VTK Vanguard

What’s new in the trusty old Visualization Toolkit
<table>
<thead>
<tr>
<th>Year</th>
<th>Revisions</th>
<th>Emails</th>
<th>Committers</th>
<th>KW Downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>2.4.0</td>
<td>1429</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>3.1.0-.2</td>
<td>2807</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>3.2.0</td>
<td>3588</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>5222</td>
<td></td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>4.0.2, 4.2.0-.5</td>
<td>5164</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>4.2.6, 4.4.0-2</td>
<td>5670</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>5.0.0</td>
<td>4776</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>5.0.1-.2</td>
<td>4689</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>5.0.3</td>
<td>4052</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>5.0.4, 5.2.0-.1</td>
<td>4797</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>5.4.0-.2</td>
<td>7067</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>5.6.0-.1</td>
<td>10825</td>
<td>91</td>
<td>3969</td>
</tr>
<tr>
<td>2011</td>
<td>5.8.0</td>
<td>7982</td>
<td>71</td>
<td>4712</td>
</tr>
<tr>
<td>2012</td>
<td>5.10.0-.1</td>
<td>7683</td>
<td>57</td>
<td>7243</td>
</tr>
<tr>
<td>2013</td>
<td>6.0.0</td>
<td>6730</td>
<td>68</td>
<td>10707</td>
</tr>
<tr>
<td>2014</td>
<td>ao Apr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>6.1.0, ?6.2.0?</td>
<td>1564</td>
<td>36</td>
<td>1065</td>
</tr>
</tbody>
</table>
6.0.0 Jun 2013

- Pipeline updates
- Modularization
- Mean Value Coordinates mesh interpolation
- vtkPiston
- etc: Categorical Colors, Vector Graphics outputs
Pipeline Updates

- Removed VTK 4 backward compatibility
  - `vtkDataObject::GetProducerPort()`?
  - `vtkAlgorithm::SetInput()` vs `SetInputConnection()`

![Diagram showing the difference between VTK 5 and VTK 6](image-url)
Modularization

Data

VTK Minimal
Modularization

- Reorganize and modernize cmake script
- 19 kits -> 107 modules
- Can remove just by deleting directories (100MB->5MB)
- Easier to add external Modules
- Code behavior is constant
  - ex, VTK_USE_PARALLEL doesn’t make any class compile or behave differently, it merely enables additional classes
DeformPointSet

• Mean Value Coordinates mesh warping
vtkPiston & ParaView Piston Plugin
vtkDataSetToPiston

vtkPistonData

void * PISTONHANDLE

vtkPistonSort

vtkPistonData

void * PISTONHANDLE

vtkPistonContour

vtkPistonData

void * PISTONHANDLE

vtkPistonToDataSet

vtkPolyData

Piston::sort

ImageData

Piston::isocontour

Polys

Piston::toPolyData

To GPU

To CPU
6.1.0 Jan 2014

- vtkDax, vtkSMP
- vtkWeb
- InSitu Data Structures (zero copy arrays)
- vtkPython binaries

- etc: ext netcdf, qt5, python binaries, JOGL, testing infrastructure
vtkSMP - threaded unstructured filters

- [http://www.vtk.org/Wiki/VTK/VTK_SMP](http://www.vtk.org/Wiki/VTK/VTK_SMP)
- A lightweight threading abstraction
  - various back ends including Intel’s TBB and Inria’s Kaapi
- Minimal building blocks above abstraction layer
  1. parallel for
  2. thread local storage
  3. atomic operations, especially fast atomic ints
  4. efficient parallel point merge operator
- See: Common/Core/*SMP*,
  - Filters/SMP/[Transform|Contour]
- WARNING: Does not mean that all of VTK is thread safe!
6.1 Ettinger et al 2013
Figure 6.1: Comparison of speedup for Kaapi, OpenMP, and TBB with respect to the number of cores. The dashed line represents linear speedup.
vtkWeb - visualization over the web

- Core ParaviewWeb Version 3* promoted to VTK
- Dependencies:
  - before: activemq-cpp, apr, apr-util, tomcat
  - now: zope, py, autobahn.py, twisted.py all in VTK3rdParty

1. VTK_WRAP_PYTHON, VTK_Group_Web=ON
2. Compile
3. vtkpython Wrapping/Python/vtk/web/vtk_web_cone.py \ --content www  -p 11111

* has not burned, fallen over and sunk into the swamp yet
Where does it work?

- Any modern browser that supports WebSocket
- Safari on iOS (iPhone, iPad, iPod)
- Chrome on some Android devices
vtkWeb demo
In Situ (zero copy) data structures

- Problem - conform to VTK array layout or die!
  ex. $x^1y^1z^1x^2y^2z^2x^3y^3z^3...$ vs $x^1x^2x^3...y^1y^2y^3...z^1z^2z^3...$

- API gives direct access to implementation
  - `vtkDataArray::GetVoidPointer()`
  - `vtkUnstructuredGrid::GetCells()` etc..

- Solution: template meta programming to the rescue (with STL like iterator access)
- [http://www.vtk.org/Wiki/VTK/InSituDataStructures](http://www.vtk.org/Wiki/VTK/InSituDataStructures)
Mapped Arrays

- Derive from vtkMappedDataArray
  - Fill in virtual data access methods for your data layout
  - Will spit out warnings wherever RawPointerAccess happens, and default to returning pointer to a manually extracted copy
- incompatible filters take slow path
- they will be updated as needed
- iterator access pattern makes updating straightforward
Mapped Unstructured Grids

- CellConnectivity

- you write impl that fulfills abstract API

- iterators (ex vtkCellIterator) make it fairly easy to update

- See TestInsituExodus to get started

- https://github.com/gahansen/Albany/tree/master/src/disc/catalyst
6.2.0 ~Sept 2014 to include some of...

- Ghost cell generation
  - George Zagaris for European customers
- XDMF3
  - Dave DeMarle with Ken Leiter at ARL
- Adios++
  - Chuck Atkins with ?
- Sane structured/unstructured dichotomy
  - Berk Geveci for InSitu SBIR
- Binary SDK, Java Maven and Python PIP
  - DeMarle/Jourdain for NIH and others
6.3.0 ~Feb 2015 to include some of...

- vtk2vtkm?
- NIH VTK Maintenance Grant
  - update rendering/interaction
    - Chaudary/Avila/Hanwell/Martin et al
    - SceneGraph
    - OpenGL >=2.3, VBOs, improved context creation/sharing
    - Reduced memory footprint
    - Preliminary tests: 14 million pt Lucy.ply
      - 4.02s/f -> .16s/f, CPU 100%->5%, RAM -3GB
  - Infrastructure to encourage community code

![Diagram of graphics model](image-url)