IDAJ Conference Online 2023

Real-time Visualization and On-the-fly Post-processing in iconCFD

Prepared by:
Martin Novak et. al.
m.novak@iconcfd.com
Nov 2023

Agenda

- Introduction
- iconCFD V5 Integration
- Demo: CHT Case
- Demo: Piston Pump
- Summary and Outlook

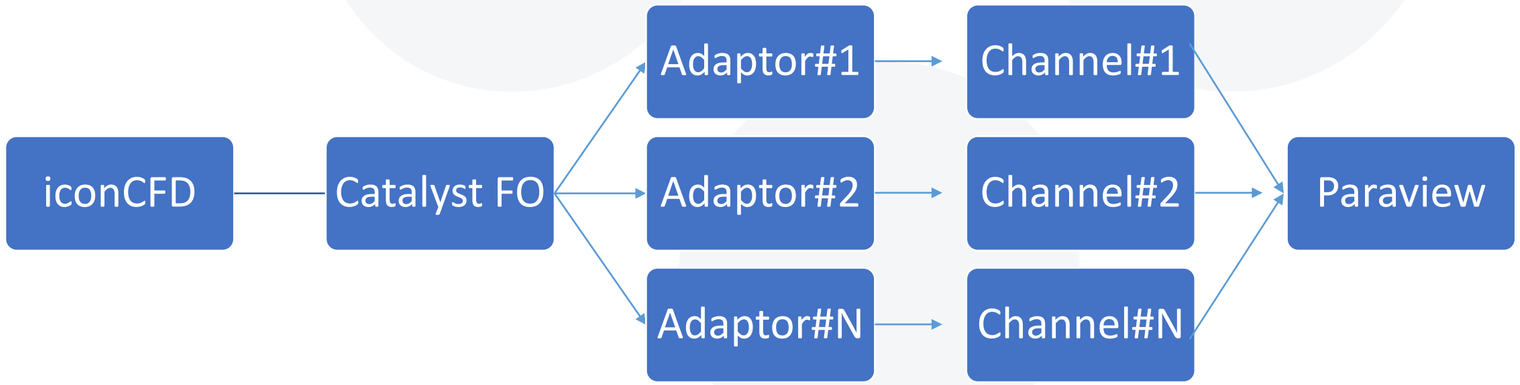
Introduction

- CFD simulations tend to produce large amounts of data that need to be analyzed and visualized efficiently
- Traditional post-processing methods have some drawbacks, such as:
  - They require a lot of disk space to store the data
  - They take a long time to load and process the data
  - They limit the interactivity and flexibility of the analysis
- In-situ post-processing can alleviate all those issues
  - The post-processing utilizes the data in memory, making the data dumps redundant
- Paraview-Catalyst framework provides such real-time visualization capability
- In iconCFD V5 it is now possible to utilize this framework to
  - Observe live data
  - Interactively create a post-processing pipeline for later use

iconCFD V5 Integration

Adaptor

- IconCFD data needs to be communicated to ParaView
- The communication on the iconCFD site is handled via the Catalyst function object
- The Catalyst function object uses adaptors to supply data to ParaView
  - Adaptor can adapt:
    - Volume data
      - Cell centred
      - Face centred
    - Poly data



