1.1.1.1 ECP/VTK-m (WBS 2.3.4.13)

Short Name: Overview

High level overview / goals / objectives (.75 page) (text) - done

- A couple of paragraphs to a page of technical description
- Table with 'scope and intent', R&D themes, delivery process, Target Users (update from the PDR)

VTK-m: Project Overview Table

Scope & Intent	R&D Themes	Delivery Process	Target ECP Users
Consolidated update of DOE HPC sci vis software to shared memory parallelism.	Managing parallel computation over highly connected topologies without duplicated halo regions.	Open Source development (GitLab); Continuous testing for code stability; Regular Software Releases; Releases available through Spack	Primary delivery through sci vis apps (VisIt, ParaView) and in situ libraries (Ascent, Libsim, Catalyst) with tight integration with the ALPINE project.

VTK-m: Software Products

L4 software product descriptions (table of code name, purpose/capability, support model, dependencies) (.5 page) (table)

Code Name	Purpose/Capability	Dependencies	Support Model
VTK-m	Provides a collection of scientific visualization algorithms that work efficiently on ECP accelerators. Also provides a framework for developing new algorithms.	C++11, DIY, OpenMP (optional), CUDA (optional)	Ongoing developer support. Established consolidated codebase among multiple projects for sustainable funding. Provide comprehensive user's guide.

VTK-m: Impact Goals & Metrics

KPP-3 Goals and Metrics and current actual value (if there is something unique here we should state it) (.5 page) (table)

Short Name					
Impact Goal:	VTK-m is integrated with ECP applications				
Metric:	TBD				
Threshold Value:	TBD	**Objective Value:**	TBD	**Actual Value:**	TBD

Impact Goal:	VTK-m is high quality, sustainable software				
Metric:	TBD				
Threshold Value:	TBD	**Objective Value:**	TBD	**Actual Value:**	TBD
Impact Goal:	VTK-m provides optimized performance on ECP hardware				
Metric:	TBD				
Threshold Value:	TBD	**Objective Value:**	TBD	**Actual Value:**	TBD

VTK-m: Major Integration Points

Major integration points and dependencies table (includes Application codes, other ST products, or Facility line sights) (.5 page) (table) - use the planning process information to start

Target User	Software Product Provided	Capability Provided
ALPINE (2.3.4.16)	VTK-m will be integrated into ALPINE tools (ParaView, VisIt, Ascent) and core visualization algorithms will be delivered.	Porting of visualization algorithms to A21, O21, and exascale platforms.
ATDM SNL (2.2.5.03)	VTK-m integration in ALPINE tools.	Visualization capabilities that work in situ.
ATDM LANL (2.2.5.01)	VTK-m integration in ALPINE tools.	Support for in situ visualization to generate Cinema products.
ATDM LLNL (2.2.5.02)	VTK-m integration in ALPINE tools.	Rendering capabilities for radiative transfer

VTK-m: Accomplishments to Date

Short statement of status and work completed to date (what technical risks have been retired) (1 page) (text) –build from input to CAR reports?

- I think of this as possibly a short paragraph with a bulleted lists of major accomplishments (O(1-2) per year)

The VTK-m project is organized into many implementation activities. The following is a sample of features that have so far been implemented.

- *Key Reduce Worklet:* This adds a basic building block to VTK-m that is very useful in constructing algorithms that manipulate or generate topology.
- *Spatial Division:* Introductory algorithms to divide space based on the distribution of geometry within it. This is an important step in building spatial lookup structures.
- *Basic Particle Advection:* Particle advection traces the path of particles in a vector field. This tracing is fundamental for many flow visualization techniques. Our initial implementation works on simple structures.

