

# Visualization Work Supporting Data Staging and Code Coupling

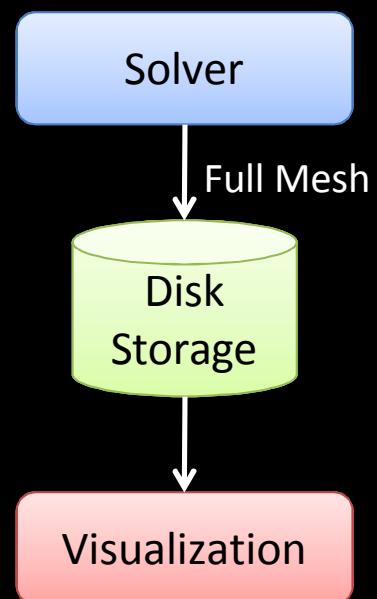
## Data Staging, In-Transit Processing, and Code Coupling in HPC Systems

Friday, July 16, 2010

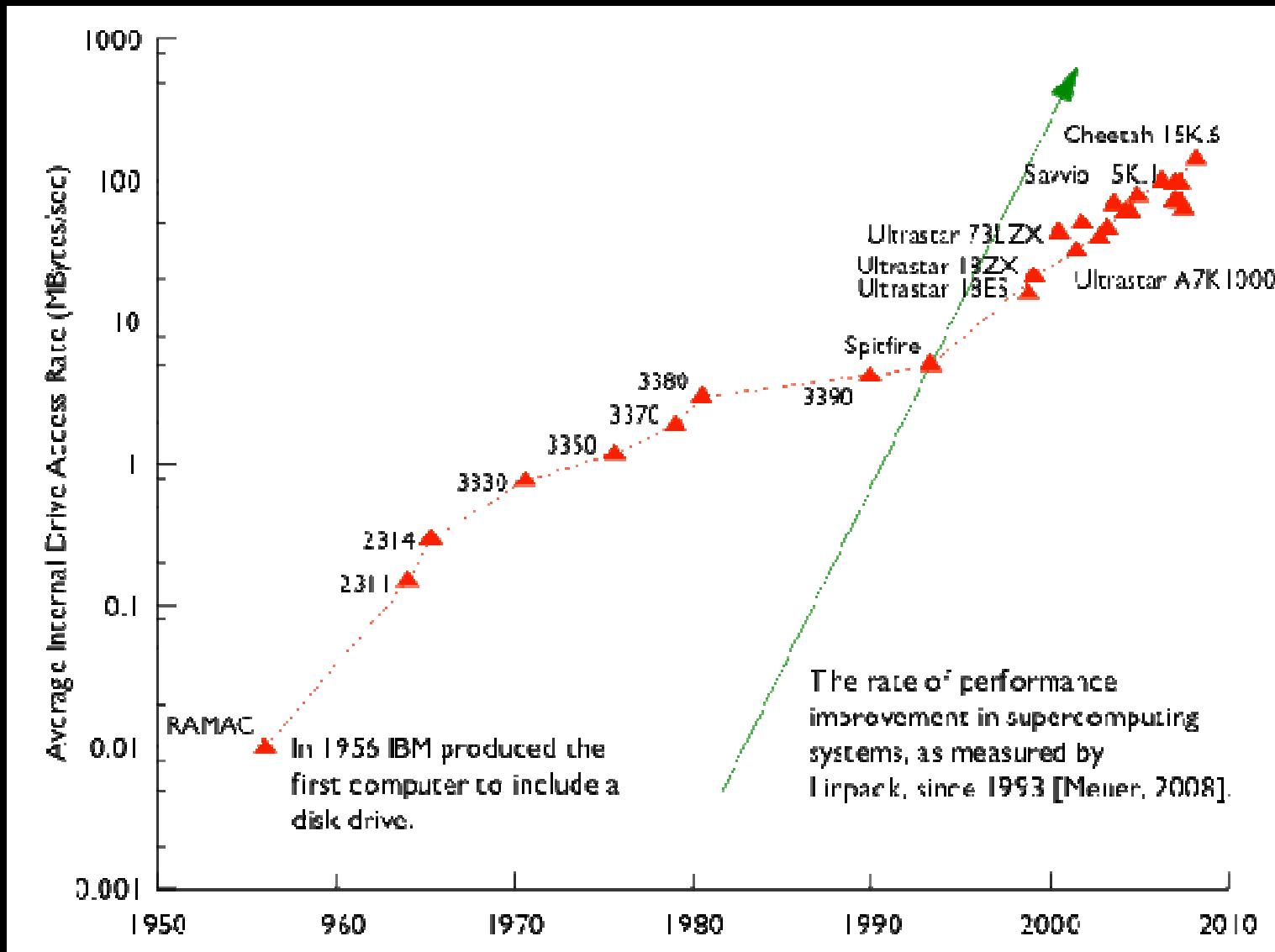
Kenneth Moreland  
Sandia National Laboratories

Release Marking (e.g. Not Approved for Release, SAND XXXX, etc.)

