New information technologies, Computing simulation

Accelerating the visualization of spatio-temporal simulations with non-evolving meshes

A. Ribés[1], A. Geay[1], and M. Westphal[2]

EDF Lab[1] Kitware-Europe[2]

Context

Numerical simulations, such as finite elements, deal with both space and time discretizations of fields. Fields are continuous physical quantities (represented by a number or a tensor) that have values for each point in space and time. For the spatial discretization of fields, a mesh is typically used as geometrical support.

Concerning the temporal evolution of the spatial discretization, in the general case, a different mesh is used per time step. However, it is often the case that the mesh is non-evolving, meaning that it stays the same for all time steps. This case is of great importance in industrial practice, in our experience, most industrial simulations belong to this category. Indeed, mesh time-dependent simulations are currently only used for advanced simulation projects.

In this work, we show that taking into account the non-evolving geometrical nature of a simulation leads to a strong acceleration of the visualization operations. We demonstrate this point by use of the ParaView open-source software (***www.paraview.org***). We implement this idea, by use of several plugins, and compare the results with the baseline current implementation of ParaView.

A static mesh optimized (SMO) implementation of a ParaView filter typically uses a cache to store the geometrical information. It will then only compute operations on field data and attach it to a shallow copy of the cached mesh. A static filter can detect if its cache becomes invalid. In that case, the cache will be recomputed and the general architecture of ParaView does not lose its generality. Figure 1 presents a comparison between the standard VTK pipeline used by ParaView (top diagram) and the developed cached VTK pipeline (bottom panel). The StaticMeshPlugin for ParaView can be cloned and compiled from:

https://gitlab.kitware.com/paraview/staticmeshplugin

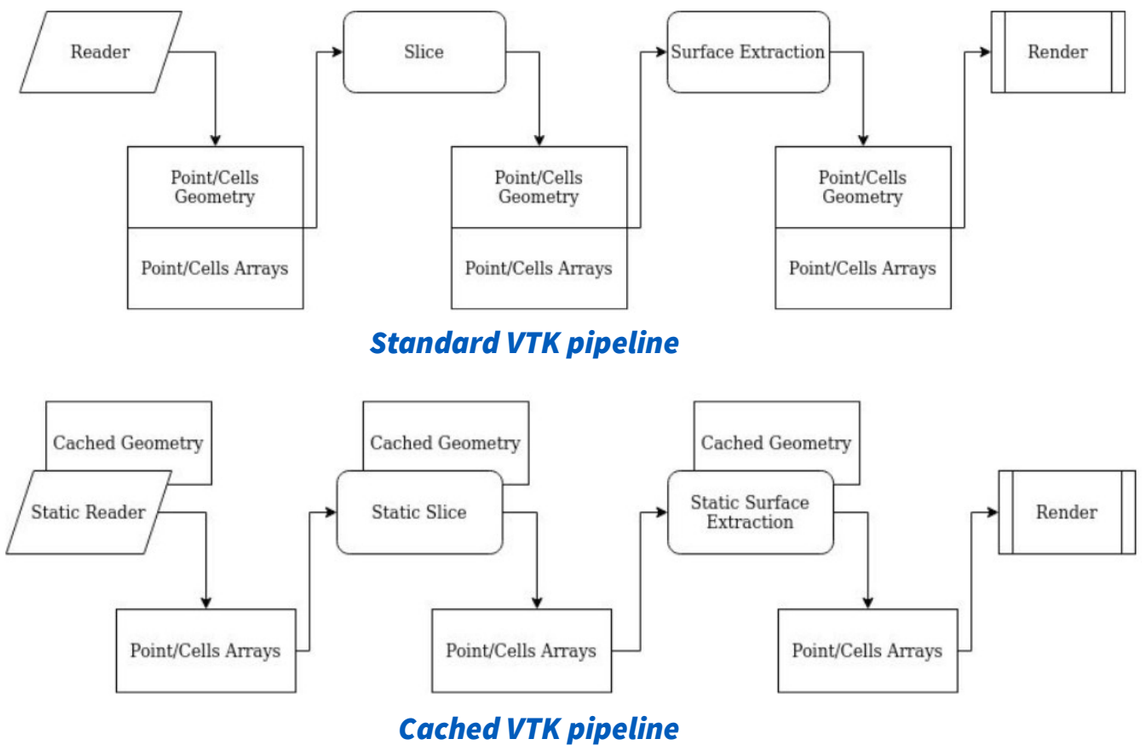

FIGURE 1
Comparison between the standard VTK pipeline used by ParaView (top diagram) and the developed cached VTK pipeline (bottom panel).

