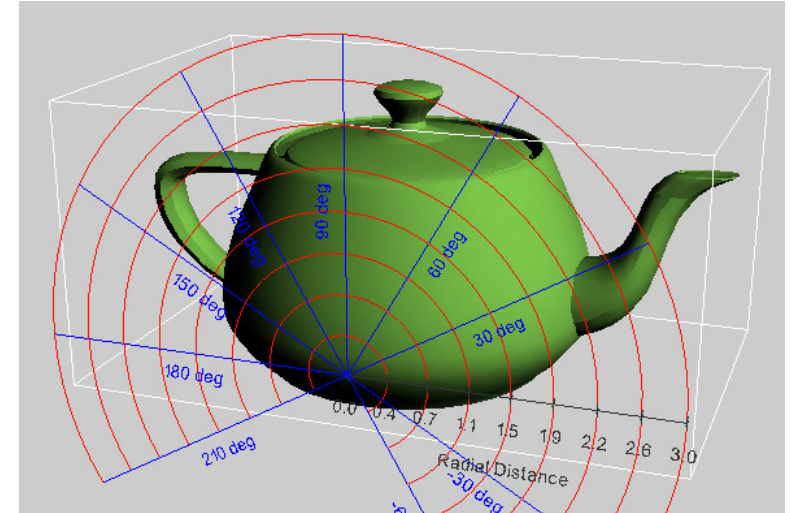
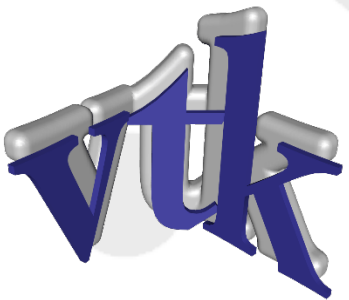
# LLVM-Pipe Algorithm Performance