iconCFD V5 Integration

Communication

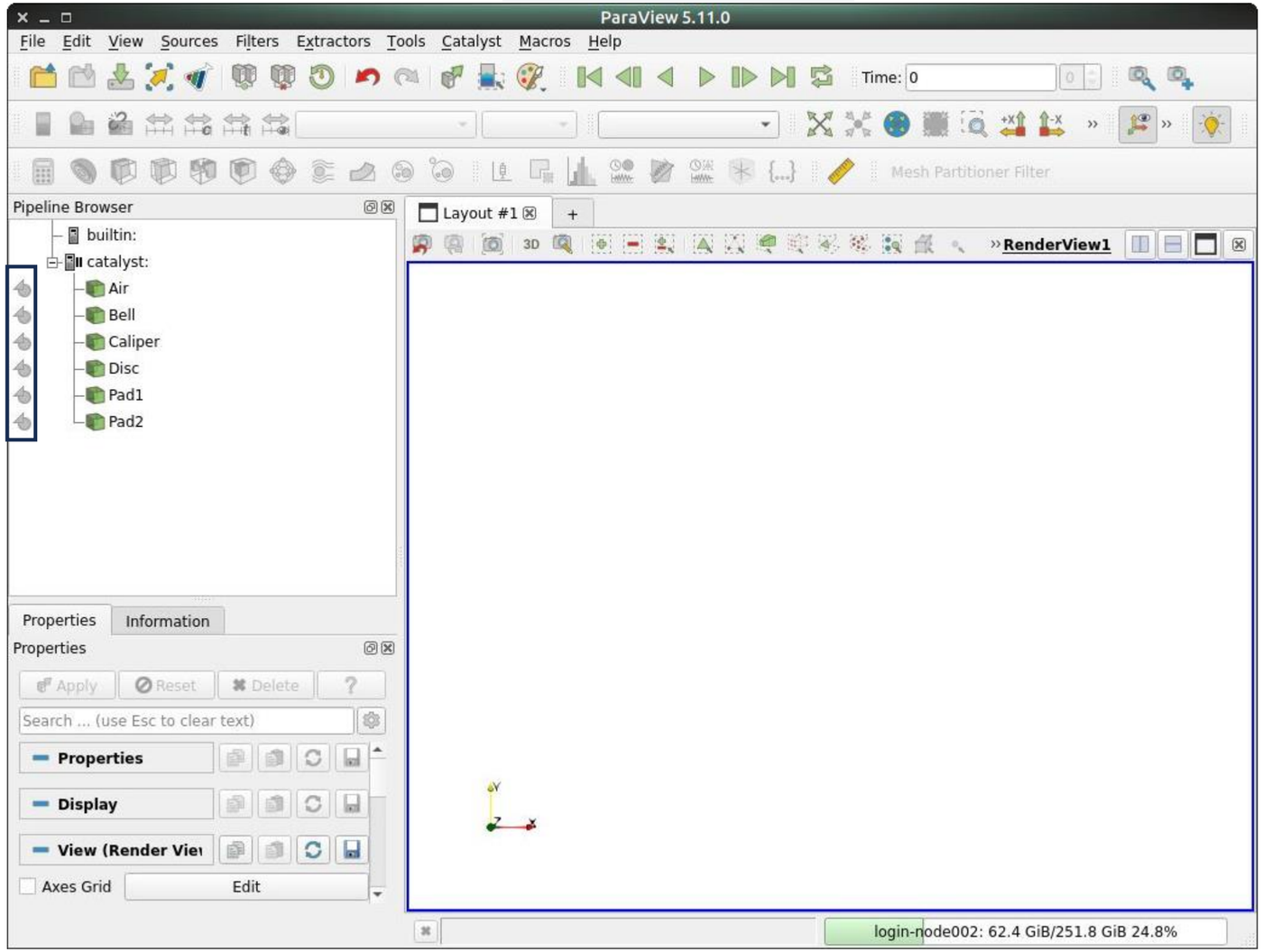
- ParaView pipeline can be controlled via a Python Script
- This applies to ParaView-Catalyst as well
  - Specifies data producers (from channels)
  - Registers producers
  - Determines if live connection is required
  - Defines post-processing pipeline
- Catalyst function object enables to define pipelines to use or sets the most minimal one if nothing is provided
  - The only required information is the hostname and port for live reverse connection from the simulation
  - Paraview pipeline can be then setup using the GUI