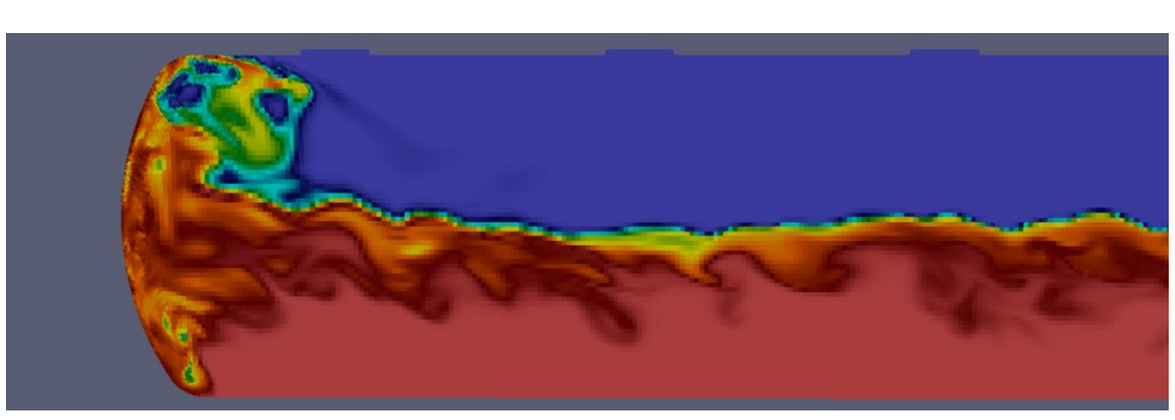

FIGURE 2
A screenshot from a temporal dataset, used in our tests, with a non-evolving mesh, from the CFD solver Code_Saturne. Thanks to a static mesh optimized filter, the dataset shown in this figure was animated interactively by ParaView on a laptop, instead of using the ParaView server mode on a cluster.

Experimental results

We present results on **two datasets** using non-evolving meshes. The first dataset contains 16M cells. It is synthetic and has been generated by scripting for test purposes.
The second dataset has been obtained by executing the CFD solver Code_Saturne (***www.code-saturne.org***), which generated an Ensight Gold Binary output file containing approximately 13M cells and 34M nodes. Figure 2 shows a slice of this dataset.

We chose **two ParaView filters** for benchmarking these datasets:

- *SurfaceFilter*, which extracts the polygons forming the outer surface of the input dataset;
- *SliceWithPlane*, which extracts the portion of the input dataset that lies along a specified plane.

We remark that *SurfaceFilter* is called by ParaView each time a 3D rendering of an object surface is performed, thus being probably the most CPU time-consuming filter involved in the visualization of fields.

Table 1. Results for the synthetic dataset

	SurfaceFilter	SliceWithPlane
Baseline	25.5137s	17.9882s
Accelerated	0.8198s	0.0674s
Speedup	**31.2**	**266.65**

Table 2. Results for the CFD Code Saturne dataset

	SurfaceFilter	SliceWithPlane
Baseline	7.4548s	2.4890s
Accelerated	0.0693s	0.0032s
Speedup	**107.5**	**777.8**

TABLES
*Tables 1 and 2 show the results for both datasets and filters. The presented timings are averaged values of three executions of the code. We observe that speedups for the **SurfaceFilter** are very significant (**31.2** and **107.5**). These speepups are even better, of the order of hundreds, for **SliceWithPlane** (**266.65** and **777.8**).*

Contact EDF R&D :
Alejandro RIBES, alejandro.ribes@edf.fr