Allen Sanderson - Univ. Of Utah SCI Inst.

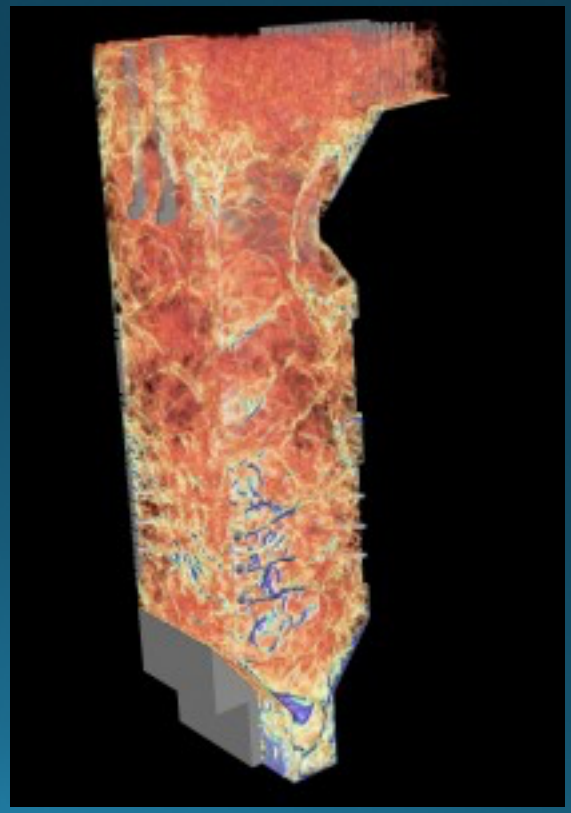
VisIt and Uintah an In-situ Marriage

2016 DOE Computer Graphics Forum

Uintah

- The Uintah software suite is a set of libraries and applications for simulating and analyzing complex chemical and physical reactions.
- http://www.uintah.utah.edu/projects.html

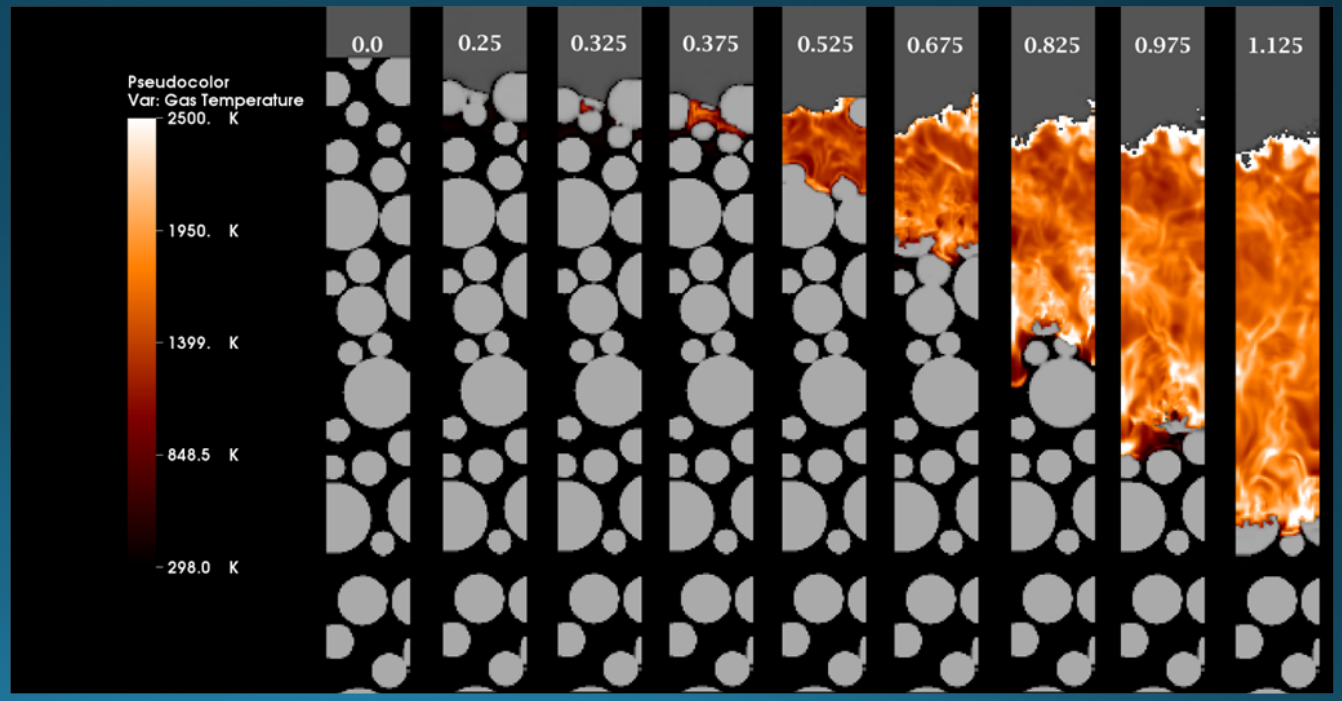
- **The Carbon-Capture Multidisciplinary Simulation Center –** exa-scale computing with V&V/UQ to more rapidly deploy a new technology for providing low cost, low emission electric power generation



Uintah

- The Uintah software suite is a set of libraries and applications for simulating and analyzing complex chemical and physical reactions.
- http://www.uintah.utah.edu/projects.html

