ECP/VTK-m Updating HPC Visualization Software for Exascale-Era Processors

Kenneth Moreland Sandia National Laboratories
David Pugmire Oak Ridge National Laboratory
David Rogers Los Alamos National Laboratory
Berk Geveci Kitware, Inc.
Hank Childs University of Oregon

<WBS 1.3.5.05>

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.
SAND2017-0582 D

U.S. DEPARTMENT OF ENERGY Sandia National Laboratories

EXASCALE COMPUTING PROJECT

VTK-m

http://m.vtk.org

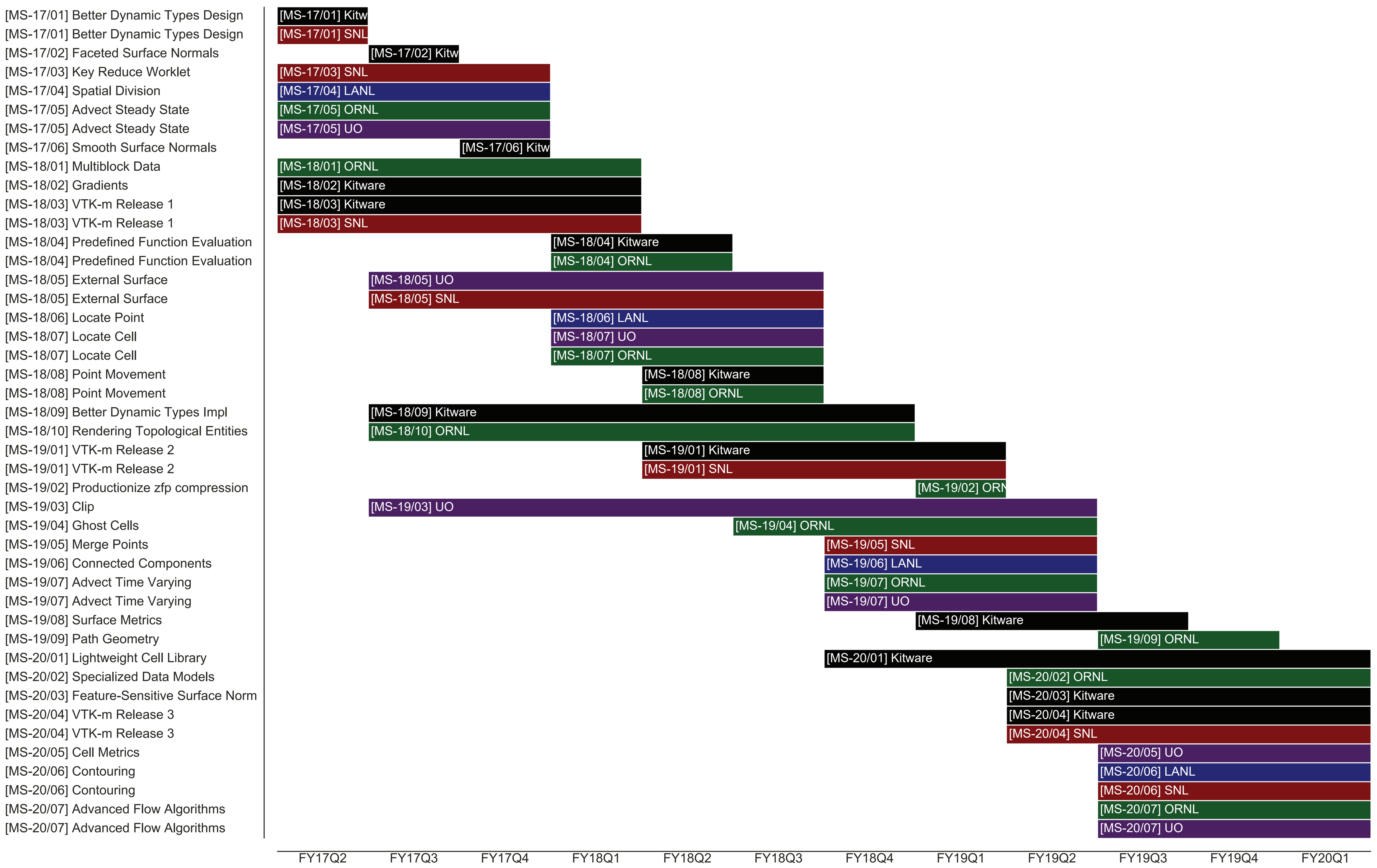
Project Description:

The VTK-m software is DOE's solution to develop and deploy scientific visualization software that take advantage of the shared-memory parallelism available on many-core CPUs and accelerators such as Intel Xeon Phi's and Nvidia GPUs.

