



中國地質大學
China University of Geosciences

艰苦朴素 求真务实

温家宝

艰苦朴素
求真务实
温家宝

2 用户使用Catalyst

李健

中国地质大学



使用Catalyst增强的工作流程，用户预设需要的可视化和分析输出，作为一个预处理步。在模拟运行时生成输出，之后用于用户的分析。Catalyst输出可以有多种格式，诸如使用伪颜色制作的几何形体的渲染图形、绘制图形（如条状图、线状图等）、数据提取（如等值面、切面或流线等）以及计算量（如最大剪切力、升力等）。

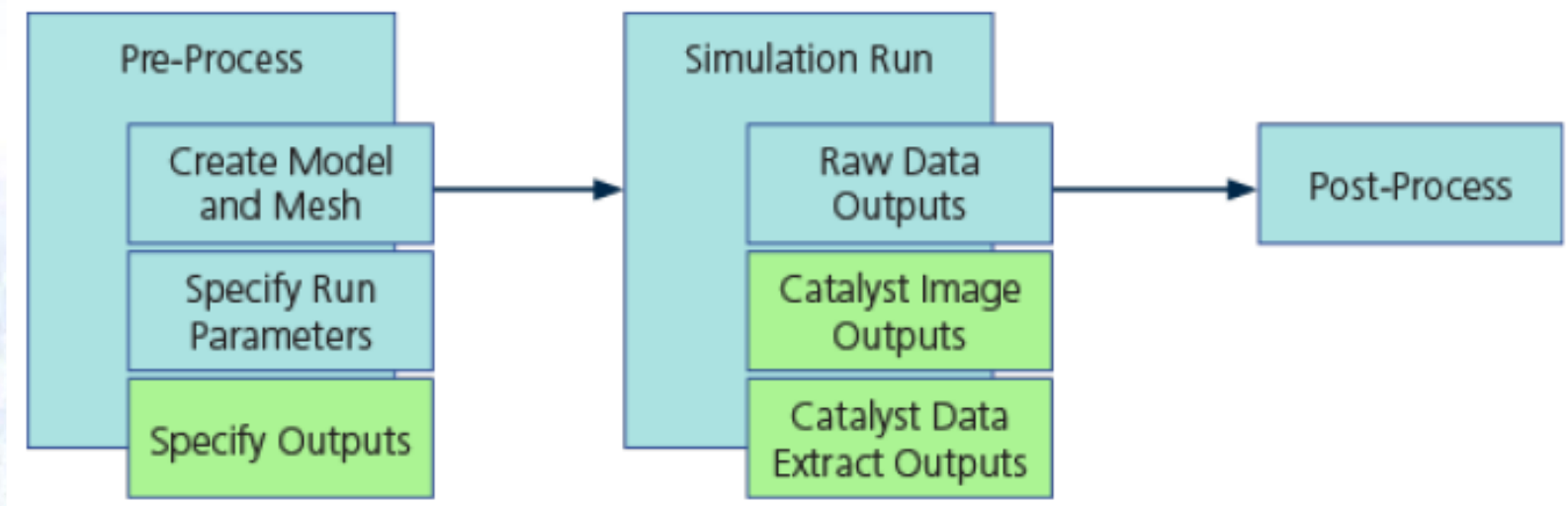


图2.1传统的工作流程（蓝色）和Catalyst增强的工作流程（绿色）



ParaView的In-situ可视化实现

用户有2种方式使用Catalyst实施in-situ可视化：

第1种，定义一套参数，传入预设的Catalyst管线；

第2种，用户在Paraview GUI中创建Catalyst管线脚本。



ParaView的In-situ可视化实现

ParaView使用VTK来处理数据和作为渲染引擎，使用Qt开发用户界面。使用Python语言实现完全的脚本操作。通过Python对引擎做的所有改变都自动反馈到用户界面。

ParaView以batch作业来运行。可以通过编写接口的XML描述，或者编写C++类增加其他的模块。XML接口允许用户或开发者增加自己的VTK过滤器，且无需编写特别的代码和重新编译。



ParaView的In-situ可视化实现

ParaView以MPI并行运行，包括集群、可视化系统、大型服务器和超算。ParaView使用数据并行模型，其中数据分解为pieces，由不同进程处理。使用ghost points/cells通信，交换相邻信息。还支持**分布式渲染**（数据在各计算节点上渲染，之后使用**depth buffer**合成）、**本地渲染**（在一个计算节点上收集计算得到的多边形，本地渲染）以及**两者混合使用**（例如level-of-detail模型，可以本地渲染，而完整模型以分布式方式渲染）。