### VTK Rendering Module Tests

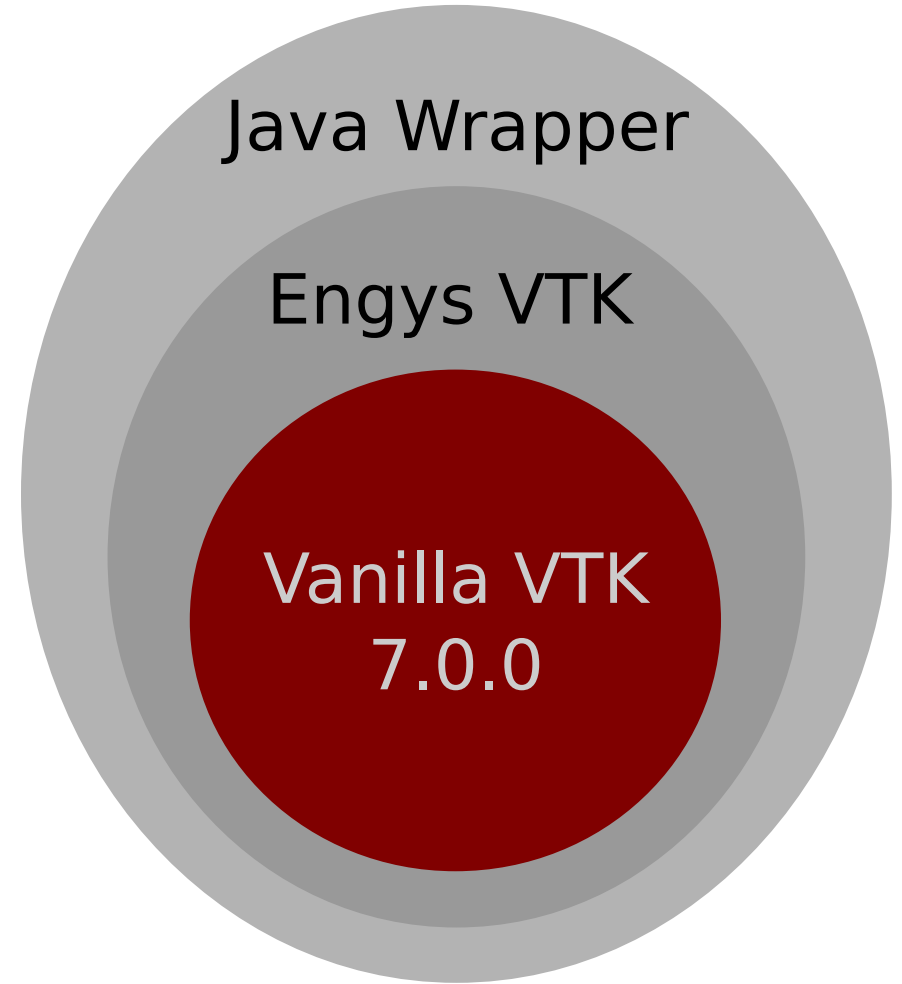


Ref: [http://www.paraview.org/Wiki/ParaView/ParaView\\_And\\_Mesa\\_3D](http://www.paraview.org/Wiki/ParaView/ParaView_And_Mesa_3D)

- New OpenGL 2 backend (+ 6M GPU primitives)
- Java wrapper support
- Rendering ~350 times faster than v6.x.x
- MPI support
- Off-screen rendering through OSMesa
- EGL support
- Composite rendering



- High complexity structures at C++ side modified by ENGYS
- Java side: high-level objects only
- New formats readers/writers
- Enhanced efficiency and performance



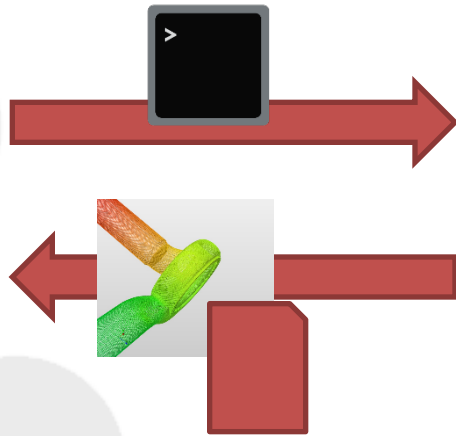
- Motivation
- HELYX<sup>®</sup> Client-Server Network Architecture
- **HELYX<sup>®</sup> HPC Usage**
- Demo
- Future Developments

- Remote execution (via SSH)
- Case data located at server side only
- Lightweight client application
- Computational complexity at server side
- Headless (no GPU) cluster support
- Queue system support (PBS, LSF, etc.)

