

Coven
a Framework for High Performance
Problem Solving Environments

Nathan DeBardeleben

Problem Solving Environments

- Historically there has been a gap between those who want to use computers (HPCs) and those who know how to use them.
- Help manage complexity of modern scientific computing
- Integral part of modern HPC
- Hides many details of the system, application, or both where applicable
- Flexible enough to solve the problem yet powerful enough to provide reasonably high performance
- Builds on work done by Dr. Stanzione for his Ph.D.

Where Are We?

- Robust, extensible GUI for dataflow graph creation
- Parallel runtime engine capable of performing data flow between modules on separate processors
- Scheduling system for scheduling execution of modules on separate processors and processes
- Full MPI support
- Profiling hooks
- Several prototype PSEs
- Still very developmental

Where Are We Going?

- ⇒ Coven programming language
- ⇒ Performance tuning and profiling
- ⇒ Type checking and automatic code generation tools
- ⇒ More prototype environments in distinct problem domains

Goals - by Supercomputing

- ⇒ Molecular dynamics demo / environment
- ⇒ Porting of old environments to new system
- ⇒ Front-end parser for user modules
- ⇒ Elementary type checking
- ⇒ Coven programming language

Goals - by Next Summer

- ⇒ Advanced type checking with automatic code generation
- ⇒ Profiling GUI tools
- ⇒ Profiling automatic tuning / suggestion tools
- ⇒ Changes to the system module structure to ease system module programming
- ⇒ Automatic converting of C functions to Coven modules

Who Are We?



- ⇒ Advisor: Dr. Walter B. Ligon III
- ⇒ Ph.D. Student: Nathan DeBardeleben
- ⇒ Masters Students: Sourabh Pandit
- ⇒ Undergraduate Students: Bryce Howard