

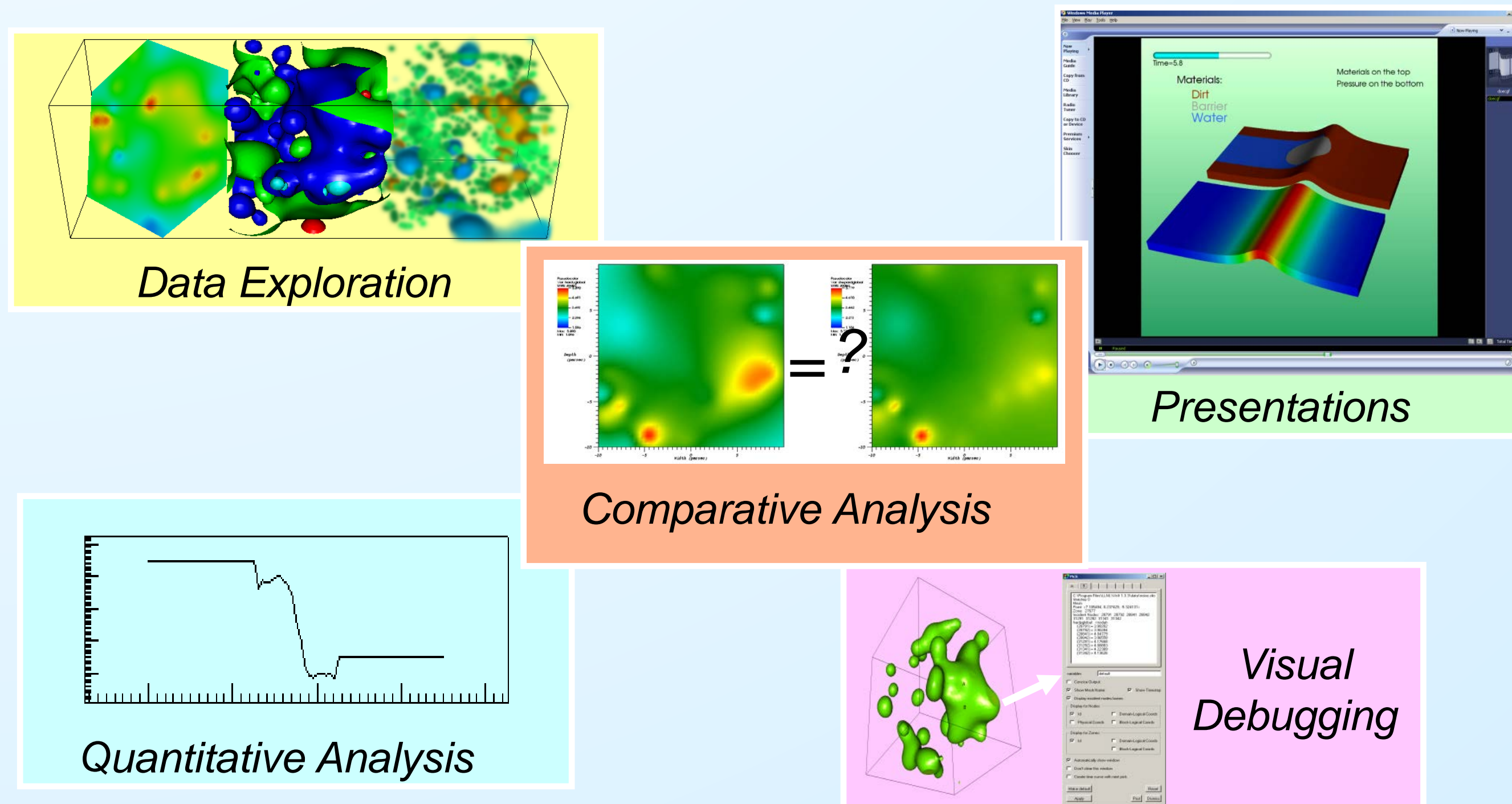
# Transforming VisIt for Multicore / Manycore Architectures

Eric Brugger, Cameron Christensen, Jeremy Meredith, Dave Pugmire, Kenneth Moreland, Berk Geveci, Christopher Sewell

## What is VisIt?

VisIt is a distributed, parallel visualization and analysis tool designed to handle big data.