Catalyst Demo

CHT Case

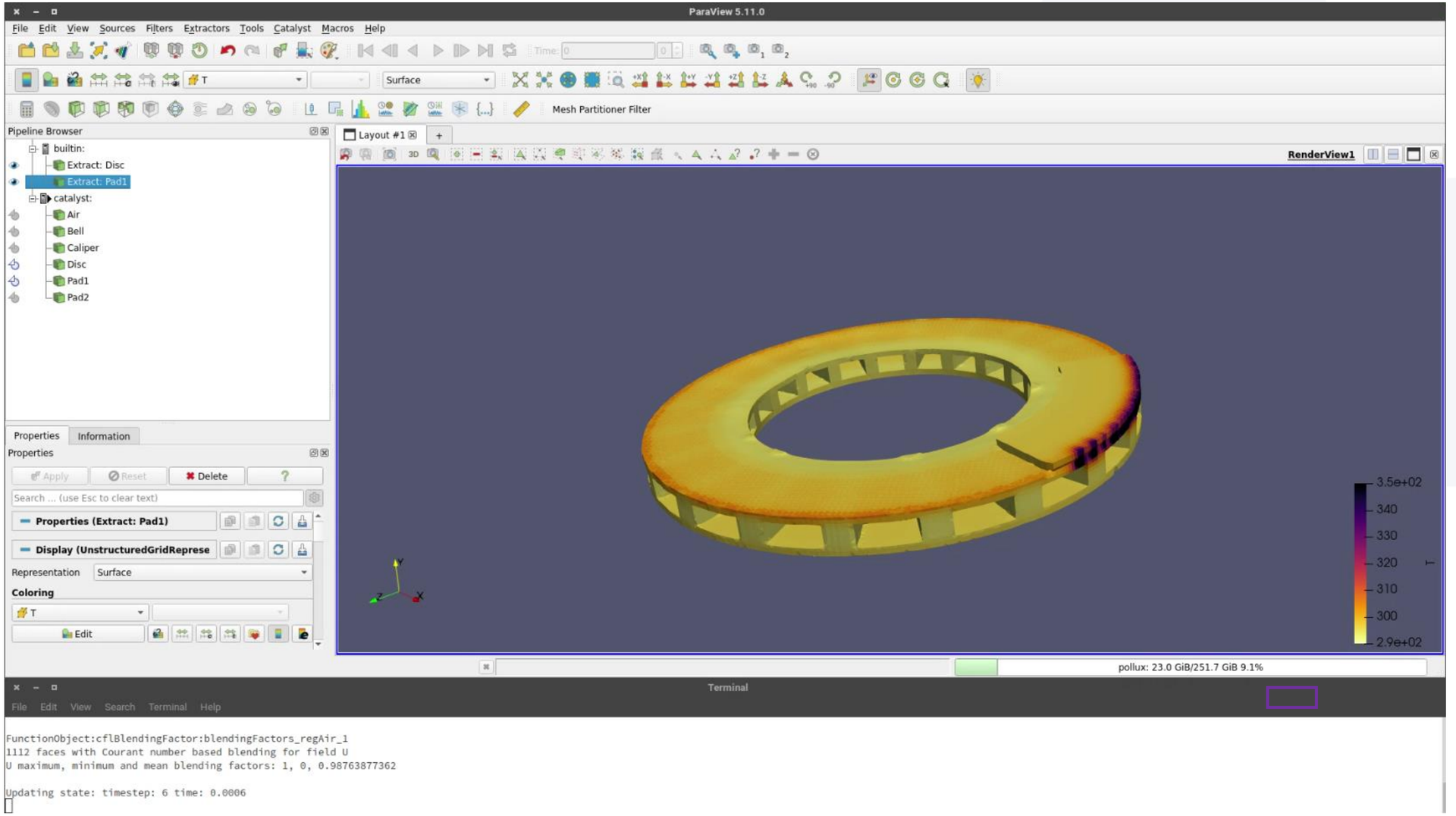
- CHT case with 6 regions
- Each region as one channel

```
from paraview.simple import *
from paraview import catalyst
options = catalyst.Options()
options.EnableCatalystLive = 1
options.CatalystLiveURL = 'login-node002:22222'

Air = TrivialProducer(registrationName='regAir')
Bell = TrivialProducer(registrationName='regBell')
Disc = TrivialProducer(registrationName='regDisc')
Pad1 = TrivialProducer(registrationName='regPad1')
Pad2 = TrivialProducer(registrationName='regPad2')
Caliper = TrivialProducer(registrationName='regCaliper')

def catalyst_execute(info):
    global Air, Bell, Disc, Pad1, Pad2, Caliper
    Air.UpdatePipeline()
    Bell.UpdatePipeline()
    Disc.UpdatePipeline()
    Pad1.UpdatePipeline()
    Pad2.UpdatePipeline()
    Caliper.UpdatePipeline()
```