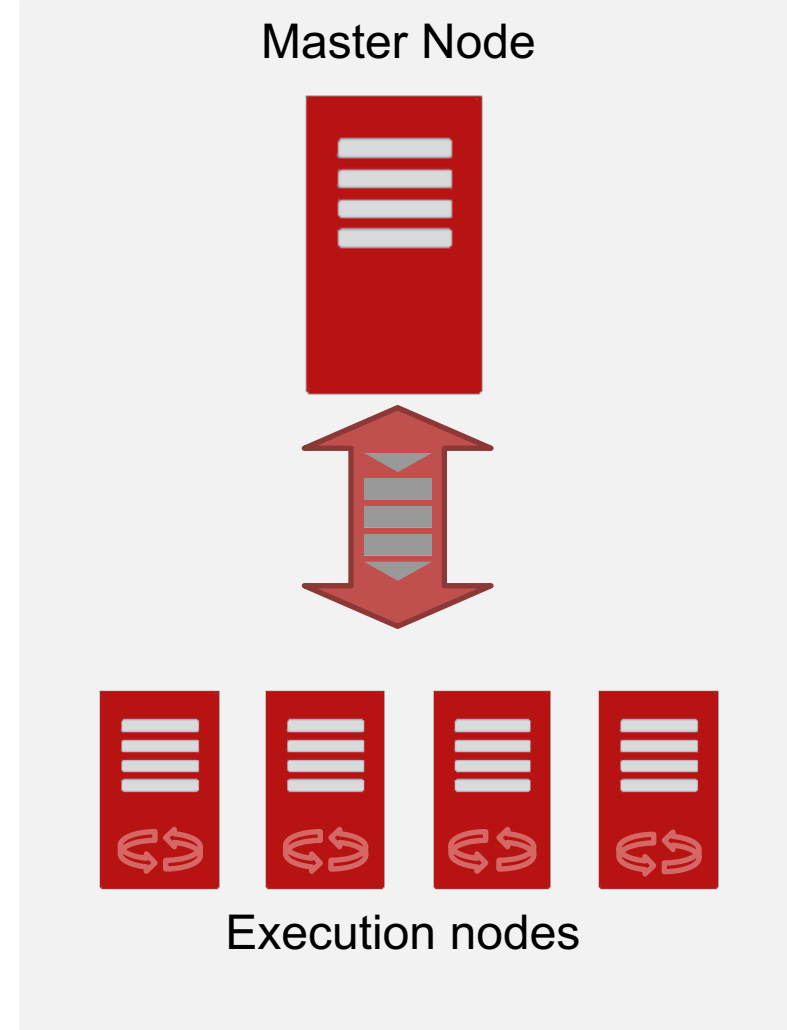
# Helyx | HELYX – HPC Usage



HELIX<sup>®</sup> Client



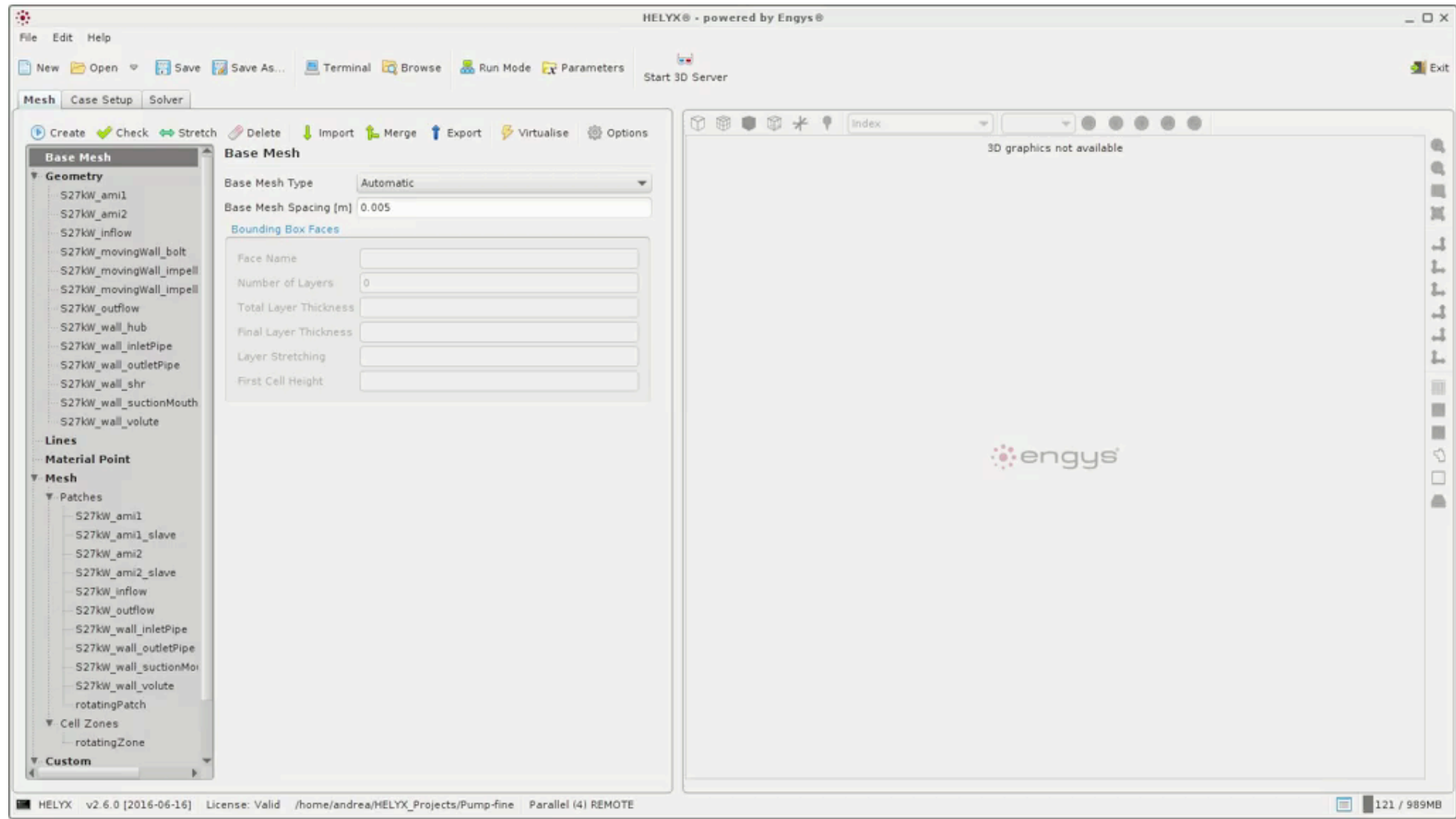
## HPC System



- Motivation
- HELYX<sup>®</sup> Client-Server Network Architecture
- HELYX<sup>®</sup> HPC Usage
- **Demo**
- Future Developments