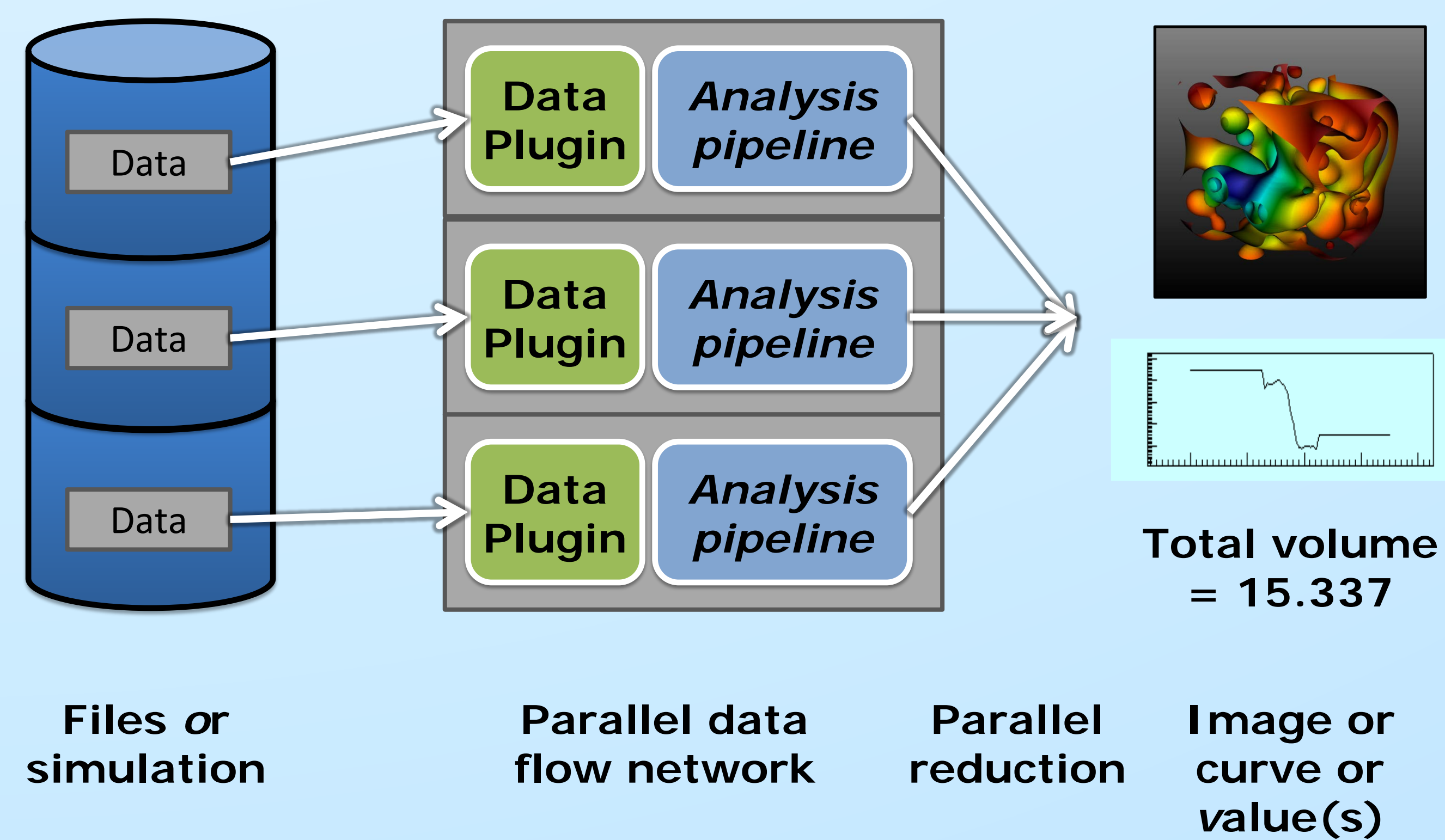
### Major use cases



### Key big data features

- Data typically post processed from parallel file system
- In situ capability targeted at simulation steering
- Client / server architecture allows remote access to data
- Run interactively or in batch via Python

### Architecture



## What is our approach?

We will change VisIt to base its processing and rendering infrastructure on VTK-m. We will take an incremental approach using the existing multicore and manycore toolkits to gain experience with such toolkits and prepare ourselves for a quick transition once VTK-m is ready.