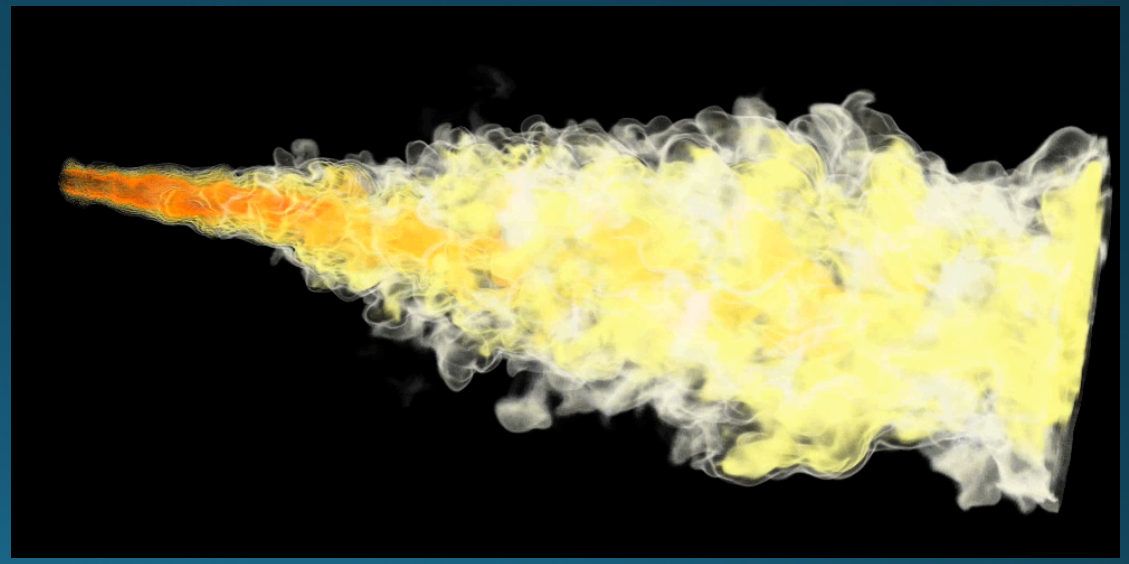
- **Multiscale Multidisciplinary Modeling of Electronic Materials Collaborative Research Alliance**



Uintah

- The Uintah software suite is a set of libraries and applications for simulating and analyzing complex chemical and physical reactions.
- http://www.uintah.utah.edu/projects.html

- **Clean Energy from Fossil Fuels –** model various energy technologies from traditional air-fired coal, oxy-fired coal/natural gas, fluidized bed coal combustion and coal gasification to more exotic coal technologies such as chemical looping and under ground thermal treatment



VisIt – libsim

- Description of VisIt's libsim by Brad Whitlock this afternoon.
  - Allows connection between the application and VisIt
  - Uses a middle layer to move application data to VisIt's VTK.

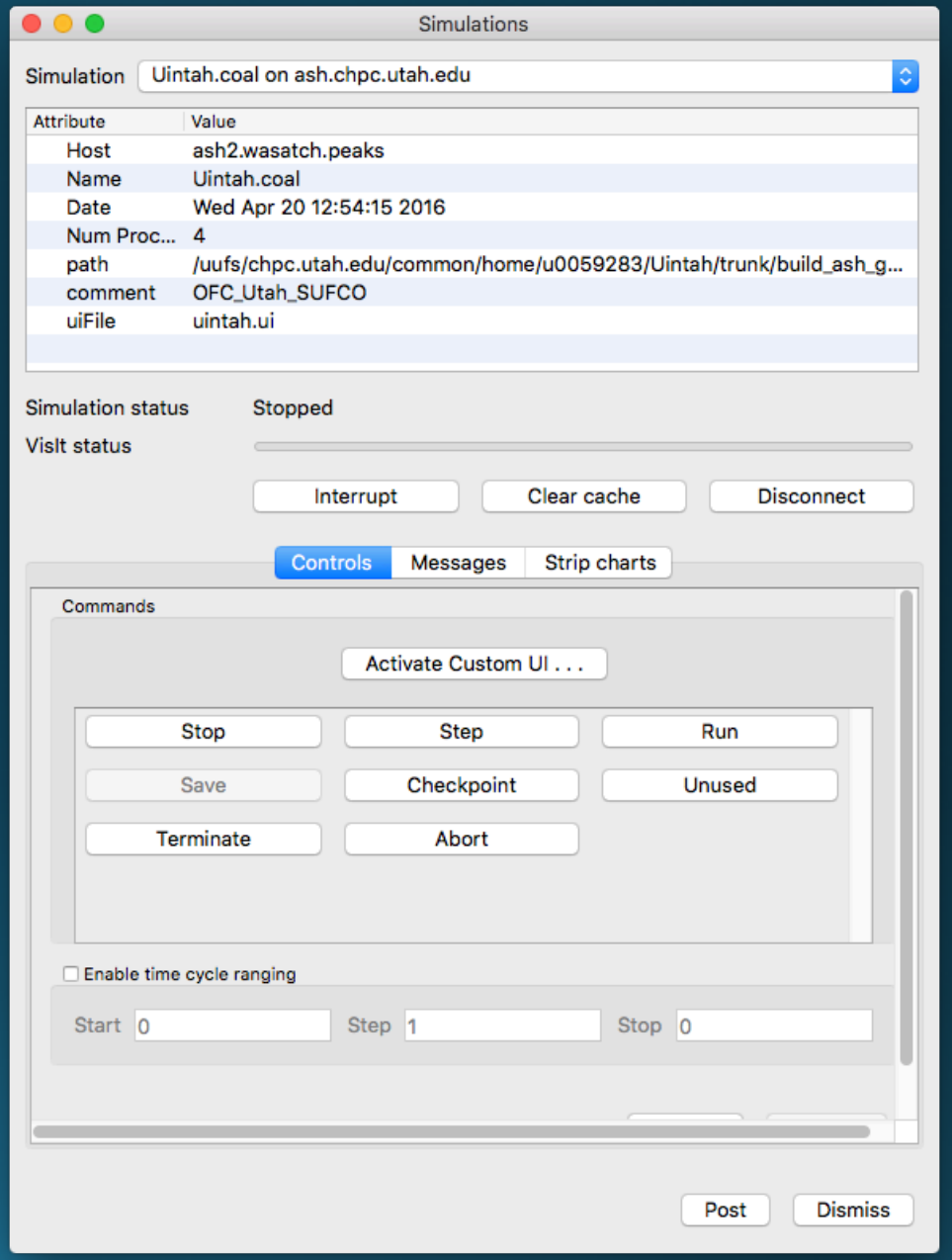
- Missing bits
  - Better connection to the simulation variables and the runtime performance values.
  - Simulation variables are global to the simulation and can be of the following:
    - Time based (time step)
    - Mesh based (AMR)
    - Problem spec (initial values, solution order)
    - Operational (output and dump file frequency)
    - Runtime analysis variables
  - Runtime performance values (memory, I/O, task time, mpi timimg)