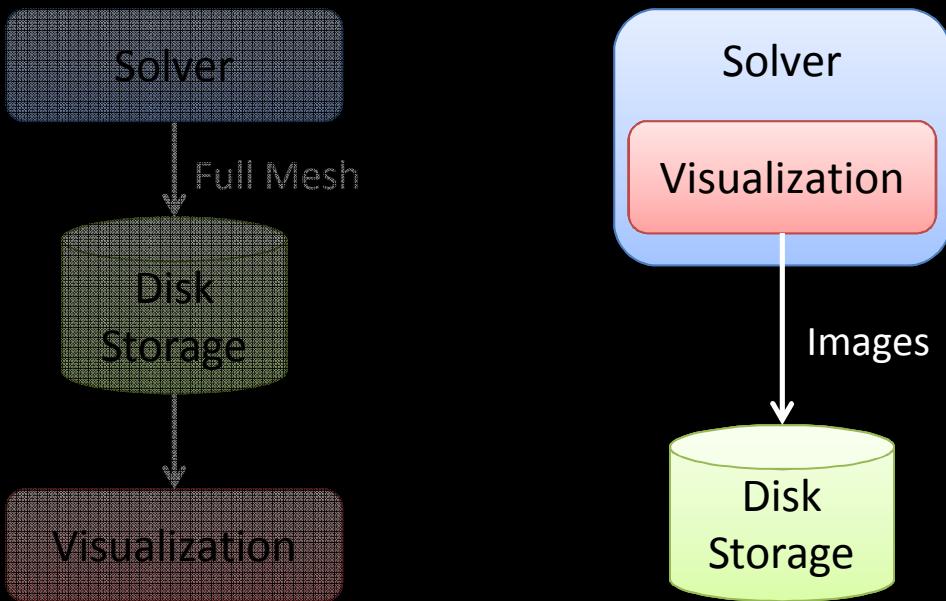
# Traditional Visualization Workflow



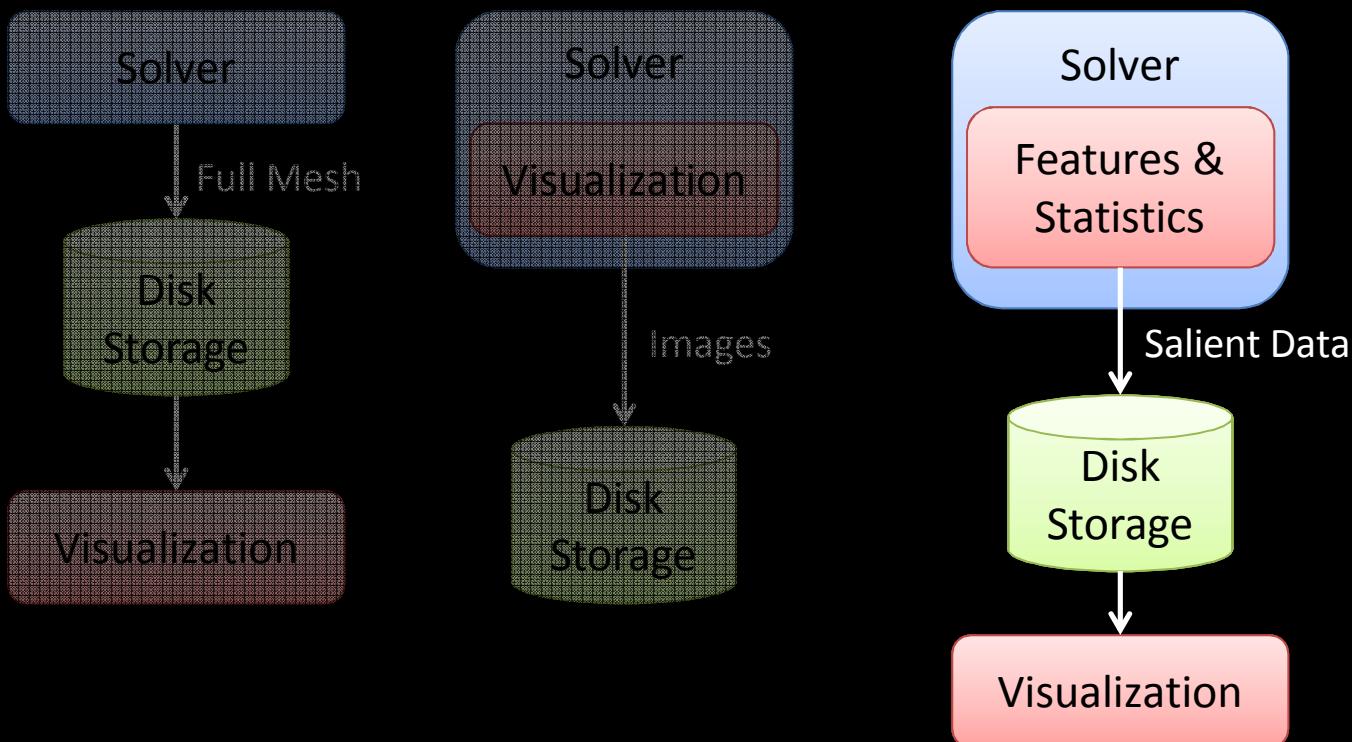
# Traditional Visualization Workflow is Breaking Down