ParaView的In-situ可视化实现

用户界面（ParaView-GUI）使用client/server模式在电脑上运行。以这种方式，用户可以利用远程的HPC渲染集群。ParaView Client是串行应用程序，总是以paraview命令运行。Server是MPI并程序，必须用并行化作业来启动。有2种模式来启动ParaView Server:

- (1) 所有数据处理和渲染都以相同的并行作业运行，使用pvserver命令启动server
- (2) 数据处理是以1个并行作业处理，而渲染是以另一个并行作业处理，分别使用pvdataserver和pvrenderserver命令。将数据处理与渲染分开的目的是：可以使用2个不同的并行计算机，一个具有高端的CPU，一个具有高端的GPU。但是将Server功能分成2个部分需要重新分区数据和两者之间转移数据，这造成的开销，相对以相同作业下实施数据处理与渲染，一般不可忽略。因此，建议几乎所有的应用就简单地用一个server执行。



预设Catalyst管线

这部分工作主要由**模拟开发者**来完成，将方便模拟使用者，降低使用门槛。概念就是：

对于大多数过滤器，仅需要设置**有限数目的参数**即可。例如，对于切面过滤，用户仅需要设置穿过平面的点和平面的法向。

另一个例子是，设置**阈值过滤器**，其中仅需要定义变量和范围。对每一个管线，参数还需要包含**待输出的文件名和输出频率**。



创建Catalyst管线Python脚本

这种方法允许用户可自由添加增多的功能来控制输出。用户可在Paraview GUI中创建他们自己的Catalyst Python脚本。

这种方式需要2个条件：

(1) Paraview启用了**CoProcessing Script Generator**插件（编译时默认启用），
注意：**插件版本应对应Catalyst版本。**

(2) 用户已有**代表性数据集**，由此开始：读入的数据集有相同的数据类型（vtkUnstructuredGrid, vtkImageData等），与模拟**适配器(adaptor)**代码中网格上定义的属性相同。



创建Catalyst管线Python脚本

在GUI中创建Catalyst Python脚本的步骤如下：

1.加载创建脚本的插件：

Tools->Manage Plugins...-> CatalystScriptGeneratorPlugin -> Writes and CoProcessing

(2) 加载代表性数据和创建脚本：不是真的要输出文件，而是创建将在模型运行期间将要**在哪输出**文件。在**Writers**菜单下，选择提取数据的合适信息。用户应定义文件名及输出频率。文件名必须包含**%t**，当创建文件时，将被**时间步**代替