ECP Scope:

Redeveloping, implementing, and supporting necessary visualization algorithms on many-core under VTK-m. There is a large base of complex, computationally intensive algorithms in regular use that need to be redesigned for advanced architectures. Updating the many critical scientific visualization algorithms in use today requires significant effort.

Project Plan

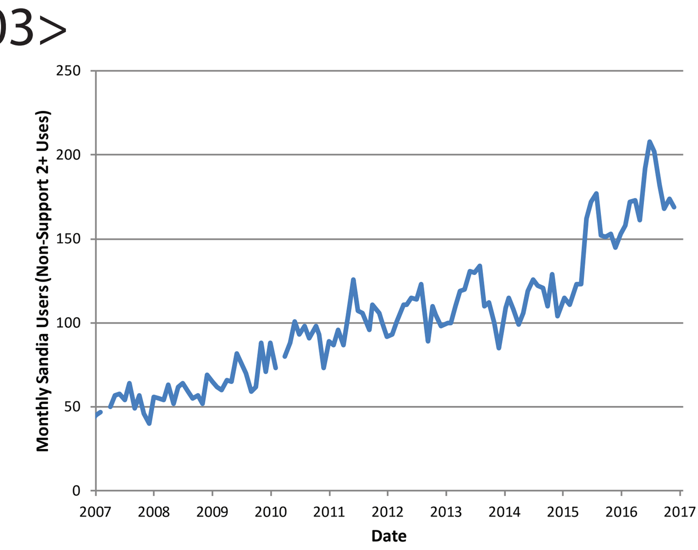


Complementary Project

ATDM Scalable Visualization <WBS 1.3.5.03>

Project Description:

Provide visualization and analysis needs of the ASC/ATDM applications on next-generation, many core platforms. Re-engineer the visualization algorithms, services, and tools that enable ASC customers to carry out data analysis on ASC systems and ACES platforms. Current tools include scalable data analysis software released open source through ParaView, VTK, and the in situ data analysis library (Catalyst).



ECP Scope:

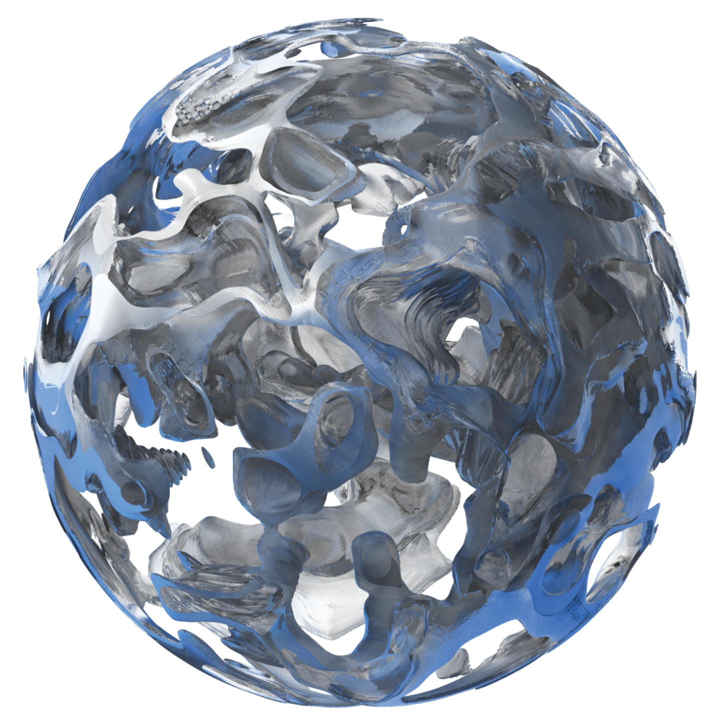
ASC/ATDM funded R&D and ASC application support.

Maturity Indicators and Metrics:

ParaView released in production, 15+ years development.