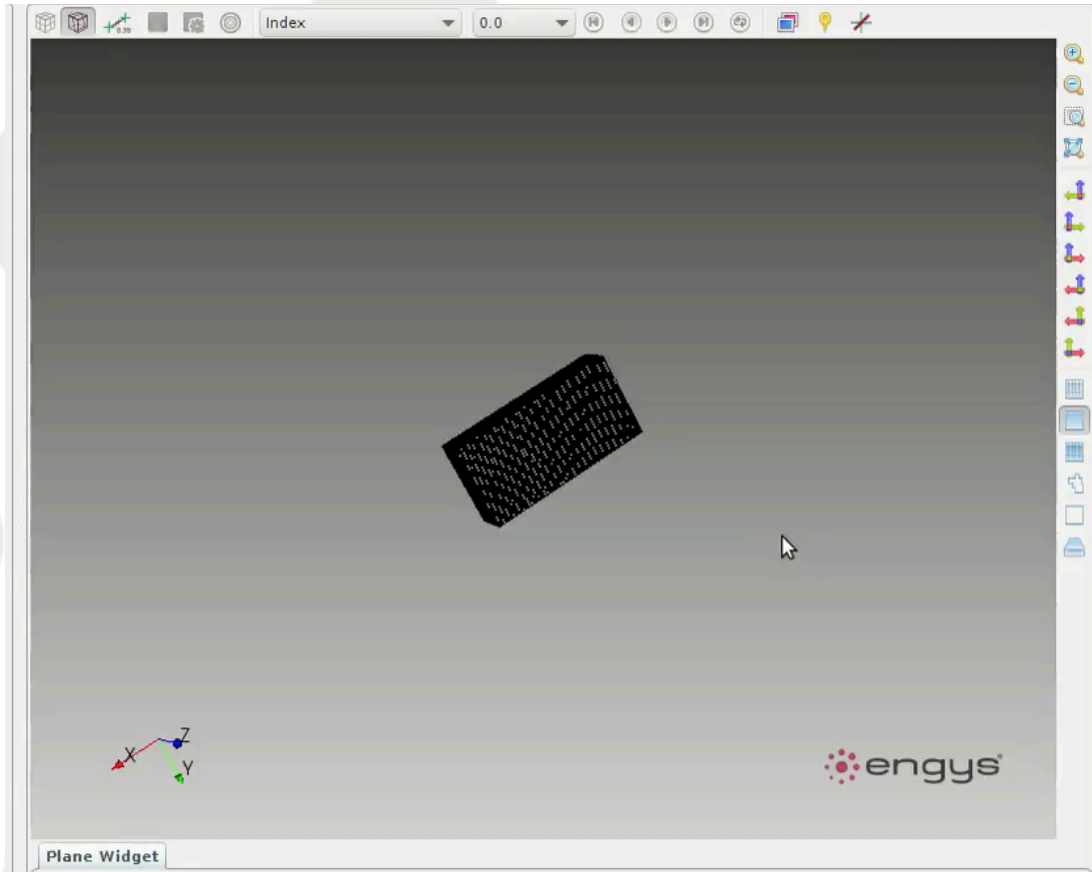
# Demo: HELYX<sup>®</sup> Remote Usage



# of cells: 1.907.054

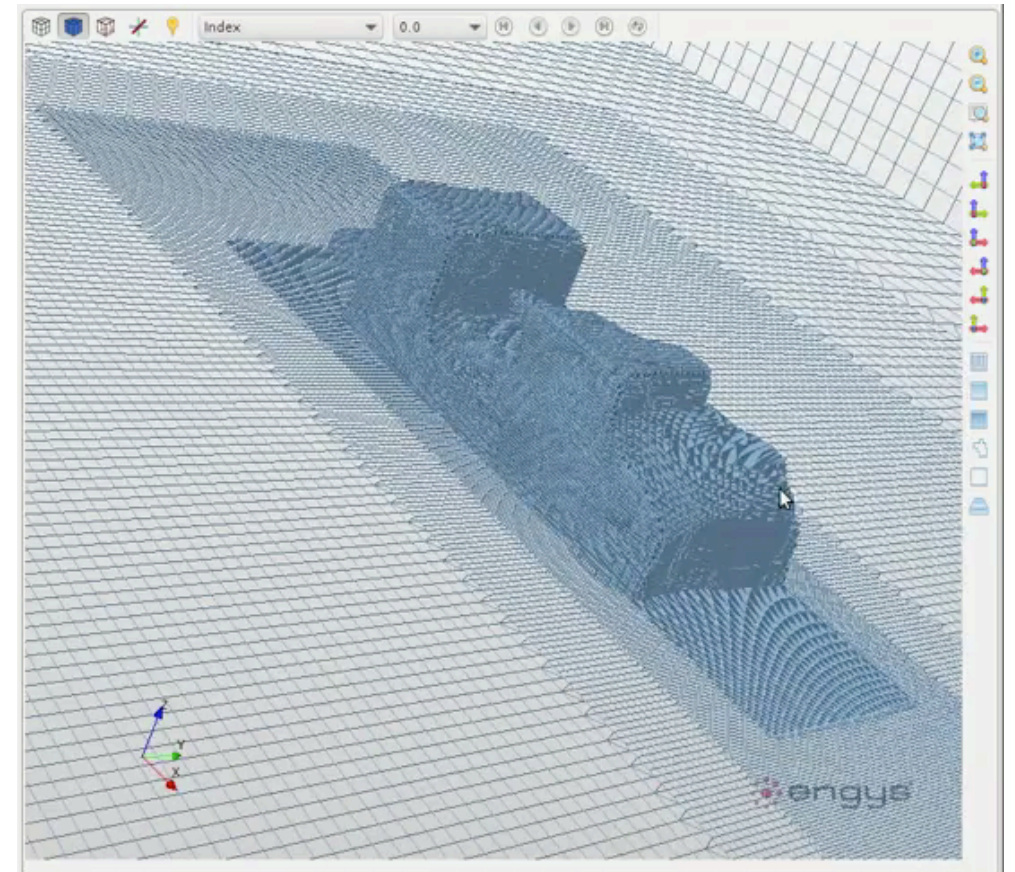
# Helyx | Demo: New Rendering Engine

VTK 6.1 Rendering Engine



Mesh size: 58.260.284 cells

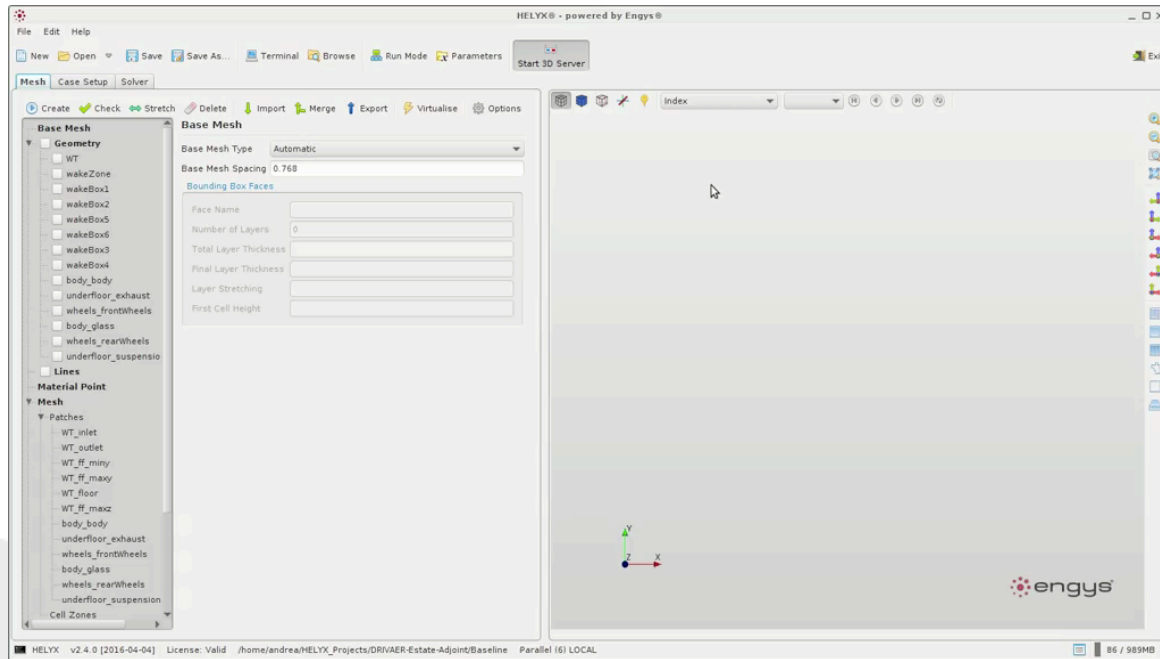
VTK 7.0.0 Rendering Engine



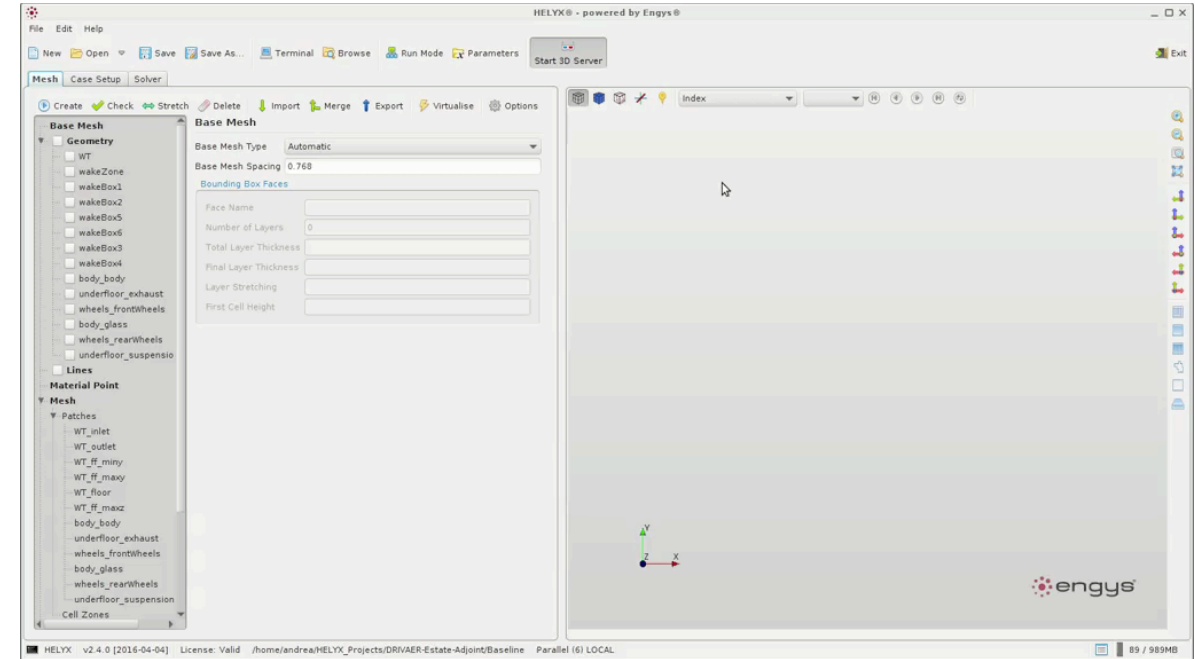


# Demo: Serial vs Parallel Reader

## Serial Reader



## Parallel Reader (Rank 6)



	Serial	Parallel
# of Processes	1	6
Mesh Size	3.881.803	3.881.803
Scenario	GPU Rendering	GPU Rendering