创建Catalyst管线Python脚本

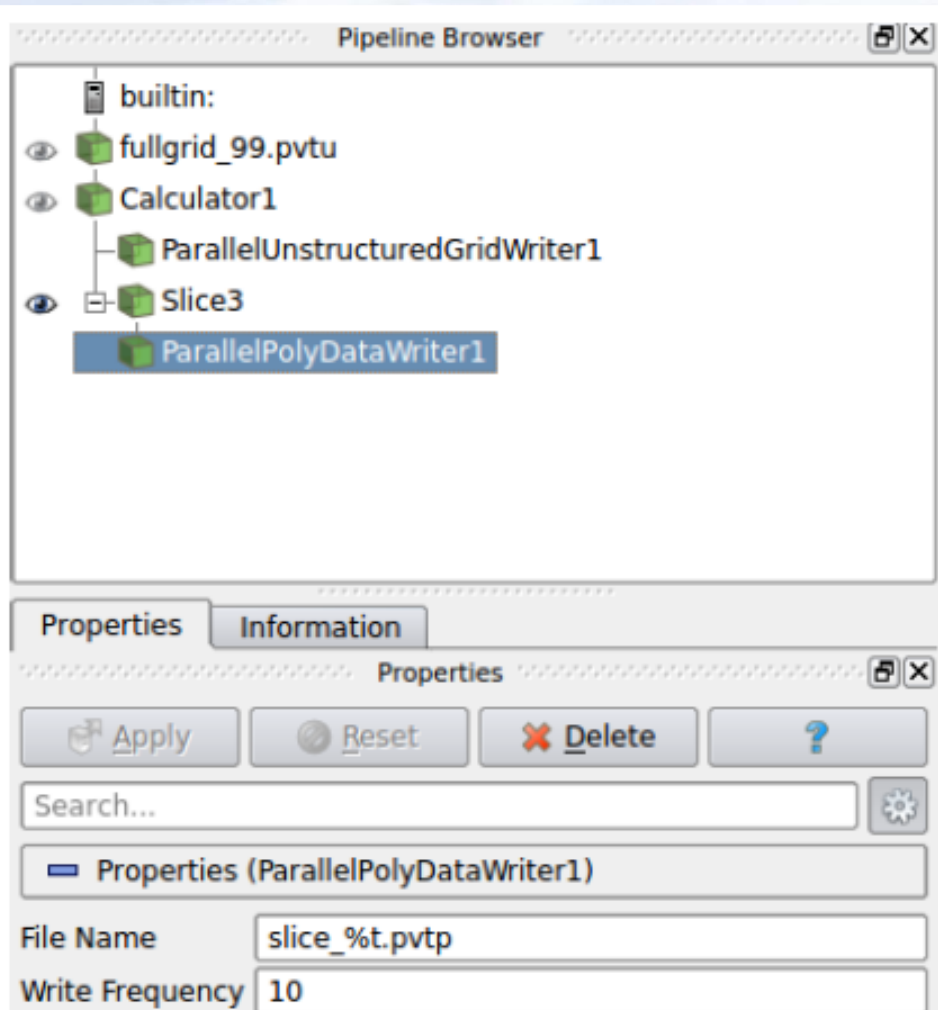


图2.2 一个管线有2个writer的例子：
第1个从Calculator过滤输出，第2个
从切面过滤输出



创建Catalyst管线Python脚本

(3) 当创建了完整的Catalyst管线后，必须从Paraview输出Python脚本。

选择CoProcessing菜单下面的Export State控件。用户可点击跳出的初始化窗口中的Next按钮。



(4) 完成以上步骤，用户必须选择源（即没有输入的管线对象），适配器将创建和添加这些源到输出。注意：通常这不包括从Sources菜单，因为生成的Python脚本将实例化这些对象（例如画流线的seed）。图2.2中的源是filename_10.pvti读取器，类比于模型代码适配器提供的输入。用户或者点击左box中的源，添加到右box，或者选中左box中源，点击Add。如图2.3。选中所有的源后，点击Next。