Recent Technical Achievements

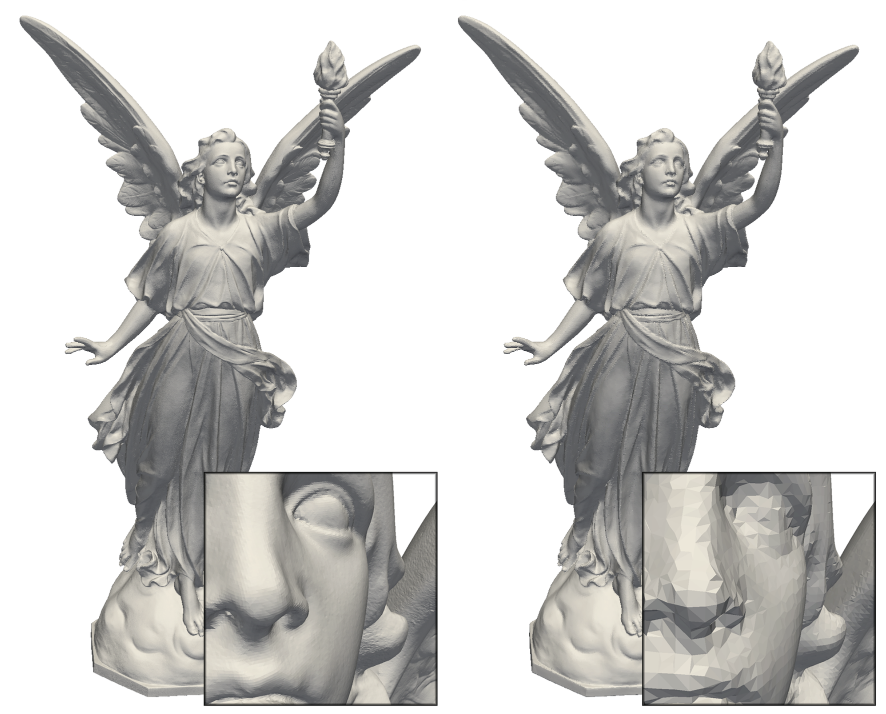
Ray Cast Rendering



Data Set	Algorithm	Millions of Rays Per Second
LT_350K	Embree	51.9
	EAVL	27.7
	VTK-m	38.5
LT_372K	Embree	56.5
	EAVL	26.1
	VTK-m	36.0
RM_350K	Embree	64.8
	EAVL	33.3
	VTK-m	47.8
RM_650K	Embree	65.9
	EAVL	35.6
	VTK-m	49.1
RM_970K	Embree	59.1
	EAVL	29.3
	VTK-m	41.0
RM_1.7M	Embree	52.4
	EAVL	27.0
	VTK-m	37.8
RM_3.2M	Embree	48.4
	EAVL	28.3
	VTK-m	33.9
Seismic	Embree	43.2
	EAVL	25.2
	VTK-m	34.5

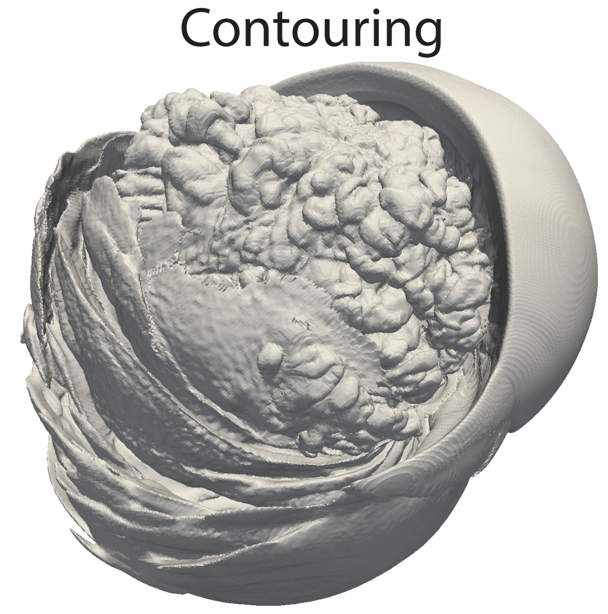
Data Set	Algorithm	Millions of Rays Per Second
LT_350K	OptiX Prime	357.6
	EAVL	150.8
	VTK-m	164.5
LT_372K	OptiX Prime	322.4
	EAVL	124.7
	VTK-m	140.8
RM_350K	OptiX Prime	436.5
	EAVL	197.5
	VTK-m	200.8
RM_650K	OptiX Prime	420.4
	EAVL	172.9
	VTK-m	166.0
RM_970K	OptiX Prime	347.1
	EAVL	152.8
	VTK-m	163.5
RM_1.7M	OptiX Prime	266.8
	EAVL	136.6
	VTK-m	148.8
RM_3.2M	OptiX Prime	264.5
	EAVL	124.8
	VTK-m	134.5
Seismic	OptiX Prime	267.8
	EAVL	106.3
	VTK-m	119.4

Surface Simplification



Algorithm	Device	512^3	1024^3	2048^3
VTK	Serial	3.65 s	11.40 s	
VTK-m	Serial	2.73 s	2.93 s	5.22 s
VTK-m	TBB 36 threads	0.36 s	0.45 s	0.72 s
VTK-m	TBB 72 threads	0.41 s	0.49 s	0.74 s
VTK-m	CUDA	0.18 s	0.19 s	0.20 s