Time [ms] *	124.296	21.352

**~6 times faster**

(\* Inclusive of both reading and rendering time)



- Motivation
- HELYX<sup>®</sup> Client-Server Network Architecture
- HELYX<sup>®</sup> HPC Usage
- Demo
- **Future Developments**

# Helyx | Future Developments

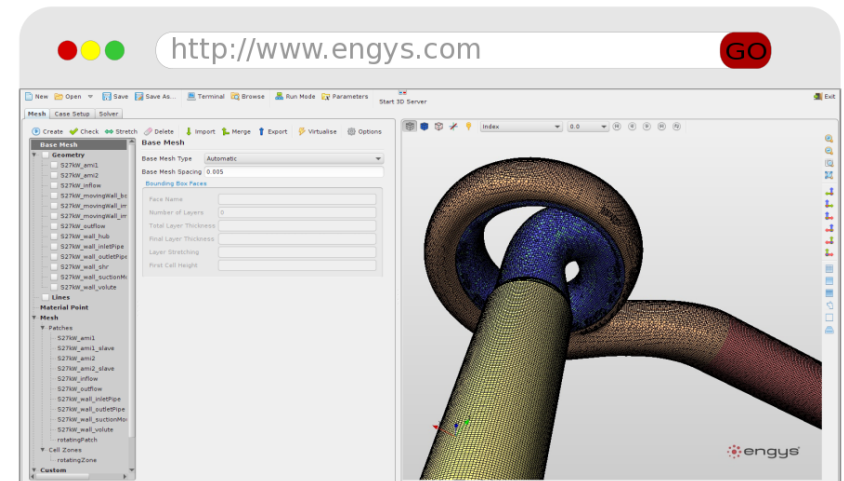
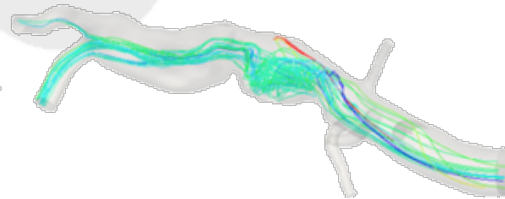
- GPU rendering in remote machines
- HELYX<sup>®</sup> server as a Web Service
  - Browser access
  - Mobile client
- In-situ visualisation



$$\rho \frac{D\vec{v}}{Dt} = -\nabla p + \mu \nabla^2 \vec{v}$$

From Paraview Wiki

Simulation Code +  
ParaView Catalyst



- OpenGL - the industrys foundation for high performance graphics, <https://www.opengl.org>
- E. Miretsky, “Software based gpu framework,” Ph.D. dissertation, 2013.
- D. DeMarle, B. Boeckel, and C. Atkins. (2016) VTK 7.0.0. <https://blog.kitware.com/vtk-7-0-0/>
- Kitware. (2016) Vtk/api changes 6.3.0 to 7.0.0.
- T. Richardson and K. R. Wood, “The rfb protocol,” *ORL, Cambridge, January, 1998.*



**Thank You Very Much!**

Any Questions?