创建Catalyst管线Python脚本

本例中，选择filename_10.pvti源，做法：

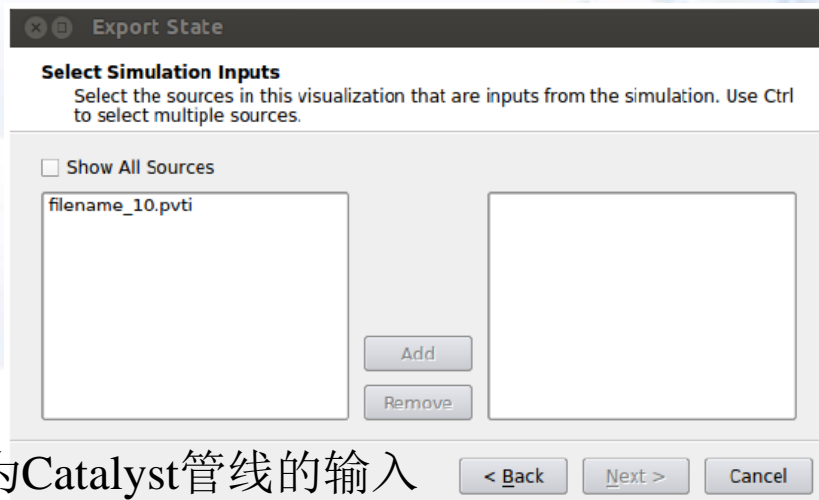


图2.3 选择filename_10.pvti作为Catalyst管线的输入

(5) 标记输入。点击Next

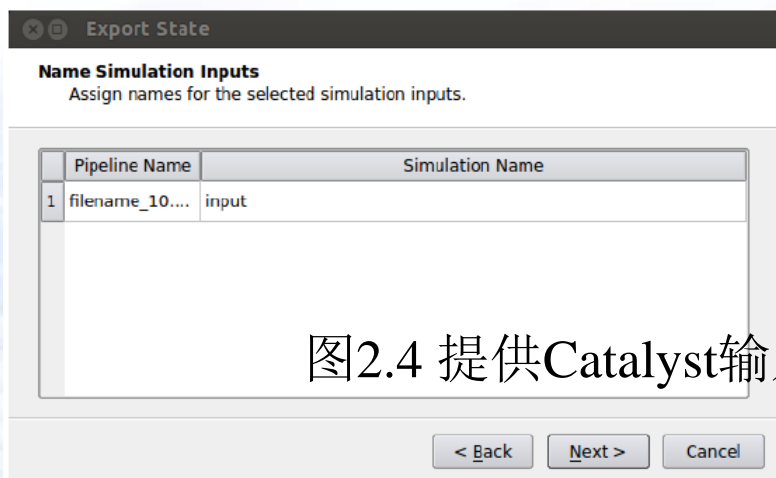


图2.4 提供Catalyst输入的标识字符串



创建Catalyst管线Python脚本

(6) 允许Catalysy检查Live Visualization连接，从不同视图输出屏幕图像。

Paraview Live Visualization和Cinema在后文讨论。

对于**屏幕显示(Screenshots)**，有多个选项，见图2.5：

第1个全局选项是**Rescale the lookup table**，将伪颜色尺度化到数据范围。每个视图的其他选项有：

Image Type — 输出到屏幕的图片格式；

File Name – 创建的文件名，必须包含%t，实际模拟时间步将代替%t；

Write Frequency – 创建屏幕显示的频率；

Magnification – 用户将创建比当前Paraview GUI分辨率更高的图像；

Fit to Screen – 指定是否将数据扩展到合适的屏幕显示，类似于GUI中点击**扩展**按钮。



创建Catalyst管线Python脚本

Configuration

Select state configuration options.

- ☐ Live Visualization
- ☒ Output rendering components i.e. views
- ☐ Rescale to Data Range

Image Type

png

File Name

image_%t.png

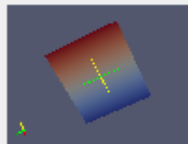
Write Frequency

1

Magnification

1

Fit to Screen ☐



Next View

Previous View

< Back

Finish

Cancel

如果有多个视图，用户需逐个拖入，使用Next View和Previous View按钮。完成后，点击Finish按钮，创建Python脚本。

(7) 最后1步：指定创建的Python脚本文件名。

图2.5 设置输出屏幕图像的参数



创建Catalyst管线Python脚本

创建代表性数据集

有2种方法：

- 第1种，运行带输出完整网格及完整属性信息的Catalyst的模拟；
- 第2种，在Paraview中使用源和过滤器。

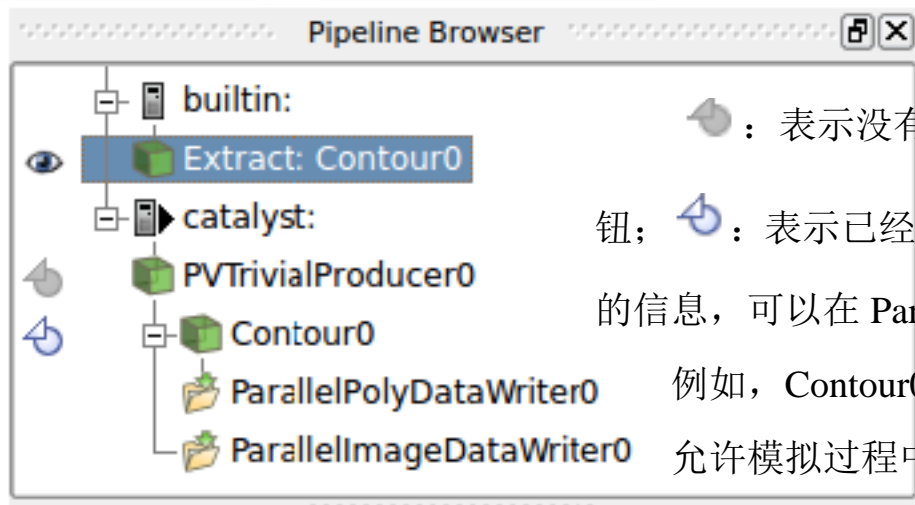
操纵Python脚本

需要会使用Python编程的开发者。



Paraview Live

除了能提前设置管线，通过Paraview的Live功能，分析者可通过Paraview GUI 连接到模拟，可修改当前管线。这对于通过修改管线，改善Catalyst-模拟输出信息的质量。使用Catalyst->Connect...完成连接，这将连接模拟到pvserver。完成连接后，GUI管线如图2.6。