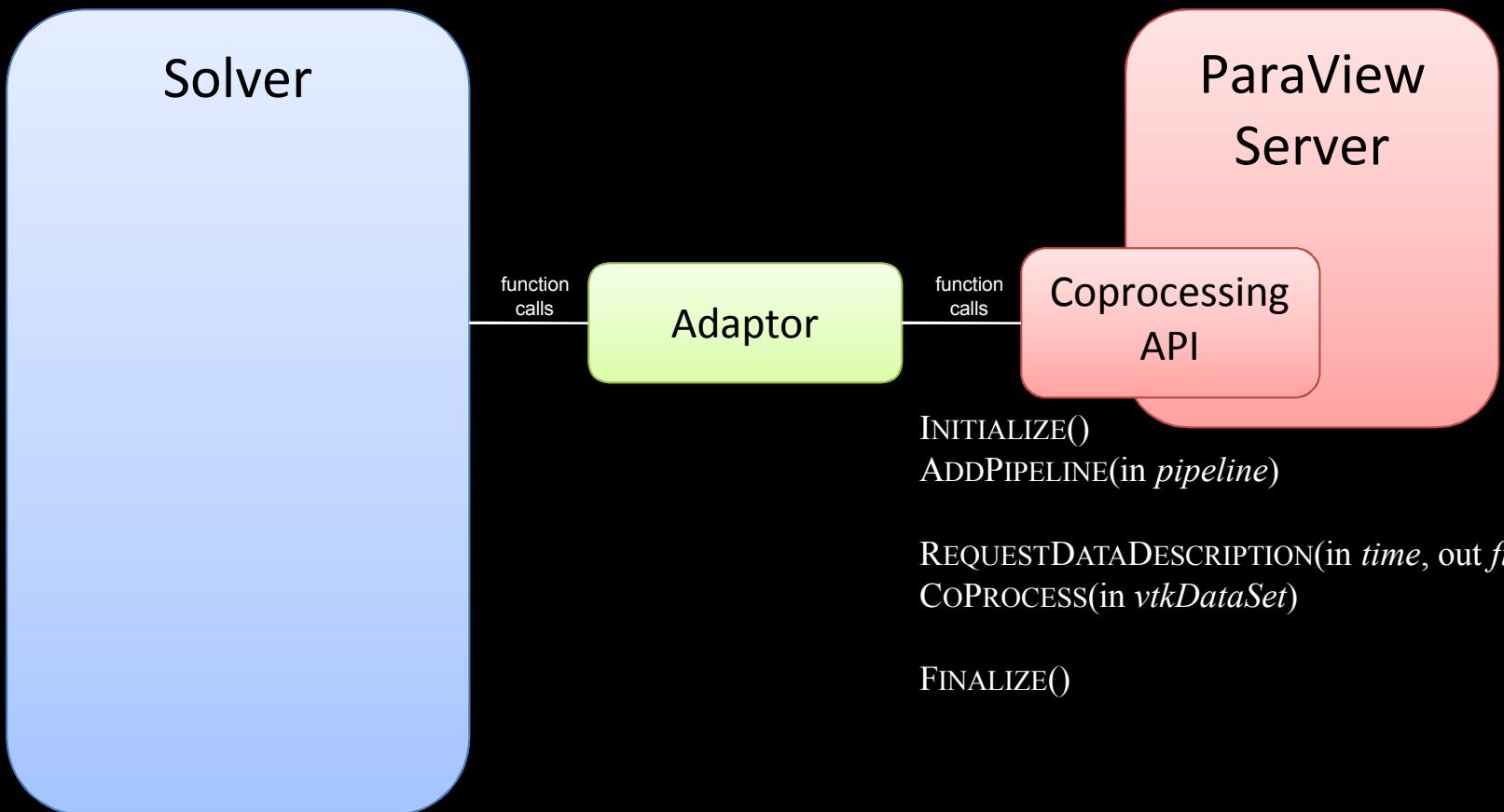
# Traditional *In-Situ* Visualization

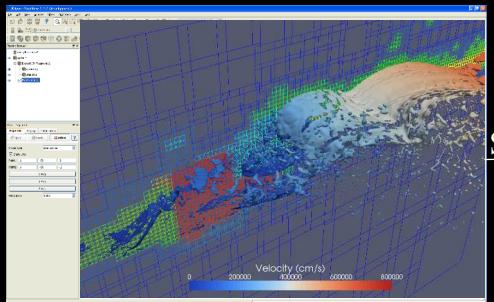


# Coprocessing



# Coprocessing Library





## Script Export

```
# Create the reader and set the filename.
reader = servermanager.sources.Reader(FileNames=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader, view)
reader.UpdatePipeline()
dataInfo = reader.GetDataTableInformation()
pDInfo = dataInfo.GetPointDataInformation()
arrayInfo = pDInfo.GetArrayInformation("displacement9")
if arrayInfo:
    # get the range for the magnitude of displacement9
    range = arrayInfo.GetComponentRange(-1)
    lut = servermanager.rendering.PVLookupTable()
    lut.RGBPoints = [range[0], 0.0, 0.0, 1.0,
                     range[1], 1.0, 0.0, 0.0]
    lut.VectorMode = "Magnitude"
    repr.LookupTable = lut
    repr.ColorArrayName = "displacement9"
    repr.ColorAttributeType = "POINT_DATA"
```

Augmented  
script in  
input deck.

# Simulation

(CTH, Presto, S3D, etc.)

