Date: 1/4/2019

To: James Ahrens

cc: David Pugmire, Berk Geveci, Robert Maynard

From: Kenneth Moreland, WBS 2.3.4.13 / ECP/VTK-m

Milestone Deliverable – STDA05-16

Milestone Due Date: September 30, 2018

Milestone Completion Date: December 27, 2018

Description of Milestone:
The STDA05-16 milestone comprises the following 3 distinct deliverables.

OpenMP VTK-m currently supports three types of devices: serial CPU, TBB, and CUDA. To run algorithms on multicore CPU-type devices (such as Xeon and Xeon Phi), TBB is required. However, there are known issues with integrating a software product using TBB with another one using OpenMP. Therefore, we will add an OpenMP device to the VTK-m software. When engaged, this device will run parallel algorithms using OpenMP directives. This will mesh more nicely with other code also using OpenMP.

Rendering Topological Entities VTK-m currently supports surface rendering by tessellation of data structures, and rendering the resulting triangles. We will extend current functionality to include face, edge, and point rendering.

Better Dynamic Types Impl For the best efficiency across all platforms, VTK-m algorithms use static typing with C++ templates. However, many libraries like VTK, ParaView, and VisIt use dynamic types with virtual functions because data types often cannot be determined at compile time. We have an interface in VTK-m to merge these two typing mechanisms by generating all possible combinations of static types when faced with a dynamic type. Although this mechanism works, it generates very large executables and takes a very long time to compile. As we move forward, it is clear that these problems will get worse and become infeasible at exascale. We will rectify the problem by introducing some level of virtual methods, which require only a single code path, within VTK-m algorithms. This first milestone produces a design document to propose an approach to the new system.

Completion Proof of the Milestone:
The predefined objective completion criteria of this milestone for all the deliverables were the following items:

- Implementation is merged to the master branch of the central VTK-m repository.
- The VTK-m User's Guide is updated to the new behavior.

The following table provides evidence for each implemented feature with links to the completed merge requests (evidence that the implementation is merged into the master branch) and a link to the excerpt from the VTK-m User's Guide documenting the feature.

Deliverable	Merge Requests	Documentation
OpenMP	• Finished development of the OpenMP backend (!1099) • Relaxed constraints to allow for OpenMP 4 and OpenMP 4.5 (!1256) • Allowed for compilation of CUDA and OpenMP backend into the same library (!1271)	https://jira.exascaleproject.org/secure/attachment/14263/OpenMP%20from%20VTKmUsersGuide.pdf
Rendering Topological Entities	• Added rendering of topological entities in vtk-m supported mesh types. (!1407)	https://jira.exascaleproject.org/secure/attachment/14421/Mapper%20Pages%20from%20VTKmUsersGuide.pdf
Better Dynamic Types Impl	• Build some virtual object handle into vtkm::cont (!1415) • Unified Type and allowing for simpler VirtualObjectHandle interface [!1334] • Extend `TryExecute` with `TryExecuteOnDevice` to allow runtime selection of accelerator device [!1356] • vtkm::cont::VirtualObjectHandle can now transfer to a device with a runtime `DeviceAdapterId` value. Previously it only worked with the templated device adapter tag. (!1316) • Resolve "ArrayHandleVirtual + ArrayHandleCompositeVector cause CUDA to run out of stack memory" (!1224) • Re-implement DeviceAdapterRuntimeDetector to avoid ODR violations. (!1206) • Solved Power Architectures issues (!1244) • Make CoordinateSystem work consistently on CUDA (!1108)	https://jira.exascaleproject.org/secure/attachment/15447/ArrayHandleVirtual%20Pages%20from%20VTKmUsersGuide.pdf https://jira.exascaleproject.org/secure/attachment/15435/Variant%20Array%20Handles%20Pages%20from%20VTKmUsersGuide.pdf

- Transfer of virtuals to the CUDA device now properly uses streams (!1114)
- Virtual Coordinates (!1021)
- Refactor arrayhandle to reduce lib size (!1084)
- Improve serialization performance of ArrayHandle with BasicStorage (!1474)
- Adding ArrayHandleVirtual, and VariantArrayHandle, Removing DynamicArrayHandle (!1454)

Tasks to Complete the Milestone:
In the case for each of the deliverables, implementation started in a private topic branch. That branch was later submitted as a merge request where the code was run through regression tests across multiple test platforms. The merge requests were also subjected to human reviewers for approval. After necessary modifications were made, the code was merged to VTK-m's master branch. Subsequently, documentation was written for the VTK-m User's Guide.

Person(s) Responsible for Completing the Milestone:
David Pugmire, Berk Geveci, Robert Maynard

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2018-XXXX R