VisIt – libsim

- VisIt's libsim provides for a custom UI
  - Can be setup by the application at run time.
  - Qt based UI
    - Any Qt widgets may be used but VisIt must be able to handle the Signals and Slots.
      - Line Edits, Checkbox, Combobox, etc.
      - New addition – Tablebox
      - Better support for other widgets.
  - Custom UI becomes an application specific dashboard

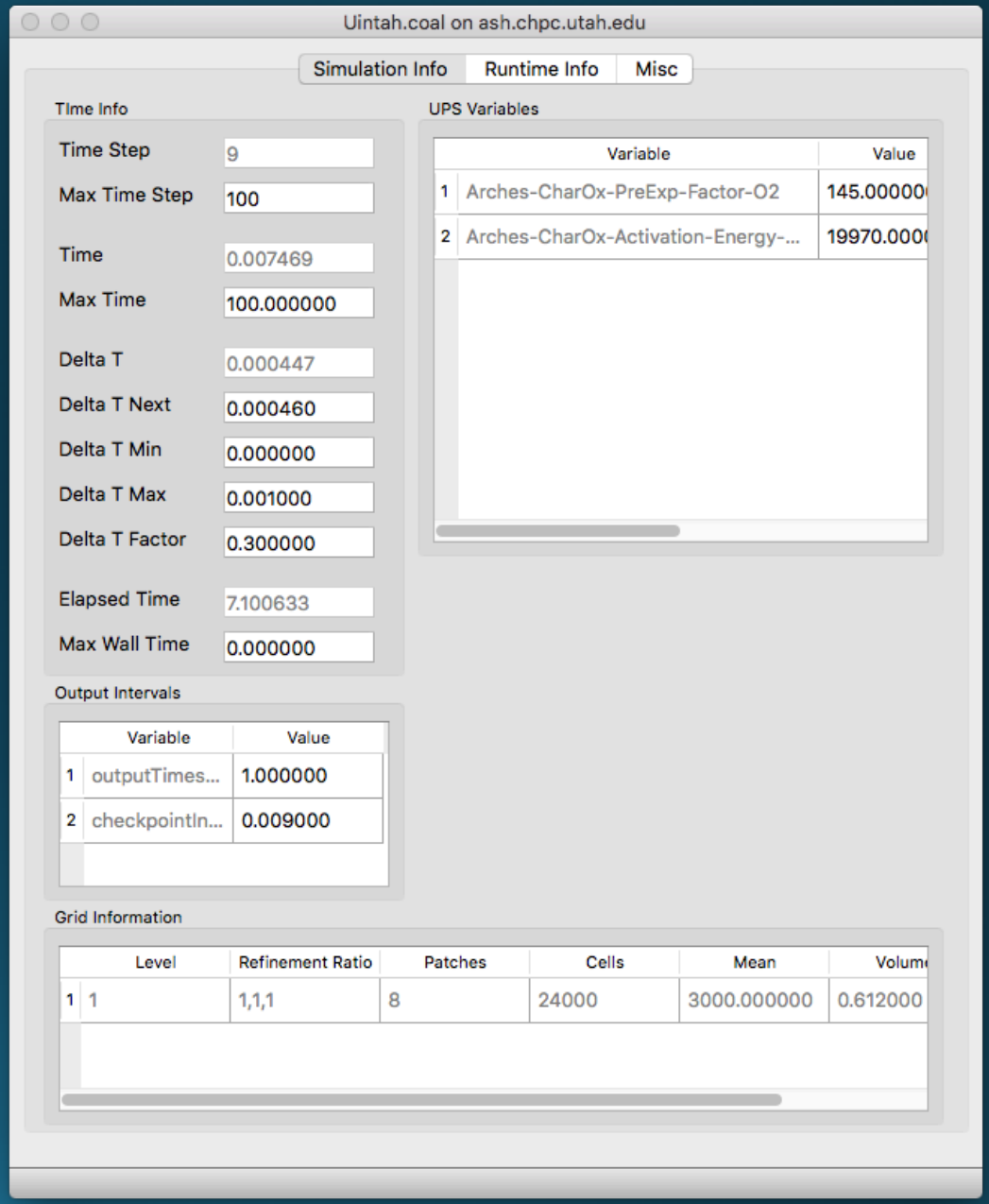
VisIt – libsim

- Simulation Window
  - Tabs are common to all simulations
    - Controls Tab
      - Commands are a unique to the application
      - Time cycle ranging now enabled.
    - Message Tab –
      - Reporting messages that might otherwise go application or error log files.
    - Strip Charts
      - Carry over from V1 libsim
      - Will be re-worked for V2