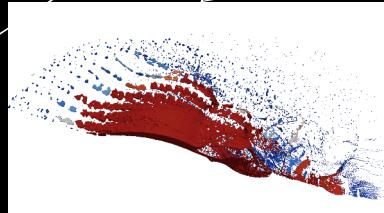
# ParaVie

W

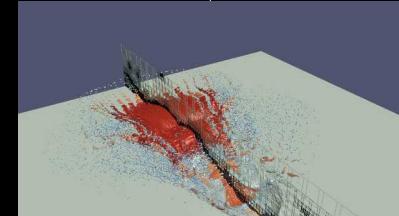
Output  
Processed  
Data

## Fragment Statistics

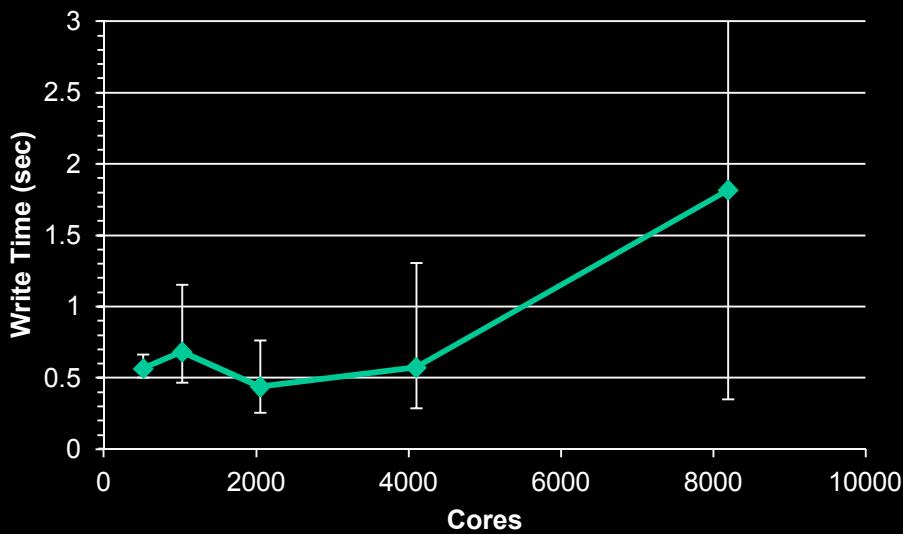
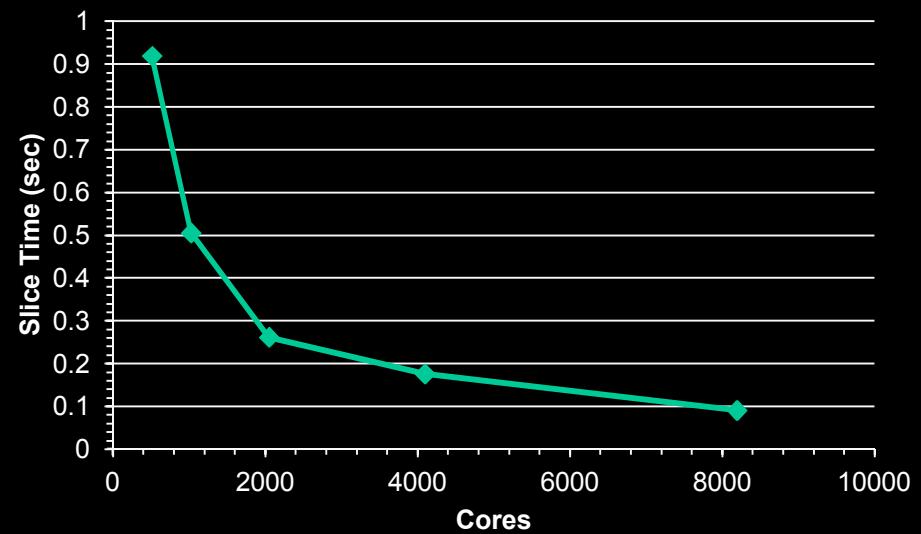
## Crater Statistics



