VisIt

Current Targets

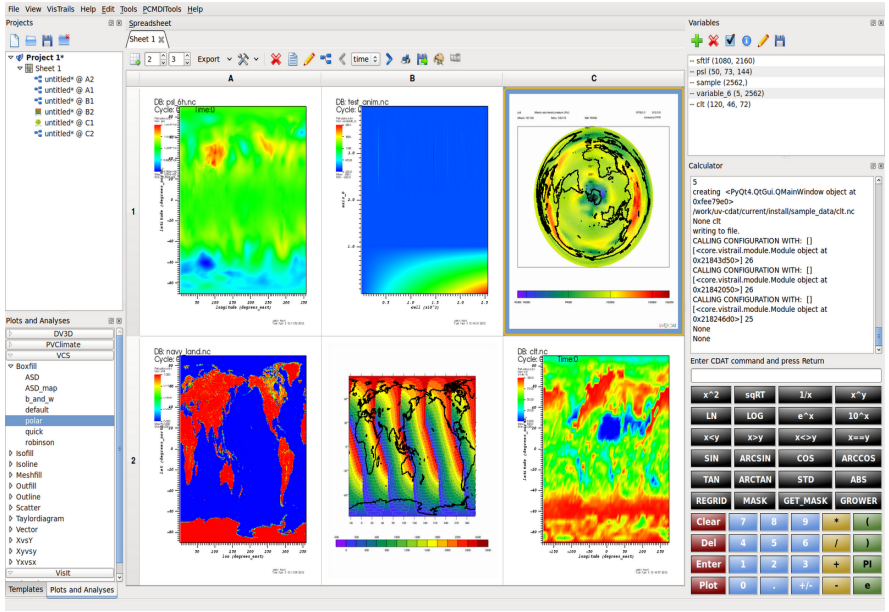
- Novel interface and easy communication between UV-CDAT and VisIt
- Provenance
- Provide Climate Science driven algorithms
  - Extreme Value Analysis
  - Peaks Over Threshold
  - Model-Based Clustering, …
  - TECA
    - Extra Tropical Cyclone detection
    - Atmospheric River Detection, ….

Climate Science Core Needs

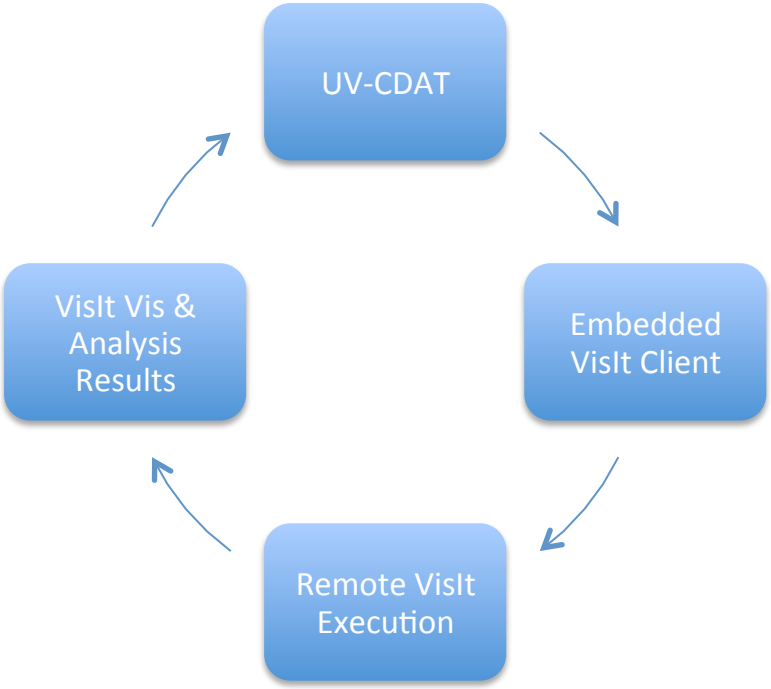
- Clean: customized embedded interface caters to climate science community.
- Scalable: Optimized analysis and visualization techniques.
- Diversifiable: Meet needs to varying capabilities and support of wide range of sources
  - R and Python scripting with MPI support.
- Customizable: Vary metrics of core algorithms within UV-CDAT/VisIt.
  - Visit_foreach_location, VisIt_foreach_time, VisIt_foreach_file, etc…

Addressing: Interface Needs

- Goal: Allow new UV-CDAT/VisIt capabilities to be intuitive and easy to use for the climate science community.