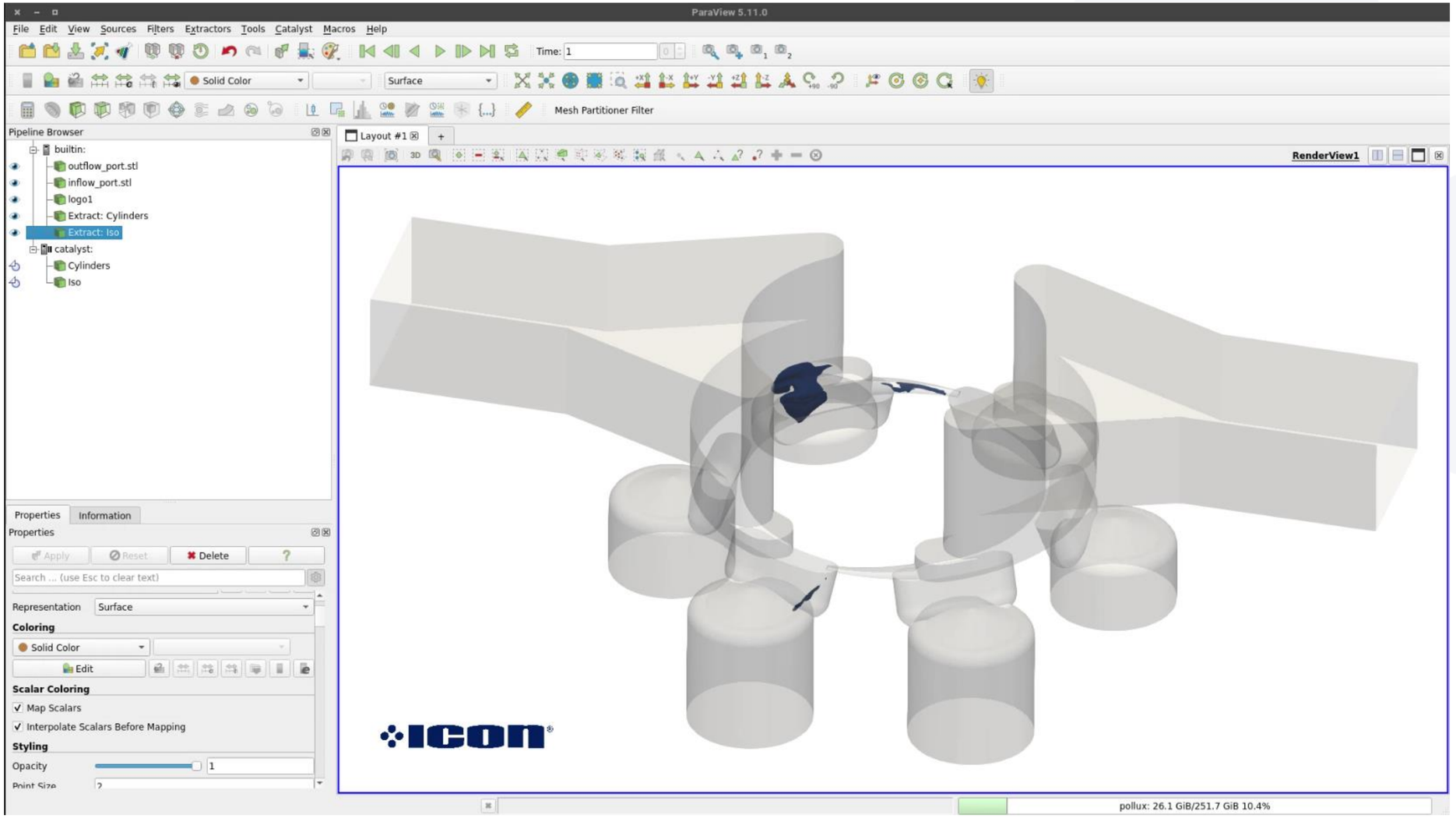
Catalyst Demo

CHT Case



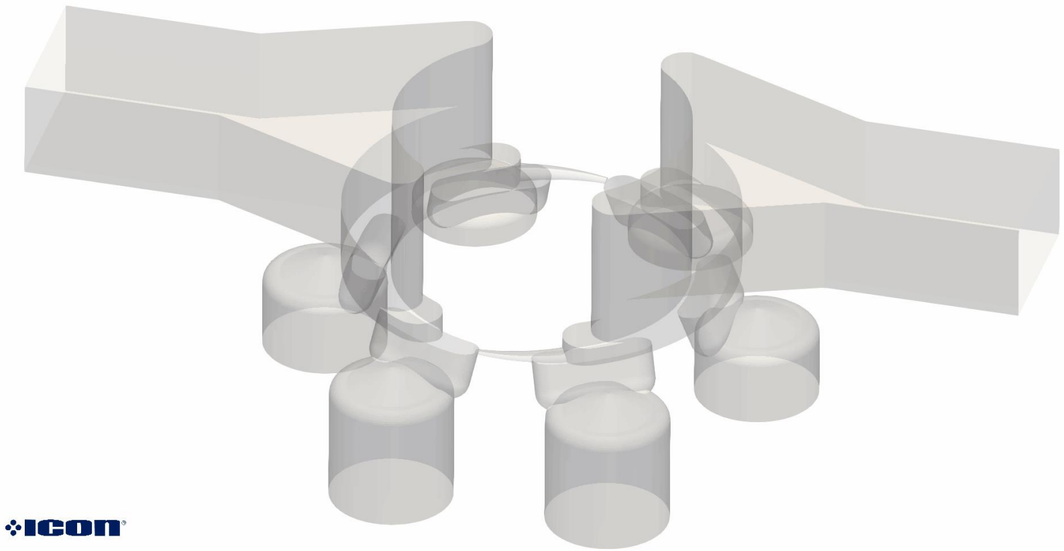
Catalyst Demo

Piston Pump



Summary and Outlook

- Description of the integration of Catalyst framework in iconCFD V5 has been provided
  - Catalyst function objection provides a straightforward interface to connect iconCFD simulation data to ParaView
- The fundamentals of how Catalyst can be used through illustrative examples has been demonstrated
- There is potential to significantly save disk space and post-processing time with Catalyst
  - There is a memory overhead for using Catalyst
- iconCFD Process V5 will support catalyst functionality in upcoming releases



www.iconcfd.com

contact@iconcfd.com

Legal Notices and Disclaimer

This presentation is supplied in confidence, and commercial security on its contents must be maintained. It must not be used for any purpose other than for which it is supplied, nor may any information be disclosed to unauthorised persons. It must not be redistributed or reproduced in whole or in part without permission in writing from ICON.

print green

Please consider the environment before printing this presentation. We recommend ‘Greyscale‘ as printing color settings.