Ascent: Flyweight In Situ Visualization and Analysis for HPC Simulations

ATPESC 2020

Monday August 3\textsuperscript{rd}, 2020

Matt Larsen (LLNL) + Cyrus Harrison (LLNL)
Ascent is an easy to use flyweight in situ visualization and analysis library for HPC simulations

- **Easy to use in-memory visualization and analysis**
  - Use cases: *Making Pictures, Transforming Data*, and *Capturing Data*
  - Young effort, yet already supports most common visualization operations
  - Provides a simple infrastructure to integrate custom analysis
  - Provides C++, C, Python, and Fortran APIs

- **Uses a flyweight design targeted at next-generation HPC platforms**
  - Efficient distributed-memory (MPI) and many-core (CUDA or OpenMP) execution
    - Demonstrated scaling: In situ filtering and ray tracing across **16,384 GPUs** on LLNL's Sierra Cluster
  - Has lower memory requirements than current tools
  - Requires less dependencies than current tools (ex: no OpenGL)
    - Builds with Spack [https://spack.io/](https://spack.io/)

Visualizations created using Ascent

Extracts supported by Ascent

[http://ascent-dav.org](http://ascent-dav.org)
[https://github.com/Alpine-DAV/ascent](https://github.com/Alpine-DAV/ascent)

Website and GitHub Repo
Ascent is ready for common visualization use cases
Ascent tutorial examples are outlined in our documentation and included ready to run in Ascent installs

http://ascent-dav.org
Ascent tutorial examples are outlined in our documentation and included ready to run in Ascent installs

- [http://ascent-dav.org](http://ascent-dav.org)
- Click on “Tutorial”
Ascent’s interface provides five composable building blocks

- **Scenes**
  (Render Pictures)

- **Pipelines**
  (Transform Data)

- **Extracts**
  (Capture Data)

- **Queries**
  (Ask Questions)

- **Triggers**
  (Adapt Actions)
Ascent’s Jupyter Extract provides a path to connect your simulation to a Jupyter Notebook

With the *Jupyter Extract*, users of any simulation code with Ascent integrated can run Jupyter Notebooks and use Python to interact with in-memory data.
This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy’s Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation’s exascale computing imperative.