Integrated VisIt Client within UV-CDAT



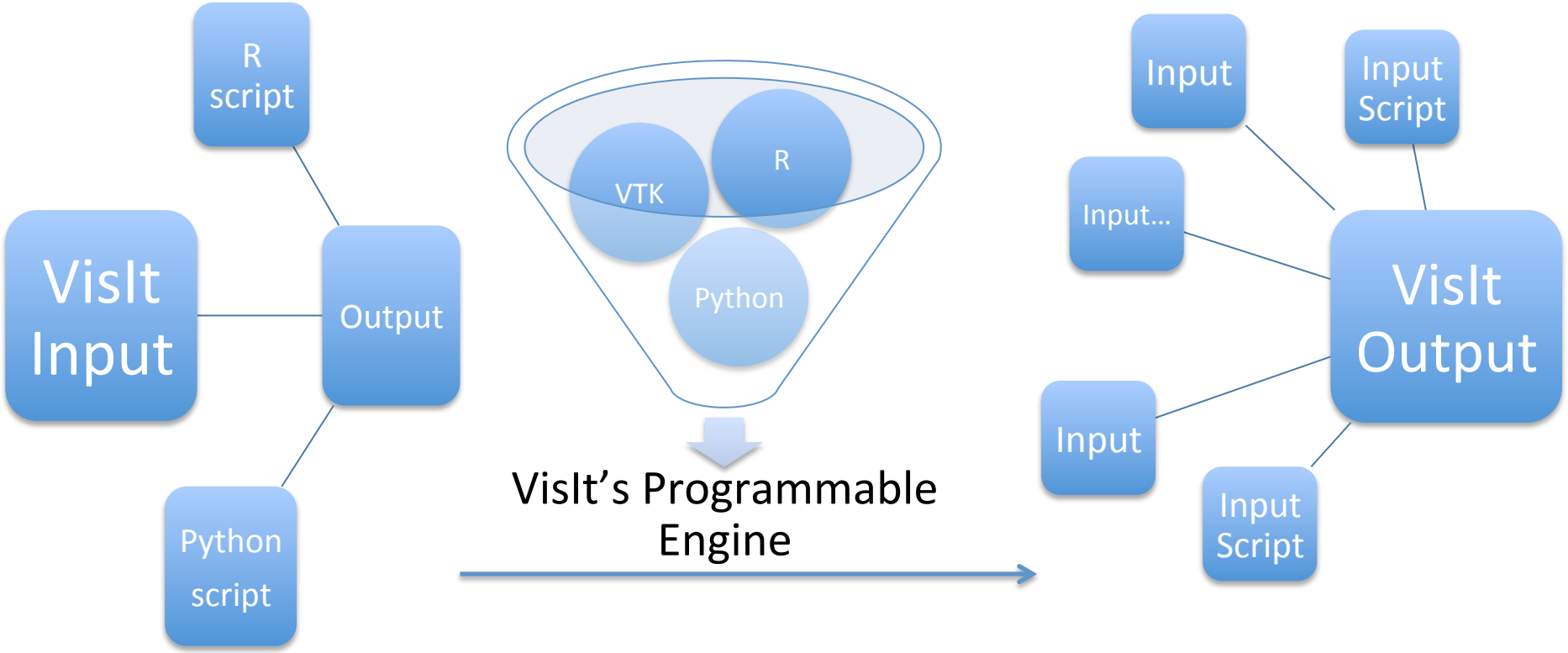
UV-CDAT & VisIt's Control network

Addressing: Interface Needs

- Loose integration with UV-CDAT
  - Rendering done within UV-CDAT, Computation done externally using local or remote resources.
- The PyQt infrastructure allows embedding, custom climate science interface, and seamless integration within UV-CDAT environment
- The VisIt embedded client provides request and response capabilities.

Addressing: Computational Needs

- Goal: Provide ability to extend climate science algorithms as well as create brand new ones with support of well supported programming environments (Python/R).