- *Surface Normals:* Normals, unit vectors that point perpendicular to a surface, are important to provide shading of 3D surfaces while rendering. These often need to be derived from the geometry itself.
- *Multiblock Data:* Treat multiple blocks of data as first-class data sets. Direct support of multiblock data not only provides programming convenience but also allows us to improve scheduling tasks for smaller groups of data.
- *Gradients:* Gradients are an important metric of fields and must often be derived using topologic data. Gradients are also fundamental in finding important vector field qualities like divergence, vorticity, and q-criterion.
- *Field to Colors:* Pseudocoloring is a fundamental feature of scientific visualization, and it depends on a good mechanism of converting field data to colors.
- *VTK-m 1.1 Release:* VTK-m 1.1 was released in December 2017.
- *Extract External Surface:* When rendering solid objects, it is only necessary to render the external surface of the object as the interior of the volume is hidden. As such, external surface extraction is one of the most common operations in scientific visualization applications.
- *Coordinate system transformations:* 3D rendering systems operate in Cartesian coordinate systems. However, data are sometimes referenced in cylindrical or spherical coordinate systems to match the physical structure of the data. Consequently, visualization tools need a fast method to convert coordinates among different spaces.
- *Affine Transformations:* Affine transformations, which include translate, rotate, and scale, are a common and important method to manipulate objects in 3D space.
- *Location Structures:* Finding mesh components from a coordinate in space requires search structures.
- *Rendering Topological Entities:* It is often helpful to analysts to view representations of the constitute components of a mesh.
- *OpenMP:* Much of the code we wish to integrate with uses OpenMP, which can conflict with other threading implementations. VTK-m now supports multiple threading libraries, including OpenMP, to better match the code it integrates with.
- *Dynamic Types:* To better support integration into dynamic programs like ParaView and VisIt, we introduced virtual methods into VTK-m's processing. This dramatically reduces the compile time and executable size of builds.
- *VTK-m 1.3 Release:* VTK-m 1.3 was released in November 2018.
- *ZFP:* Implemented the ZFP algorithm in VTK-m, which allows it to be ported across ECP architectures and integrated with other visualization code.
- *Clip:* Clip operations intersect meshes with implicit functions. It is the foundation of spatial subsetting algorithms, such as "box," and the foundation of data-based subsetting, such as "isovolume."

VTK-m: Major Remaining Technical Risks

Table of 1-3 technical risks that are within the project's control to retire and the anticipated date they will be retired, mitigation strategy (what would be required to extend or increase the size or your project to

accommodate this risk) (.5 page) (table (risk, mitigation strategy, date retired)) – Examples – algorithm doesn't scale well. Code doesn't perform well on accelerated architecture. Numerical method doesn't converge.

Major Technical Risk	Mitigation Strategy	Anticipated Retirement Date
VTK-m code not available for ST and AD needs.	Deploy VTK-m through partnering ECP software products.	6/30/2023
Visualization software does not perform well on available ECP hardware	Optimize VTK-m on O21 and A21 hardware, making changes as necessary	9/30/2022

VTK-m: FY20-FY23 Technical Plan

Short statement of technical work plan FY20-FY23 (1 page) (from the planning process)

- I think of this as possibly a short paragraph with a bulleted list of major tasks still to be completed O(1-2) per year
- These should clearly address the technical risks still to be retired

Starting in FY20, the ECP/VTK-m project will shift from primarily building functionality into the VTK-m toolkit to addressing the needs of other ST projects and ECP applications. This work will focus on the three key goals of ECP: performance, integration, and quality.

- Integration: As the linchpin for porting scientific visualization software to the accelerators and other compute hardware required for Exascale, it is critical that the VTK-m software gets integrated into the visualization software used by ECP.
    - SPACK deployment releases.
    - Initial round of ParaView and VisIt integration.
    - Expansion of scheduling across block-level objects.
    - Better management of devices.
- Quality: It is a critical challenge to ensure that VTK-m builds, runs, and performs well on all ECP platforms. VTK-m's design of write once run everywhere saves significant development time, but also challenges the development team to ensure that the software works correctly in a variety of dissimilar build environments.
    - Better input data file support in VTK-m testing.
    - Rendering-based correctness verification
    - Establish our regression testing directly on ECP platforms and expand our testing capabilities for better functionality checking.
    - Increase outreach with expanded tutorials, publications, and other activities.
    - Performance regression testing.
- Performance: The ECP Data and Visualization ST projects are all consolidating their accelerator-based software in VTK-m. It is therefore critical that the ECP/VTK-m project deliver code that performs well on processors used in ECP machines. If it does not, then

ECP's data analysis and visualization software will run poorly or not at all, and scientific discovery will be hampered.