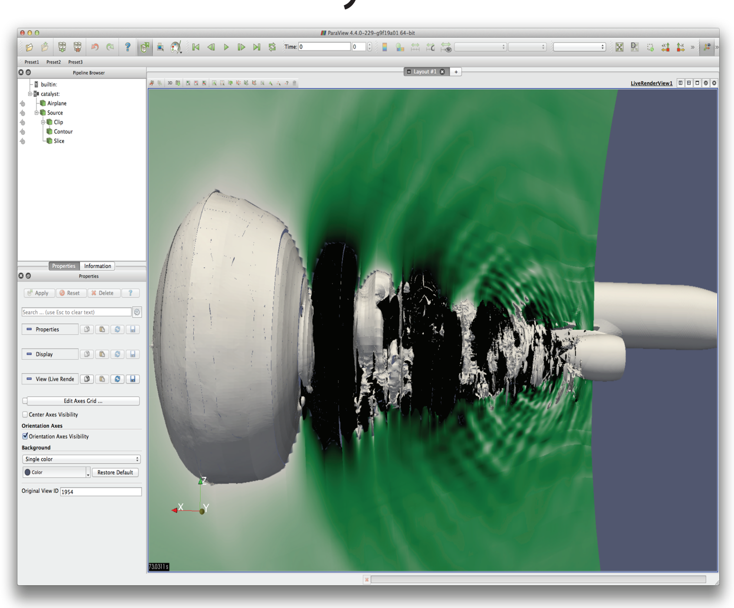
Contouring



Algorithm	Device	Time
vtkMarchingCubes	Serial	11.917 s
vtkMarchingCubes	32 MPI Ranks	1.352 s
vtkMarchingCubes	64 MPI Ranks	1.922 s
PISTON	Serial	19.895 s
PISTON	CUDA	0.514 s
PISTON	TBB	0.955 s
VTK-m	Serial	20.784 s
VTK-m	CUDA	0.560 s
VTK-m	TBB	1.161 s

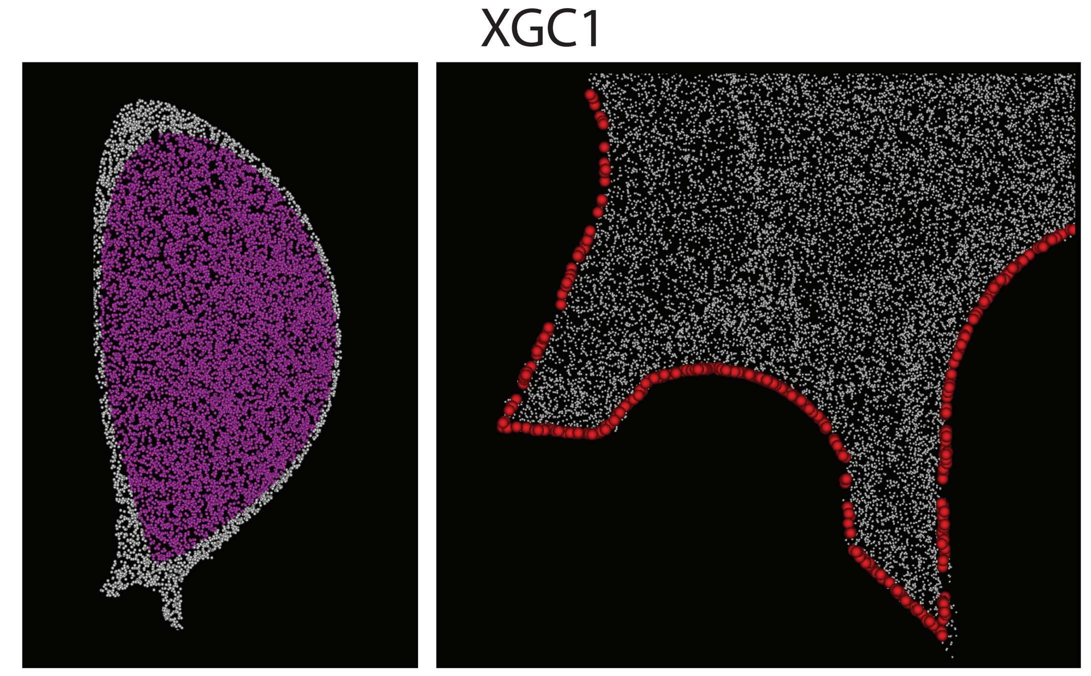
Recent Example Usage

PyFR



We have been exploring the integration of VTK-m with our existing HPC visualization tools. This demonstration performed running VTK-m inside the Catalyst framework to visualize data generated directly on the GPUs in a PyFR simulation running on Titan and interactively controlled on the SC15 expo floor in Austin, TX.

XGC1



We have been exploring the integration of VTK-m with XGC1, a highly scalable physics code used to study plasmas in fusion tokamak devices. Recent work has explored using light-weight plugins to perform visualizations of both the particles and of the field variables in XGC1 using ADIOS and DataSpaces as a lightly coupled in situ framework.