Addressing: Computational Needs (1)

- Write Python or R code and embed it into VisIt's pipeline for extension of functionality.
- Invoke optimized VisIt templates that utilize user generated execution kernels. Templates proved I/O, spatial, temporal, and windowing support.
- Create a call graph that links several kernels together allowing for rich & complex operations.
- Support for MPI in R and Python allowing users to parallelize kernels. (capable of parallelizing cdms)

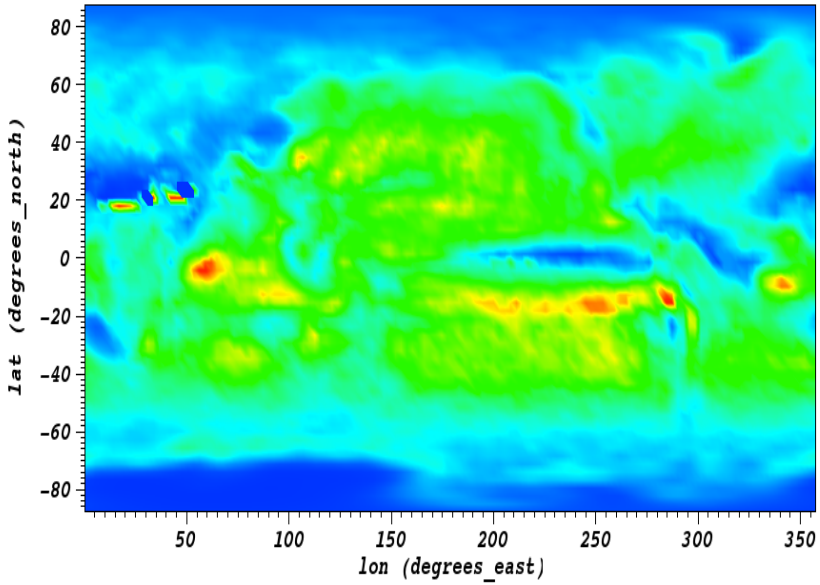
Computational Needs and Use Case

- Integrating – TECA and R Code
- Parallelized and optimized for scalability and performance.

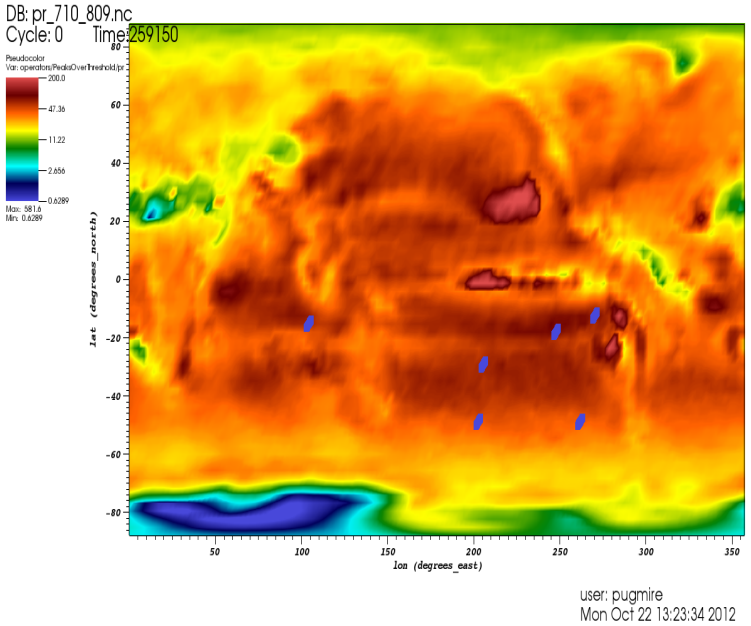
Scaling study

num CPUs	Annual Analysis	Monthly Analysis
1	600.53	7,904.97
2	320.95	5,954.16
4	181.40	4,948.32
8	112.20	2,589.61
16	67.49	2,121.38
32	34.36	1,317.06
64	19.51	673.61
128	10.33	558.18
256	5.99	576.66
512	5.10	317.34
1024	4.59	166.41

Extreme Value Analysis



Peaks Over Threshold



Use Case: Parallel Analysis of the Extremes

- Goal: Compute Extreme Daily Precipitation and Temperature.
  - Allow user to estimate the frequency and trends of extremes at many spatial locations.
  - Done in parallel within VisIt's workflow. I/O and preprocessing in VisIt and analysis in R. The results are displayed within UV-CDAT/VisIt or written to a file.
- Impact: Ability to analyze extremes in climate model output.
  - Models produce data at many locations, over long periods of time, and often for multiple runs (ensembles)
  - Use of UV-CDAT/VisIt provides faster processing, analysis of ensembles, processing over large periods of time, and over many models.

Conclusion: Work allows analysts to understand how models characterize extremes, compare between different models, and evaluate results and performance metrics to observational data.