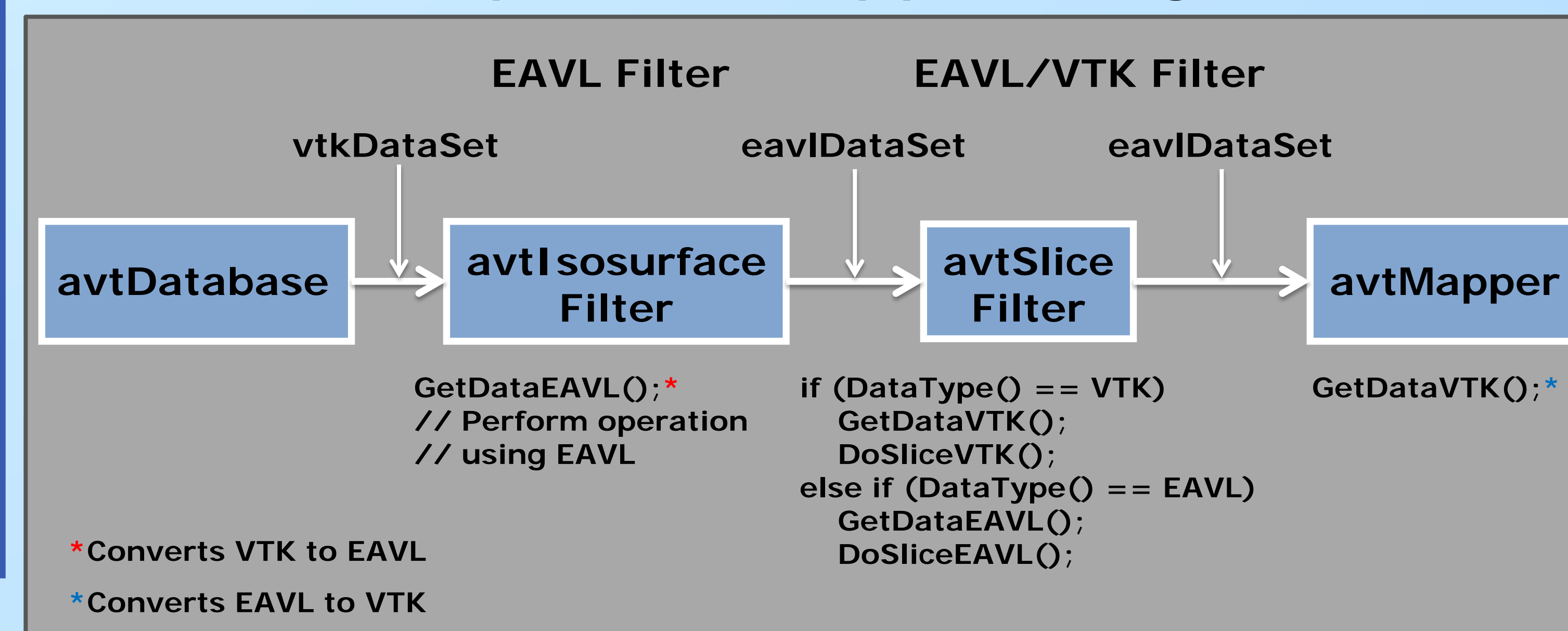
### Integration overview

- There are three portions of VisIt that will be impacted by this transition. They include the database readers, the data processing filters and the rendering. All three of these use VTK.
- We are going to start with the data processing filters
  - This will give the most performance improvement
  - We will leave our database readers alone and continue to have them return VTK data sets
  - We will convert the various toolkit data sets to VTK data sets for rendering or saving results
- We are going to modify the filter infrastructure to work with any type of visualization and analysis toolkit
- We will start prototyping with the existing toolkits
- We will then move on to prototyping the database readers and rendering
- Once VTK-m is ready we will switch to VTK-m and convert more and more of VisIt to use VTK-m until we reach a point of diminishing returns

### Software infrastructure changes

- We are enhancing our avtDataRepresentation class to handle other dataset types
- We will modify all our filters to operate on avtDataRepresentations
- We will add into avtDataRepresentation the ability to convert between VTK datasets and toolkit datasets automatically
  - These are zero-copy in most situations
- We will then start to modify the filters to use the existing toolkits

### An example mixed filter pipeline using EAVL



## What is VTK-m?

VTK-m is a new visualization toolkit combining the strengths of the existing toolkits Dax, EAVL and PISTON.



### VTK-m properties

- Targets multi and many core architectures
- Reduces the complexity of writing highly concurrent code
- Has a device independent layer that lets it run on a variety of multicore and manycore architectures
- Decompose algorithms into sections that can run a small section of data
  - The approach is essentially the same as presented by Baker and colleagues, functional mapping [Baker, et al. 2010]
- New powerful, flexible data structures
  - Can represent many structure types but with consistent access that works on all devices

### VTK-m components