## Polygonal Output with Field Data



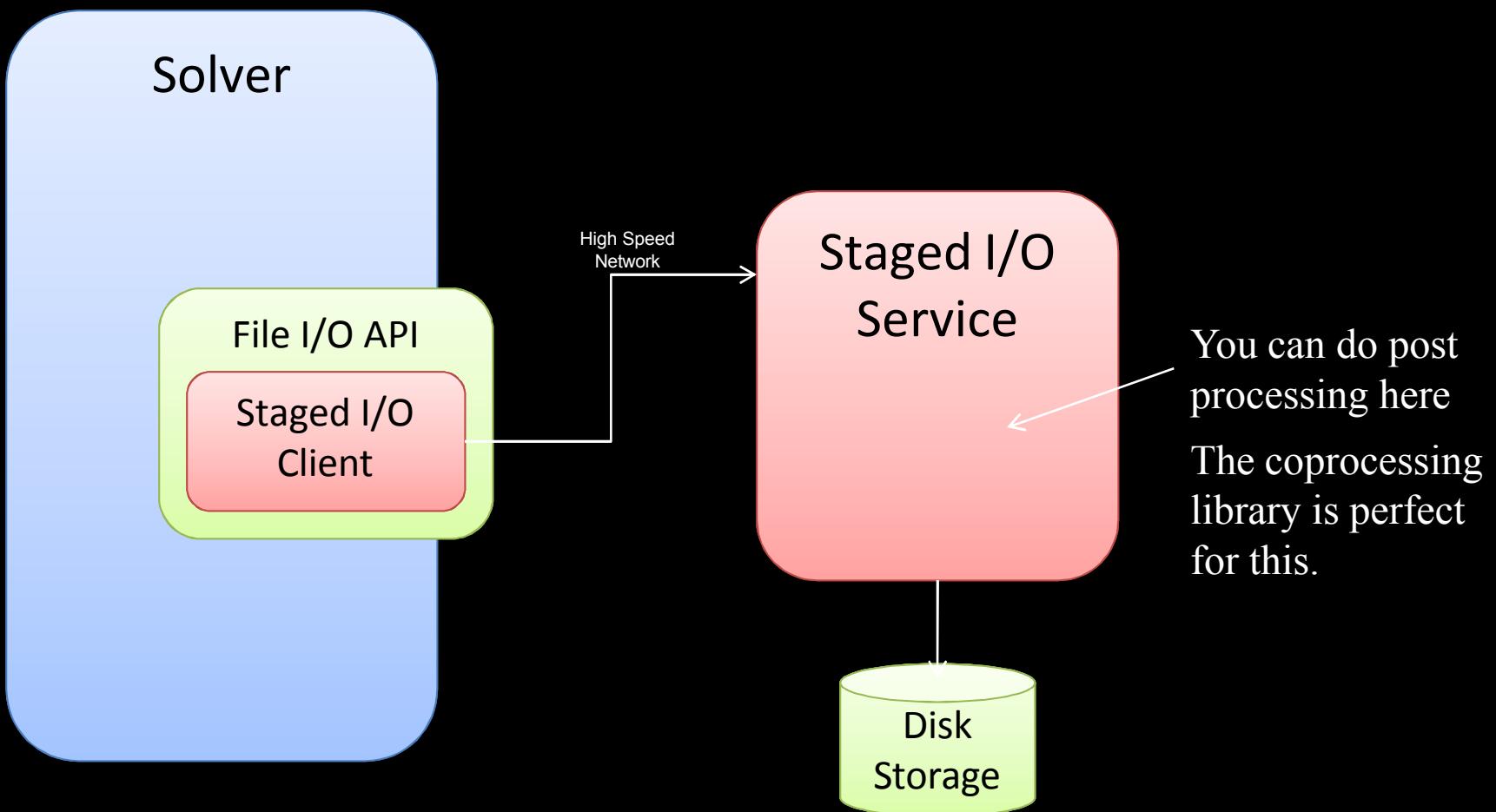
## Rendered Images



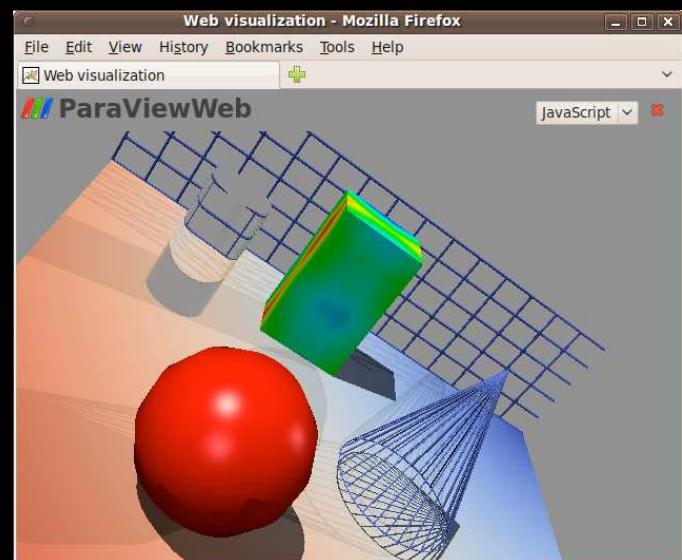
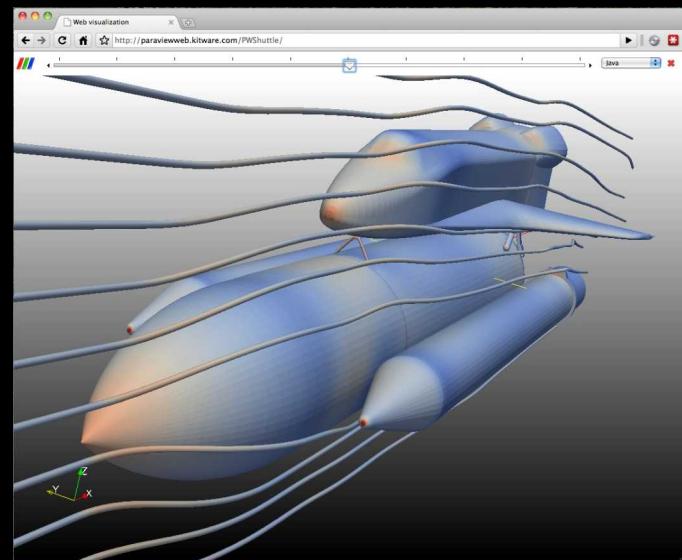
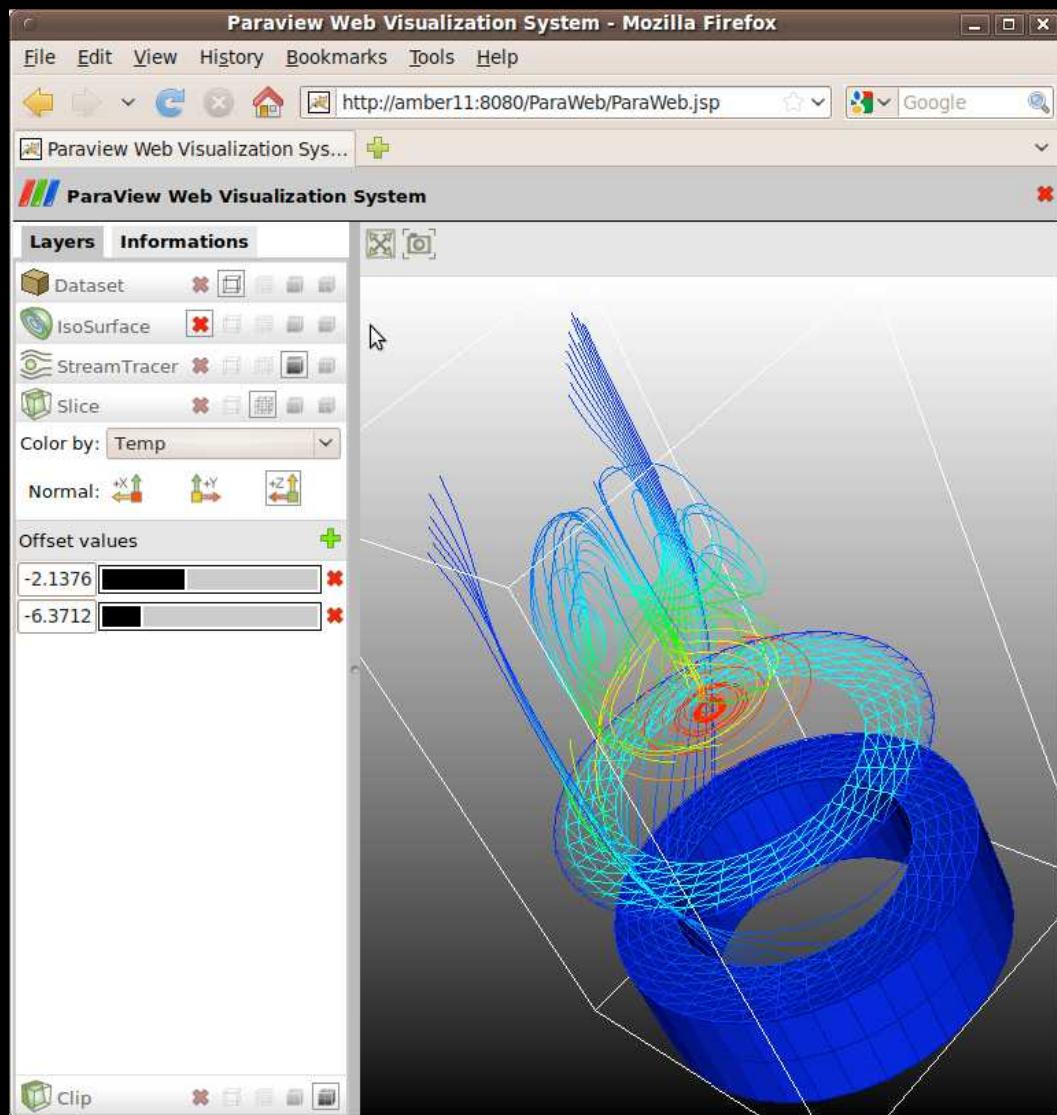
# Risks with Direct Coupling

- Running at Scale
  - Most likely some algorithms will scale, others will not.
- Bloat
  - Linker can significantly reduce code overhead *if* visualizations are known *a-priori*.
    - Python scripting is problematic for a variety of reasons.
  - Adapting memory structures is tricky.
    - Some algorithms will create their own data anyway.
- Stability (real and perceived)
  - Adding any coprocessing increases complexity, which can compromise stability.
  - What if coprocessing blamed for hero-sized crash?
  - What if coprocessing really is at fault?

# Loose Coupling via Staged I/O



# ParaView Web Services





# Titan Web Services