：表示没有 Catalyst 提取数据发送到服务器；如果需要发送，点击此按钮；
：表示已经在 Paraview 服务器上可获取数据了。停止提取发送给服务器的信息，可以在 Paraview 服务器管线中删除该对象。
例如，Contour0 — 已经发送给 pvserver
允许模拟过程中，暂停 Catalyst 功能的模拟。

图2.6 现场连接的Paraview GUI管线



Cinema-V4.2引入

Cinema是一种基于图像(image-based)的方法来在线分析和可视化的。概念就是：将一个组织好的图像集保存在Cinema数据库中，分析者可从生成的图像中实施事后分析和可视化。

可使用Catalyst创建Cinema数据库，可通过Paraview GUI的Catalyst Script Generator插件来定义Cinema输出。图2.7显示了启用Cinema输出的扩展选项。



图2.7 Catalyst输出选项的Cinema输出



Cinema-V4.2引入

选项有：

Export Type ---- 该选项定义当生成图像时，应如何操纵视图相机。当前选项包括：**None, Static, Spherical**。None表示对该视图不需要输出Cinema；Static表示不移动相机；Spherical将围绕视图中心，以一定的Phi和Theta角度旋转相机。

Cinema Track Selection --- 该选项允许改变过滤器参数，以及用于伪颜色生成的场数据。通过旋转左栏中的管线对象，users can specify the arrays to pseudo-color by in the right pane's Arrays tab or the filter's parameters in the right pane's Filter Values tab. 注意，当前仅使用**Slice, Contour和Cut过滤器**修改过滤器值。



避免数据爆炸

创建管线时，过滤器的选择和量级会极大地影响Catalyst和Paraview的计算效率。当发送大规模数据到Paraview服务器时，由于内存限制，由于内存不足会导致计算终止。最坏的情况就是：发送一般化的网格数据结构，如非结构网格；而结构网格数据很紧凑。下面罗列出常用过滤器的内存使用效率分类：

当创建管线时，过滤器通常以某种方式组织，来限制数据爆炸。例如，管线应先组织，以降低维度。另外，降低维度比提取数据更好（例如，Slice过滤器比Clip过滤器更倾向使用）。仅当降低一个或多个数量级的数据规模时才使用提取(extract)。当输出提取的数据时，能使用subsampling（例如，Extract Subset过滤器或Decimate过滤器），来降低文件大小，但需注意降低数据规模不应隐藏细节特征。



避免数据爆炸

1、几乎不占用多少内存的:

- | | | |
|---------------------|-----------------------|-------------------|
| • Annotate Time | • Glyph | • Outline |
| • Append Attributes | • Group Datasets | • Outline Corners |
| • Extract Block | • Histogram | • Plot Over Line |
| • Extract Datasets | • Integrate Variables | • Probe Location |
| • Extract Level | • Normal Glyphs | |

2、添加场数据—使用相同的网格，但需要存储额外的变量。

- | | | |
|---------------------------|----------------------------|-----------------|
| • Block Scalars | • Curvature | • Gradient |
| • Calculator | • Elevation | • Level Scalars |
| • Cell Data to Point Data | • Generate Ids | • Median |
| • Compute Derivatives | • Generate Surface Normals | • Mesh Quality |
| • Octree Depth Limit | • Random Vectors | • Transform |
| • Octree Depth Scalars | • Resample with Dataset | • Warp (scalar) |
| • Point Data to Cell Data | • Surface Flow | • Warp (vector) |
| • Process Id Scalars | • Surface Vectors | |

3、拓扑关系改变，降维---输出多边形数据，但输出单元是1个或多个，但少于输入单元维度。

- | | | |
|------------|---------------------------|----------------------|
| • Clip | • Extract Cells by Region | • Quadric Clustering |
| • Decimate | • Extract Selection | • Threshold |



避免数据爆炸

5、**拓扑关系改变**，没有降维--- 当改变数据集的拓扑关系，不减小数据集的单元数，输出多边形或非结构网格格式。

- | | | |
|-------------------|------------------------|-------------------|
| • Append Datasets | • Extract Edges | • Subdivide |
| • Append Geometry | • Linear Extrusion | • Tessellate |
| • Clean | • Loop Subdivision | • Tetrahedralize |
| • Clean to Grid | • Reflect | • Triangle Strips |
| • Connectivity | • Rotational Extrusion | • Triangulate |
| • D3 | • Shrink | |
| • Delaunay 2D/3D | • Smooth | |