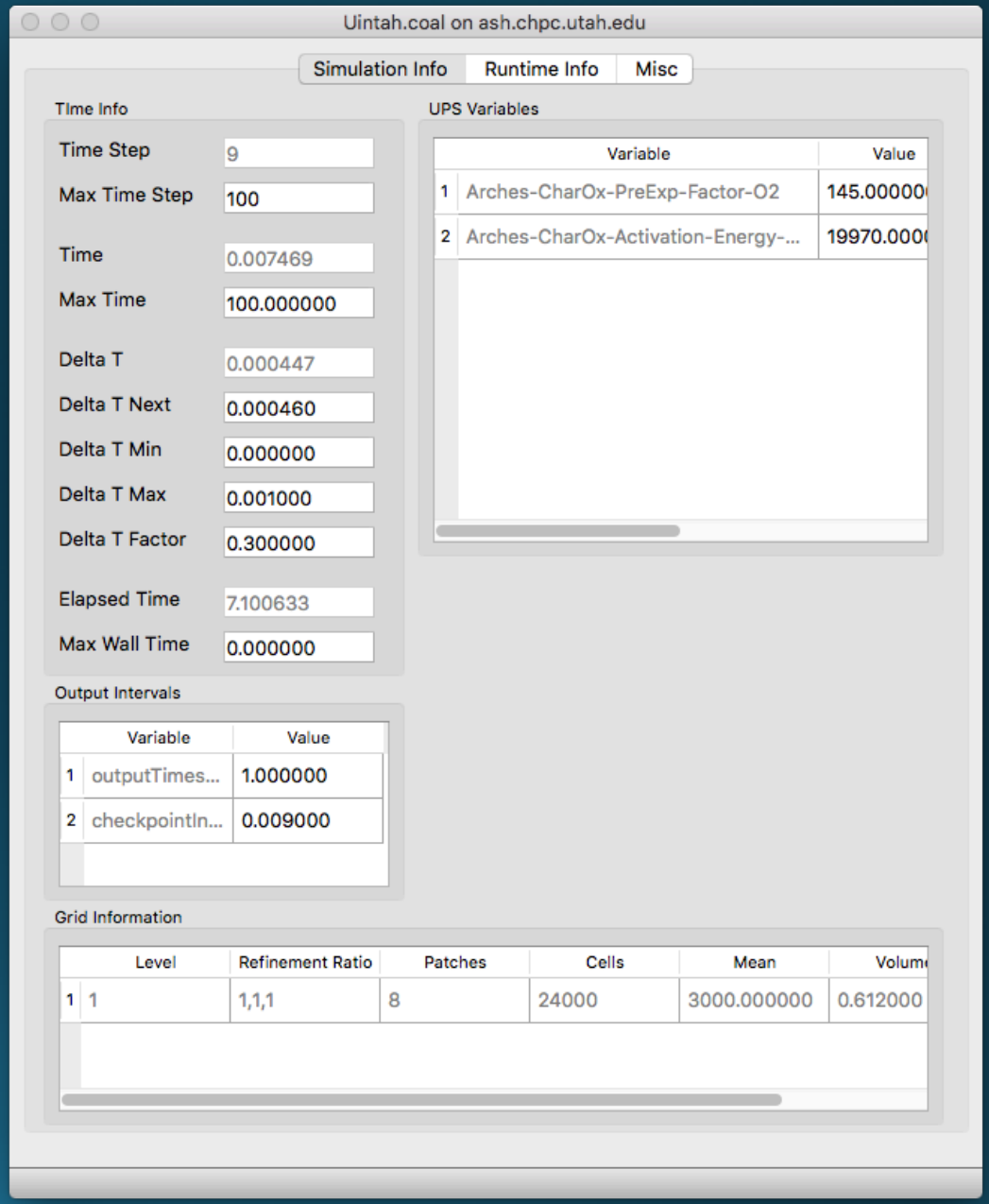
VisIt – Custom UI

- Simulation dashboard
  - Time info
  - Output parameters
  - Grid information
  - UPS Variables
    - (Uintah Problem Specification)



VisIt – Custom UI

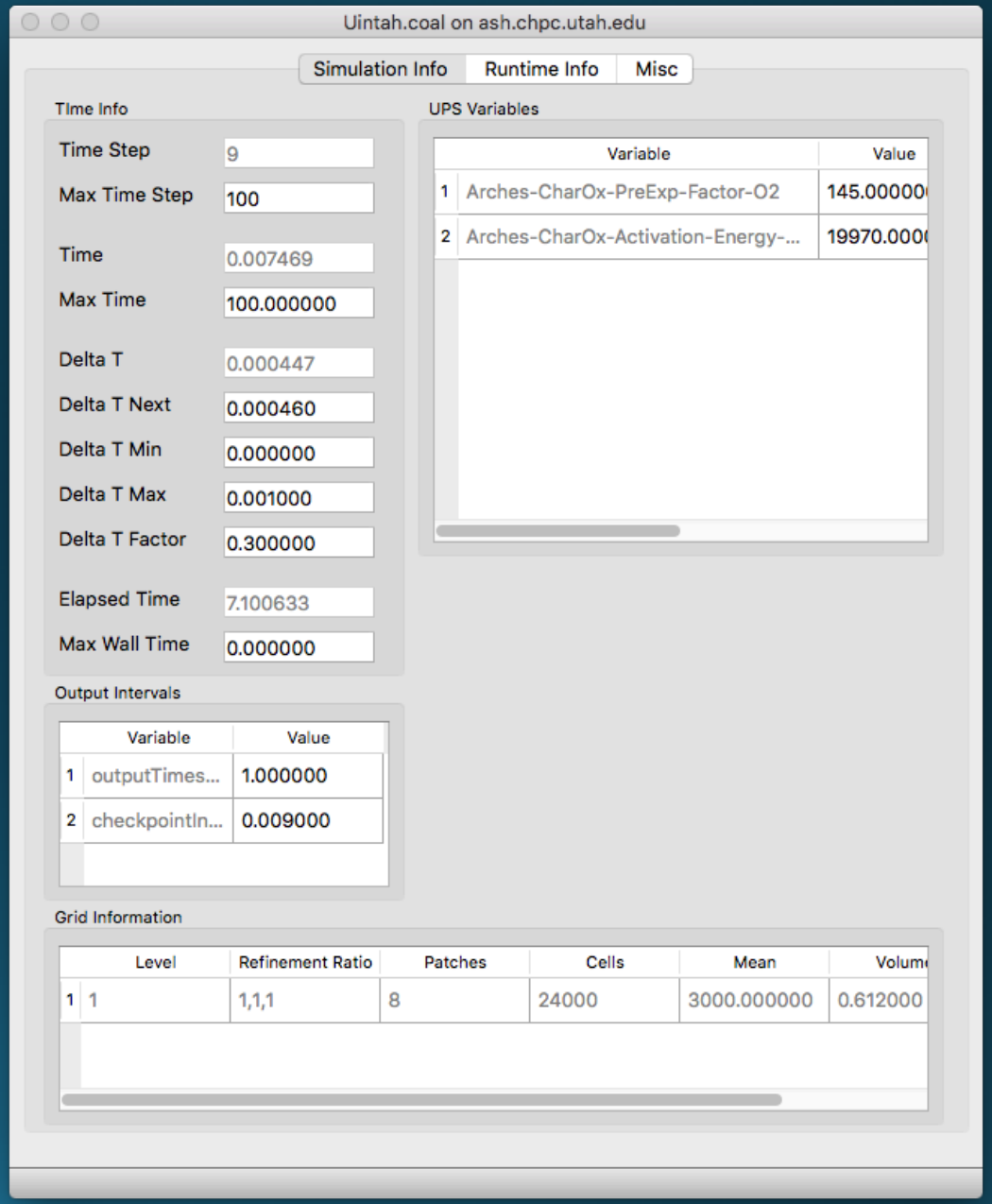
- Simulation dashboard
  - Individual vs table variables.
    - Individual variables require their own callback methods.
    - Tables can be modified on the fly (i.e. as the simulation runs variables can be added or removed)
  - Variables are directly linked to the application or via access methods.
    - Can do computational steering



VisIt – Custom UI

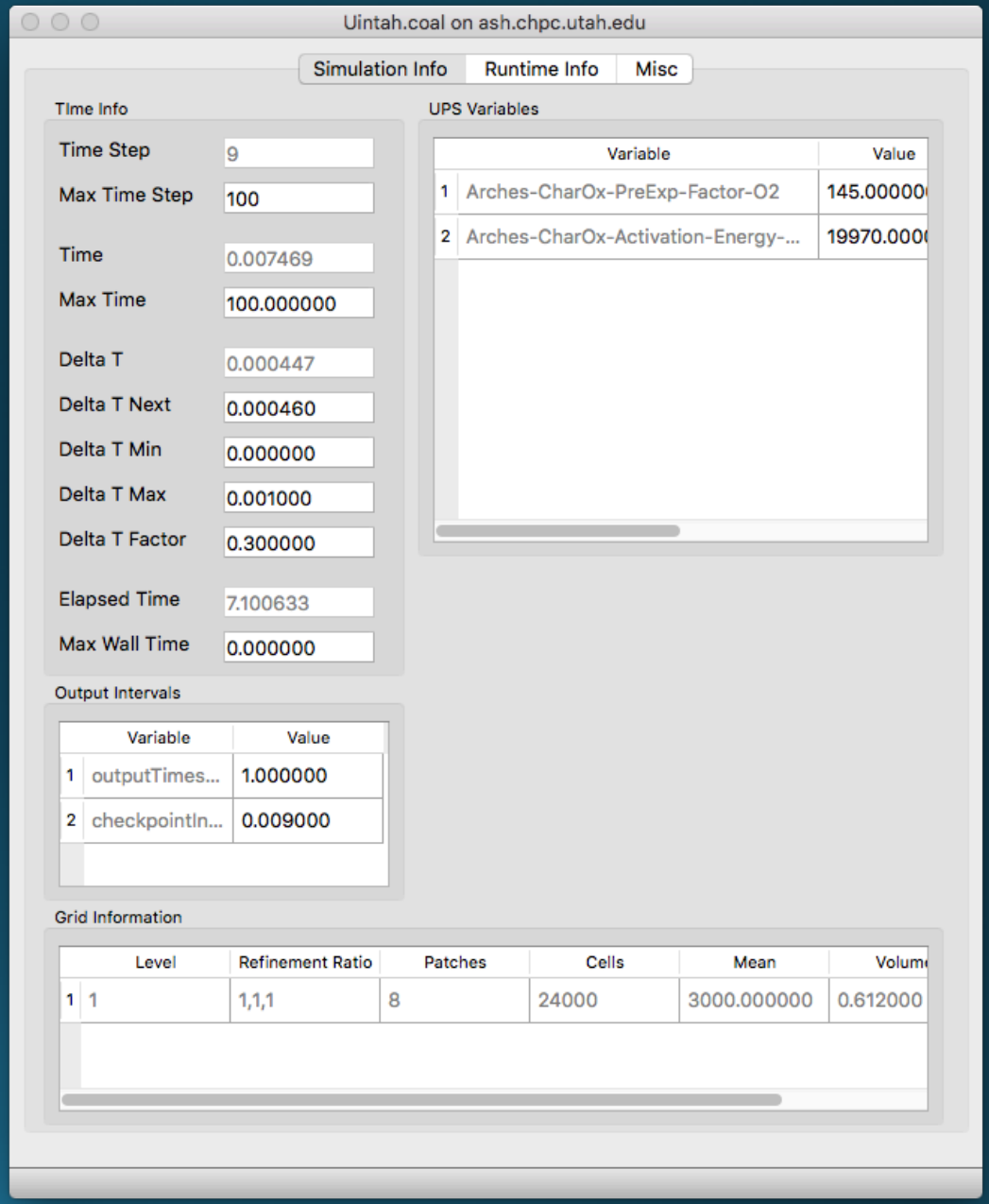
- Tables with generic variables uses a very lightweight structure to communicate between the application and libsim:

```
struct interactiveVar {
        std::string name;
        TypeDescription::Type type;
        int*    Ivalue;
        double* Dvalue;
        Vector* Vvalue;
        bool    modifiable; // If true the user may modify the value,
                            // otherwise it is read-only.
        bool    modified;   // If true the variable was modified by the
                            // user.
        bool    recompile;  // If true and the variable was modified force
                            // the task graph to be recompiled.
};
```



VisIt – Custom UI

- Application developer setups the structure at startup and inspects the structure before each time step execution.
  - Variable may require the task graph to be recompiled.
- Libsim callback is responsible updating pointer and modified flag.
- UPS Variables are generic across the application components.

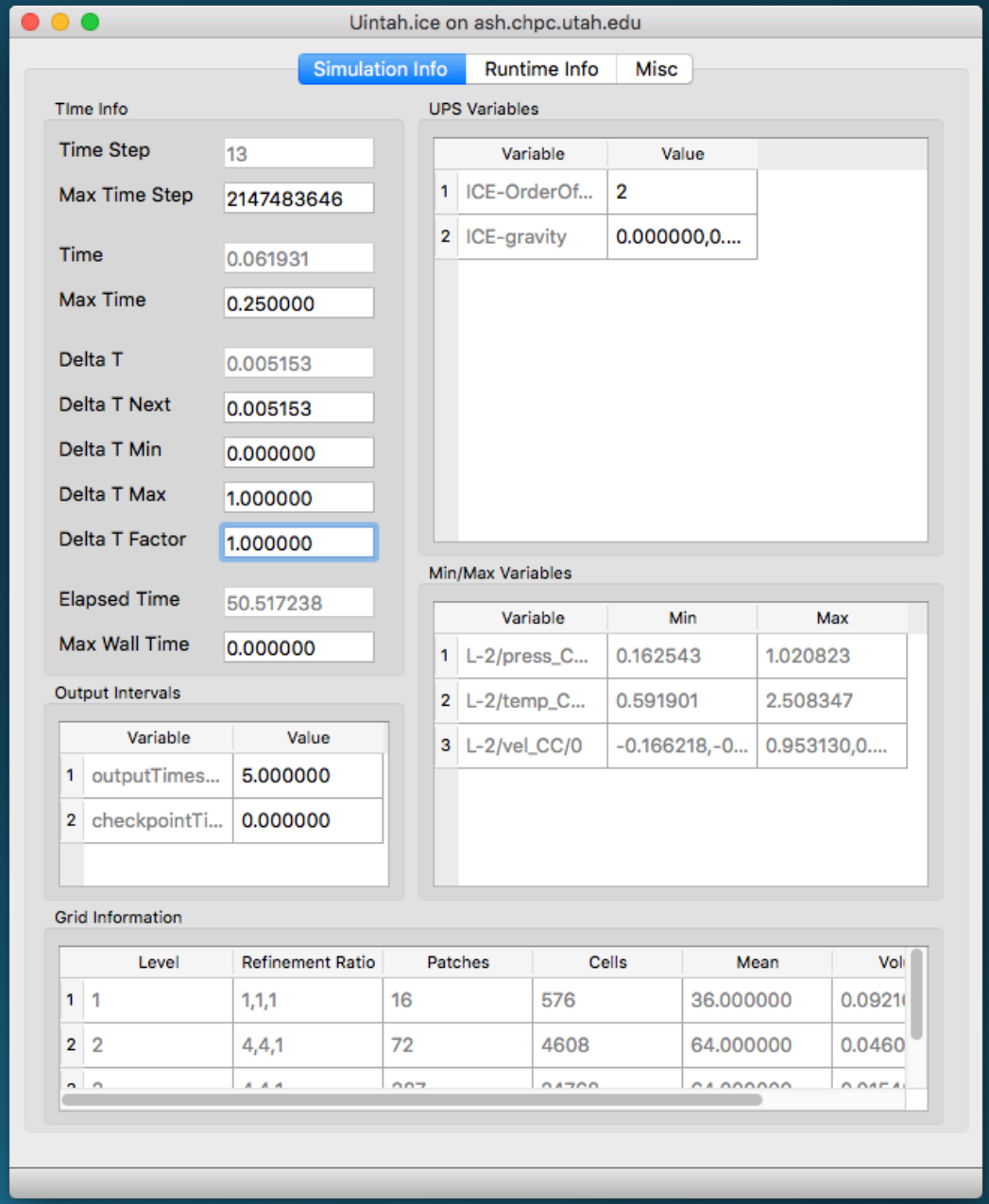


VisIt – Custom UI

- Tables with analysis variables uses a very lightweight structure to communicate between the application data archive and libsim:

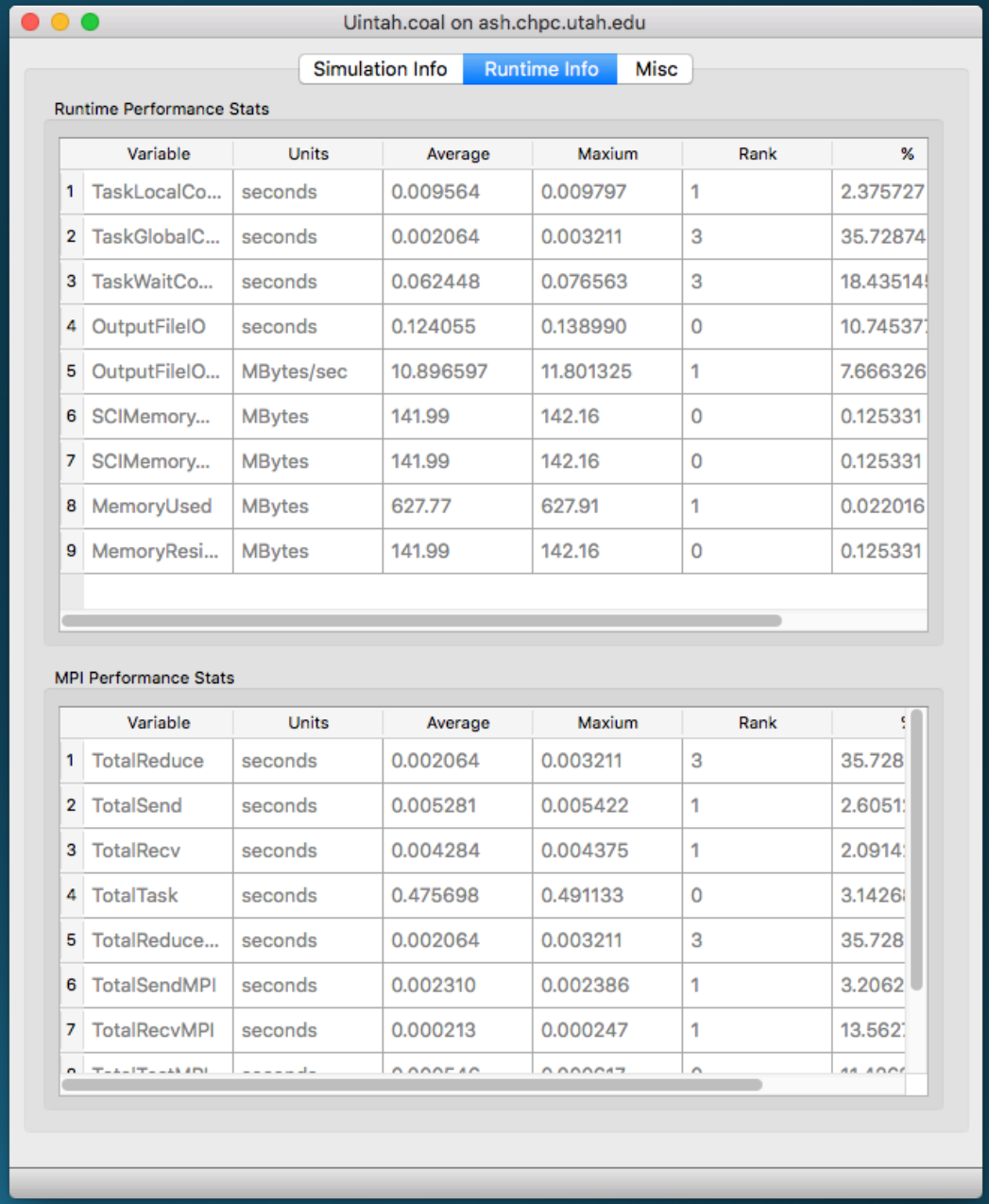
```
struct analysisVar {
        AnalysisType analysisType;
        VarLabel* label;
        VarLabel * reductionMinLabel;
        VarLabel * reductionMaxLabel;
        };
```

- Not yet generic to all analysis tools.
- Grid information – uses application grid information directly. AMR data is shared.



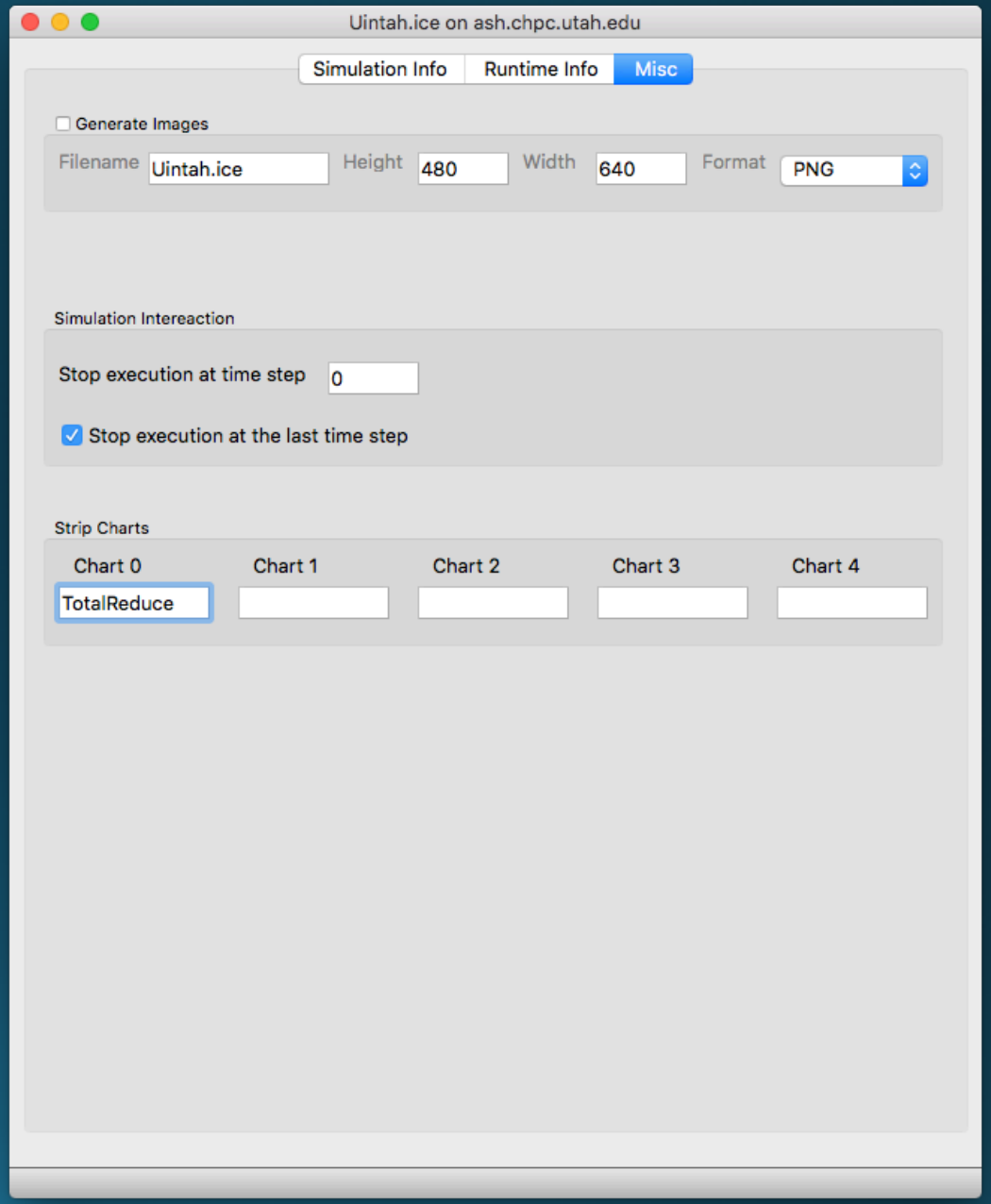
VisIt – Custom UI

- Runtime performance variables.
  - Uses a lightweight templated mapper class that holds:
    - The variable (a enumerated key value).
    - The value.
    - The variable name.
    - The variable units.
  - Application developer sees a std::map
    - Insert method is over ridden.
  - Derived type allows reductions for average value and maximal value with rank.
  - Mapper is shared with libsim.



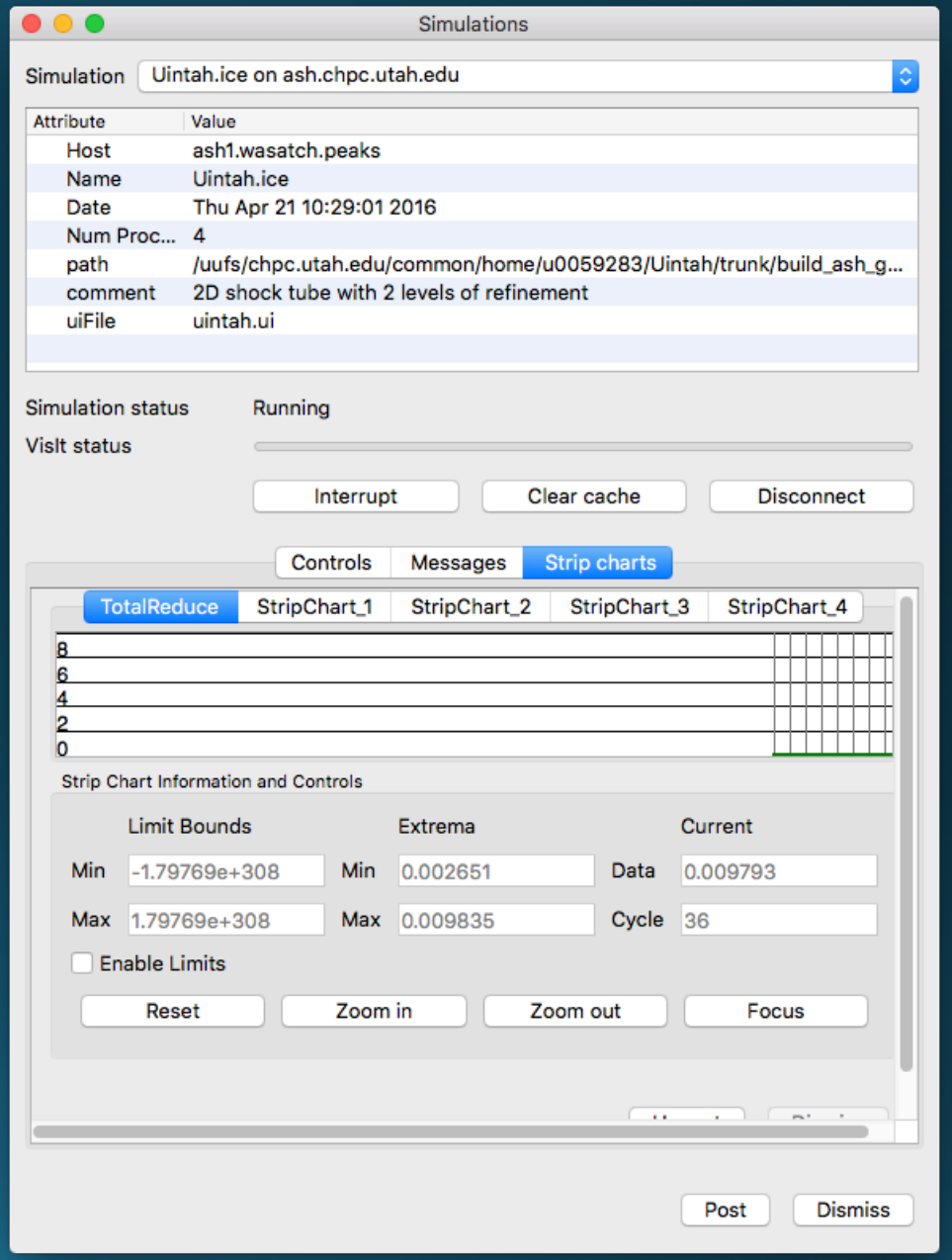
VisIt - Custom UI

- Nice to have features.
  - Generate image frames automatically
  - Stopping the execution at specific time steps – useful for debugging
  - Strips Charts



VisIt - Libsim UI

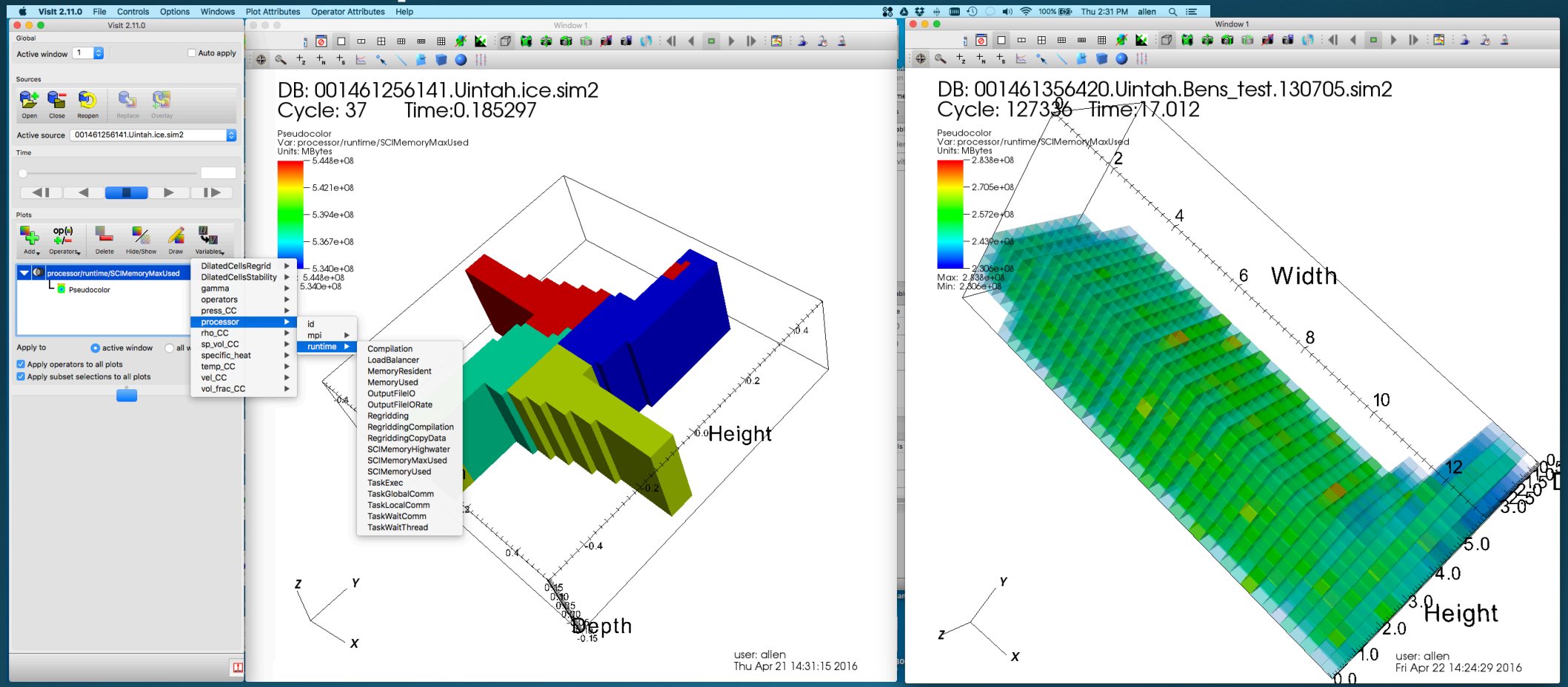
- Strip Charts
  - Carry over from V1 libsim
  - Will be re-worked for V2
  - More work to be done …



VisIt – Uintah Marriage

- Lightweight wrappers are key
  - Structures and methods that are on the application side and used by the application are preferred.
  - Qt Table very generic but not exactly the best UI.

Visualization of runtime performance values – processor based.



- Documentation for application developers:
  - http://uintah-build.sci.utah.edu/trac/wiki/VisitUintahInSitu