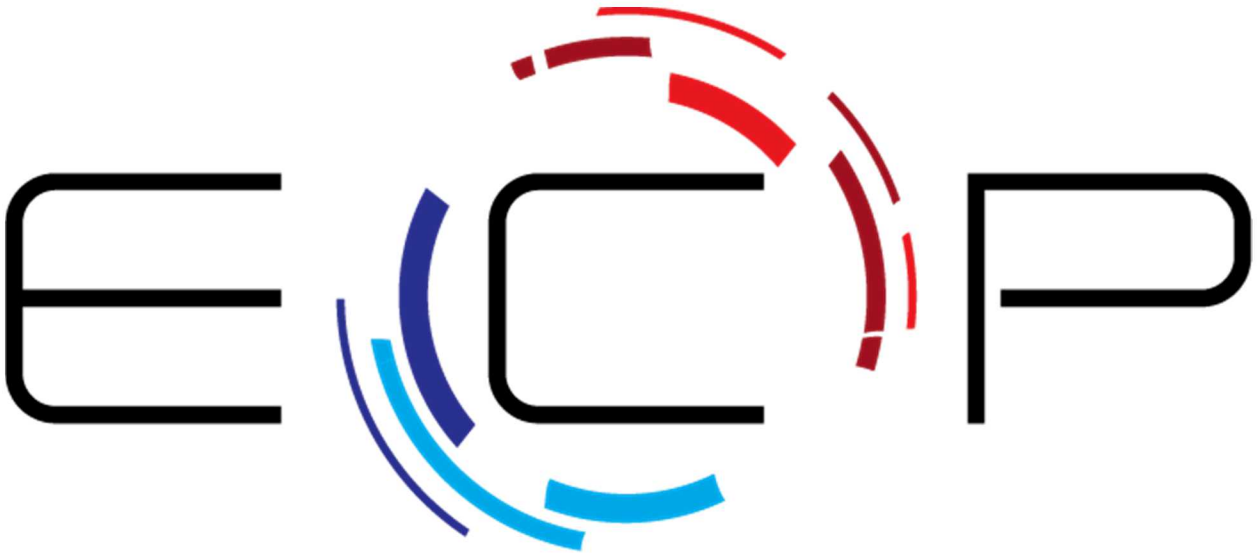
- Establish a suite of performance evaluation software to provide a consistent metric of performance of the VTK-m software on various platforms.
- Porting to O21 and A21 architecture.
- Optimizing for O21 and A21 architecture.
- Deploy performance code on O21 and A21.

VTK-m: FY20-23 Milestones

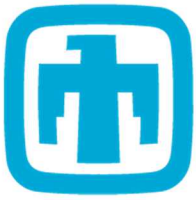
(titles that hopefully reflecting the work to be done) (.5 page) (table)

Due Date	Milestone Title
1/31/2020	SPACK deployment releases
4/30/2020	Initial round of ParaView and VisIt integration
10/31/2020	Establish performance evaluation suite
1/31/2021	Better input data file support in VTK-m testing
4/30/2021	Rendering-based correctness verification
7/31/2021	Expansion of scheduling across block-level objects
10/31/2021	Porting to O21 and A21 architecture
1/31/2022	Establish regression testing directly on ECP platforms
4/30/2022	Better management of devices
7/31/2022	Increase outreach
10/31/2022	Deploy performance code on O21 and A21
1/31/2023	Performance regression testing
6/30/2023	Complete integration into ParaView and VisIt

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.



EXASCALE COMPUTING PROJECT



Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration