
Jeffrey C. Carver
George K. Thiruvathukal
Conceptualization of a Software Institute for High Energy Physics

• High-energy Physics: Software & Computing enables our science

• Big challenges ahead for the High-Luminosity (HL-)LHC era
  • x10 projected shortfall of CPU & storage

• Advances in hardware will not get us there → need advances in Software!

• Community process → Strategic Plan for a HEP Software Institute

Strategic Plan for a Scientific Software Innovation Institute ($S^2I^2$) for High Energy Physics

Peter Elmer (Princeton University)
Mark Neubauer (University of Illinois at Urbana-Champaign)
Michael D. Sokoloff (University of Cincinnati)

December 20, 2017

M. Neubauer
Supported by ACI-1558233

P. Elmer, M. Sokoloff
ACI-1558216, ACI-1558219
Automated synchronization and boundary condition application for the Cactus framework
Samuel Cupp, Steven Brandt, Peter Diener
Louisiana State University

- Cactus Framework is an open-source environment for numerically solving Cauchy problems in parallel

- Current ghost zone synchronization and boundary condition application requires non-trivial, manual scheduling by programmers

- PreSync project replaces old system with an automated scheme
  - Tracks region of validity for grid functions (interior or everywhere)
  - Schedules synchronization and boundary conditions as needed

- PreSync reduces burden on users and programmers to understand inner workings of the Cactus Framework

We are supported by NSF Grant #1550551.
SI2-SSI: Integrating Data with Complex Predictive Models under Uncertainty: An Extensible Software Framework for Large-Scale Bayesian Inversion

Software framework (Python/c++) for large-scale Bayesian inference
Easy to use for both users and algorithm developers
Combined capabilities of MUQ and hIPPYlib

Noisy Data + Uncertain Models → Informed Decisions

Software framework (Python/c++) for large-scale Bayesian inference
Easy to use for both users and algorithm developers
Combined capabilities of MUQ and hIPPYlib

This work was partially supported by National Science Foundation grants ACI-1550487, ACI-1550547, and ACI-1550593.
SI2-SSE: Improving Scikit-learn usability and automation

Andreas C. Müller, Columbia Data Science Institute

> 150 models

20,000 datasets

= Meta learning
Project Highlights

- Symbolic computation – computer algebra
- Expert Systems: toolboxes and libraries for domain scientists and educators
- Multiple domains: Differential geometry, Lie theory, general relativity and field theory, geometry of differential equations
- Vertically integrated, interdisciplinary curriculum development
SI2-SSE: Development of a Software Framework for Formalizing ForceField Atom-Typing for Molecular Simulation

Christopher R. Iacovella\textsuperscript{1}, Peter Volgyesi\textsuperscript{2} and Janos Sallai\textsuperscript{2}

\textsuperscript{1} Department of Chemical and Bimolecular Engineering, Vanderbilt University,
\textsuperscript{2} Institute for Software Integrated Systems, Vanderbilt University

• **Challenge:** Develop a general scheme to encode and apply forcefield parameter rules
  - Forcefields describe the way atoms and collections of atoms interact via a set of adjustable parameters
    - Can contain thousands of sets that are differentiated by the chemical context of an atom, e.g.:
      - number of bonds, identity of bonded neighbors, local-environment of bonded neighbors, etc.
    - Rules for usage are typically hard-coded into software as a deeply nested hierarchy with specific rule order
    - This approach can be difficult to debug, extend, and disseminate

• **Defining parameter usage via SMARTS and overrides**
  - Encode chemical context using the SMARTS language for defining molecular patterns
    \texttt{opls\_135 = [C;X4](C)(H)(H)H}
  - Set rule precedence via “overrides”
    \texttt{opls\_148 = [C;X4]([C;X3])(H)(H)H overrides=opls\_135}
  - Rules are both **human and machine readable** and can be tested for accuracy and completeness

• **Foyer: General Python library for applying forcefields**
  - Atom types assigned using matching patterns determined by performing a subgraph isomorphism on the system graph
  - Rules can be evaluated in any order
    - Uses a fixed point iterative scheme that creates white- and blacklists, rather than rigid hierarchy
  - Source code does not change when rules change
    - Allows for easier testing, validation, versioning and dissemination
SI2-SSI: Integrated Molecular Design Environment for Lubrication Systems (iMODELS)

Peter Cummings¹, Clare McCabe¹, Ákos Lédèczi¹, Gabor Karsai¹, Adri van Duin², Paul Kent³
¹ Vanderbilt University, ² Pennsylvania State University, and ³ Oak Ridge National Laboratory

• **Challenge:** Improved lubrication strategies required for devices with nanoscale separations
  - Molecular simulation can be used to understand lubrication at the molecular level
  - Use this to screen for relationships between chemistry and tribology (i.e., lubrication properties)

• **mBuild:** a Hierarchical, Component Based Molecule Builder written in Python
  - Construct complex systems from smaller, interchangeable pieces
  - Enable programmatic variation of chemistry, required for screening

• **metaMDS:** define parameter landscape for screening

```python
# Initialize simulation with a template and some metadata
sim = mds.Simulation(name='monolayer',
                     template=configure_run_script,
                     output_dir='output')

chain_lengths = [8, 12, 16, 20]
for length in chain_lengths:
    parameters = {'chain_length': length,  
                  'n_molecules': 100,  
                  'forcefield': 'OPLS-aa',  
                  'build_func': build_monolayer}

# Parameterize our simulation template
sim.parametrize(**parameters)
```

![Diagram](image)

- Identify relationships between chemistry and tribology
Advancing Analysis for HEP

Improved Performance
To reduce the time to scientific discovery and to enable more in-depth analyses, we are increasing the rate of access to ROOT data files. This includes streamlined access to simpler data types (uproot and BulkIO) and faster compression algorithms (LZ4 and ZSTD). These efforts have already provided factors-of-several improvements.

Bridging to Big Data
“Big Data” software in industry, such as the Spark and scientific Python ecosystems, both complement and reproduce functionality of HEP software developed. To provide more options and reduce maintenance burdens, DIANA is building bridges between HEP software and the Big Data ecosystems: Spark-ROOT to Spark and uproot/OAMap to Numpy, Numba, and Dask.

High Level Tools
We are therefore striving to present HEP analysis with higher-level interfaces. Scikit-HEP incorporates HEP techniques in Pythonic idioms, uproot provides access to ROOT data as Numpy and Pandas abstractions, and OAMap compiles object-centric user code into fast array operations.

Statistical Techniques
We are developing tools and methods for statistical analysis in HEP, including research for simulator-based inference (Carl), machine learning for particle physics (Scikit-Optimize), and software for efficient numerical computations.
A **Landlab**-built cellular automaton model of hillslope evolution

**Abstract:** This poster describes and explores a new continuous-time stochastic cellular automaton model of hillslope evolution. The software was written using Landlab, a Python package for rapidly creating and modifying 2D numerical models of various scales. The Grain Hill model provides a computational framework with which to study slope forms that arise from stochastic disturbance and rock weathering events. The model can reproduce observed slope forms at the correct scale. We seek to explain the forms and evolution of rocky hillslopes like these:

**Common slope forms include parabola-, planar, and cliff-rampart:**

- **We can interrogate the scaling of height and effective diffusivity**
- **When rock is present, behavior ranges from transport- to weathering-limited**
- **Transport- vs. weathering-limited behavior is reflected in scaling of gradient and fraction soil (regolith) cover**

**Example of granular dynamics simulation:**

Empting of a silo

For the Grain Hill model, we add rules for periodic soil disturbance and soil formation by weathering

**WEATHERING**

**DISTURBANCE**

When the hill is 100% soil, the height and form depend on the ratio of disturbance frequency to uplift rate

**For more about Landlab see:**

http://landlab.github.io

We start with a lattice-grain cellular model (Tucker et al., 2016): We can investigate the scaling of height and effective diffusivity

**With the right parameter mapping, model captures specific case studies, such as the convex-up slope in panel 3a...**

**... and the planar profile in panel 3b:**

**Example of scale-invariant behavior**

**Model captures influence of fractional soil cover on average weathering rate ...**

**What about cliff-rampart morphology?**

With "collapse rule": quasi-steady forms ...

Adding a rule for "blocks" allows us to capture mesas, hogbacks, and rocky ridges

**With the right parameter mapping, model captures specific case studies, such as the convex-up slope in panel 3a ...**

**... and the planar profile in panel 3b:**

**Acknowlegments:**

Landlab was supported by a U.S. Department of Energy ASCR ASC2100067. Hillslope evolution research was supported by EAR-1159975. Additional support was provided by the Community Surface Dynamics Modeling System (CSDMS), EAR-1206767.

**References:**


**AUTHORS/DISTRIBUTORS:**

Landlab was developed by G. E. Tucker and E. Istanbulluoglu. Hillslope evolution research was supported by EAR-1159975. Additional support was provided by the Community Surface Dynamics Modeling System (CSDMS), EAR-1206767.


ADDITIONAL COAUTHORS:

Gregory E. Tucker1,2, Scott W. McCoy3, Daniel E.J. Hobley4
1 - CIRES and Department of Geological Sciences, University of Colorado, Boulder
2 - Community Surface Dynamics Modeling System (CSDMS)
3 - Department of Geological Sciences and Engineering, University of Nevada, Reno
4 - School of Earth and Ocean Sciences, Cardiff University, Cardiff, UK

**For the Grain Hill model, we add rules for periodic soil disturbance and soil formation by weathering:**

![Weathering and Disturbance](image1.png)

**When the hill is 100% soil, the height and form depend on the ratio of disturbance frequency to uplift rate:**

![Height and Uplift Rate](image2.png)

**We can investigate the scaling of height and effective diffusivity:**

![Scaling of Height and Diffusivity](image3.png)

**With the right parameter mapping, model captures specific case studies, such as the convex-up slope in panel 3a ...**

![Concave-Up Slope](image4.png)

**... and the planar profile in panel 3b:**

![Planar Profile](image5.png)

**Example of scale-invariant behavior:**

![Scale-Invariant Behavior](image6.png)

**Model captures influence of fractional soil cover on average weathering rate ...**

![Soil Cover Influence](image7.png)

**What about cliff-rampart morphology?**

![Cliff-Rampart Morphology](image8.png)

**With "collapse rule": quasi-steady forms ...**

![Collapse Rule](image9.png)

**Adding a rule for "blocks" allows us to capture mesas, hogbacks, and rocky ridges:**

![Block Rule](image10.png)

**With the right parameter mapping, model captures specific case studies, such as the convex-up slope in panel 3a ...**

![Concave-Up Slope](image11.png)

**... and the planar profile in panel 3b:**

![Planar Profile](image12.png)

**Acknowlegments:**

Landlab was supported by a U.S. Department of Energy ASCR ASC2100067. Hillslope evolution research was supported by EAR-1159975. Additional support was provided by the Community Surface Dynamics Modeling System (CSDMS), EAR-1206767.

**References:**


Solving Polynomial Systems with PHCpack and phcpy

PHCpack is software for Polynomial Homotopy Continuation
phcpy is a new Python package, available at www.phcpack.org

use case from the phcpy tutorial:

reproduces J. Mech. Design paper
# CRESCAT

**A Computational Research Environment for Scientific Collaboration on Ancient Topics**

**PI: David Schloen, University of Chicago**

## Goals

- Support all 5 stages of data for multi-disciplinary collaborative research
- Automate data transfers and transformations from one stage to the next via high-level GUI
- Accommodate heterogeneity of data sources, types, and schemas while preserving the original ontologies
- Seamless scalability for data management and algorithmic analyses
- Ensure sustainability of software maintenance and technical support
- Test and document with complex use cases from:
  - Archaeology
  - Paleontology
  - Historical linguistics
  - Ancient economics
  - Population genetics
  - Paleoclimatology
- External curated data repositories via live links using their Web APIs
- Instruments and data files with support for many data types and file formats (2D images, 3D models, audio, video, geospatial, etc.)
- Manual entry with offline mode for field input and automated syncing of data when back online

## Acquisition

- Ontology-agnostic data warehouse stores both data and the ontologies inherent in the data
- XQuery DBMS optimized for hierarchies of atomic keyed data objects representing spatial, temporal, linguistic, and taxonomic relationships
- Automatic parsing of source data to populate the integrated warehouse

## Integration

- Complex queries use hierarchical taxonomies with semantic inheritance
- Statistical analysis and visualization via tightly integrated R server with data-aware console
- Geospatial mapping and analysis via ArcGIS Online and ESRI components

## Analysis

- REST API exposes published data as XML with XSLT stylesheets to render it as JSON/HTML
- Sample Web apps provided for various research domains, to be customized as needed
- OWL-RDF ontology specification documents the top-level (upper) ontology underlying the data warehouse
- Can export RDF triples conformant to the OWL ontology, preserving all distinctions and relationships in the data, for use in other graph databases

## Publication

- Sample Web apps provided for various research domains, to be customized as needed
- OWL-RDF ontology specification documents the top-level (upper) ontology underlying the data warehouse
- Can export RDF triples conformant to the OWL ontology, preserving all distinctions and relationships in the data, for use in other graph databases

## Archiving

- Linked Data Tier
- Web servers
- External databases

- External databases are accessed via their Web APIs (e.g., ArcGIS Online for maps, Zotero for bibliographies)

- Client Tier
- Java application client (OCHRE)
- RESTful Web service
- For acquisition, integration, and analysis of data
- For building and storing queries and analytical workflows
- For viewing, searching, annotating, and analyzing published data

- Middle Tier
- Mirrored publication database
- Tamino XML Server (Software AG)
- Java Server Extension Objects
- Data items have persistent URLs

- Core Data and Analysis Tier
- R server
- Complex queries and analytical workflows are saved in the data warehouse and triggered by users via client applications.

---

**Example Use Case**

Ancient Greek economy via network analysis of thousands of coin hoards

---

**Funded by NSF SI2-SSI award 1450455**
SciGaP Hosted Gateways

Cyberinfrastructure + Domain Sciences

Science-Centric User Interfaces

Automated Metadata Extractors

Community Engagement

App Catalog

Plug-in your allocations

Domain Software (SSE/SSI Projects)

Community at large

POWERED BY APACHE AIRAVATA

POWERED BY XSEDE

Extreme Science and Engineering Discovery Environment
Portable Workflow
Expression

Experiment
Management

Scalability
and Robustness

{ "command" : "mysim.exe -p " + x*2 + 
" input.txt > output." + x + ".txt", 
"outputs" : [ "output" + x + ".txt" ], 
"inputs" : [ 
"input.dat", 
"mysim.exe"
] 
} [ for x in range(1,100) ]

Table 2. U-Net Hyperparameter Search Spaces

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Kernels</td>
<td>16, 32, 64, 128</td>
</tr>
<tr>
<td>Kernel Size</td>
<td>1, 3, 5, 7, 9</td>
</tr>
<tr>
<td>Activations</td>
<td>sigmoid, tanh, relu, elu, PReLU, LeakyReLU, ThresholdedReLU</td>
</tr>
<tr>
<td>Initializers</td>
<td>zeros, ones, glorot_normal, he_normal</td>
</tr>
<tr>
<td>Regularizers</td>
<td>11, 12, 11, 12</td>
</tr>
<tr>
<td>Dropout Rate</td>
<td>uniform distribution over [0, 1]</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>uniform distribution over [10^{-4}, 1]</td>
</tr>
</tbody>
</table>

Graph: MPI MAKER and WQ-MAKER

Graph: Execute Time vs. Total Cores
NIMBLE: Programmable Statistical Modeling for Hierarchical/Graphical Models

What do we want to do with hierarchical models?

1. More and better MCMC
   - Many different samplers
   - Better adaptive algorithms

2. Numerical integration
   - Laplace approximation
   - Adaptive Gaussian quadrature
   - Hidden Markov models

3. Maximum likelihood estimation
   - Monte Carlo EM
   - Data cloning
   - Monte Carlo Newton-Raphson

4. Sequential Monte Carlo
   - Auxiliary Particle Filter
   - Ensemble Kalman Filter
   - Iterated Particle Filter

5. Normalizing constants
   - Importance sampling
   - Bridge sampling
   - Others

6. Model assessment
   - Bootstrapping
   - Calibrated posterior predictive checks
   - Cross-validation
   - Posterior r-weighting

7. Idea combinations
   - PF + MCMC
   - MCMC + Laplace/quadrature

NIMBLE Components

1. Domain-specific language (DSL) for statistical models
   - We adopt and extend the widely-used BUGS language

2. Domain-specific language embedded within R for model-generic algorithms

3. Code-generator (compiler) that generates C++ from the model and algorithms DSLs.
   - C++ objects are managed from R by dynamically-generated interface classes

4. Algorithm library (MCMC, SMC, etc.)

Core Team

Perry de Valpine (PI); UC Berkeley
Christopher Paciorek (co-PI); UC Berkeley
Daniel Turek; Williams College
Nicholas Michaud; UC Berkeley
Duncan Temple Lang; UC Davis

R

NIMBLE (Models)
NIMBLE (Algorithm Library)
NIMBLE (User Algorithms)

igraph
R packages
NIMBLE Compiler / R

NIMBLE Components

1. Domain-specific language (DSL) for statistical models
   - We adopt and extend the widely-used BUGS language

2. Domain-specific language embedded within R for model-generic algorithms

3. Code-generator (compiler) that generates C++ from the model and algorithms DSLs.
   - C++ objects are managed from R by dynamically-generated interface classes

4. Algorithm library (MCMC, SMC, etc.)

Core Team

Perry de Valpine (PI); UC Berkeley
Christopher Paciorek (co-PI); UC Berkeley
Daniel Turek; Williams College
Nicholas Michaud; UC Berkeley
Duncan Temple Lang; UC Davis

NIMBLE Components

1. Domain-specific language (DSL) for statistical models
   - We adopt and extend the widely-used BUGS language

2. Domain-specific language embedded within R for model-generic algorithms

3. Code-generator (compiler) that generates C++ from the model and algorithms DSLs.
   - C++ objects are managed from R by dynamically-generated interface classes

4. Algorithm library (MCMC, SMC, etc.)

Core Team

Perry de Valpine, Daniel Turek, Christopher J. Paciorek, Clifford Anderson-Bergman, Duncan Temple Lang & Rastislav Bodik

https://r-nimble.org

NSF ACI-1550488, DBI-1147230 (completed), DMS-1622444
Constrained Low Rank Approximation (CLRA) for Modeling Key Data Analytics problems of clustering, topic modeling, community detection, and hybrid clustering

Our current focus: Nonnegative Matrix/Tensor Factorization (NMF and NTF) and other Variants (e.g. Sparse NMF, SymNMF, and JointNMF)

Why CLRA such as NMF and NTF? Utilize advances in numerical linear algebra algorithms and software, Behavior of algorithm easier to understand and analyze, Facilitates design of MPI based algorithms for scalable solutions PPoPP’16, TKDE’18, PPoPP’18, IPDPS’18, JGO’18

Fast Alternating Updating NMF/NTF (FAUN) Framework:

SI2-SSE: High Performance Low Rank Approximation for Scalable Data Analytics
R. Kannan (ORNL), G. Ballard (WFU), B. Drake (GTRI), and H. Park (GAtech) https://github.com/ramkikannan/nmflibrary

Titan – Dense Matrix, Low Rank 50, 100 iterations, 12650 Nodes, 202500 Cores, 2.7 million x 2.7 million

Matrix Size | Algos | NMF Time (in Secs)
---|---|---
2.7 million x 2.7 million | MU | 554
| HALS | 197.75
| ANLS/BPP | 219.8
3.03 million x 3.03 million (72 TBs) | MU | 554
| HALS | 197.75
| ANLS/BPP | 219.8

NMF on 118 million Web-graph
Sparse Webbase – 1M Vertex
Dense real world-video
GraviT Distributed Ray Tracing Framework

ACI-1339863 (TACC)  ACI-1339840 (Oregon)  ACI-1339881 (Utah)

Bring photo-quality rendering to your large-data visualizations through ray tracing, and now integrated into the SI2 yt project!

OpenGL version – flat lighting, constant shadows, limited depth perception

Embree RT version with 'glass' planes – integrated, realistic material behavior

PIs: Paul Navrátil (TACC), Hank Childs (UO), Chuck Hansen (UU), Allen Malony (ParaTools)
SI2-SSE: Collaborative Research: Extending the Practicality and Scalability of LibMesh-Based Unstructured, Adaptive Finite Element Computations

Paul Bauman
DeepForge is an open source platform for deep learning designed for promoting reproducibility, simplicity and rapid development within diverse scientific domains.
Why you must visit our poster!

A modern dense linear algebra software stack

• BLIS: Framework for Rapid Instantiation of BLAS-like functionality
• libflame: LAPACK functionality
• TBLIS: A C++ tensor contraction library

Effective outreach

• Professional development for scientific software scientists
• Massive outreach through Massive Open Online Courses. (145,000 participants to date)
• Cultivation of external contributors
The glue project was founded in 2012, with funding from NASA’s James Webb Space Telescope (JWST) project. Initial contracts continue to support development of JWST-related astronomical functionality.

Beginning in 2017, glue has also been funded by the National Science Foundation (under 1549637, 1740607, 1740229): "Collaborative Research: A sustainable future for the glue multidimensional linked data visualization package: The goal of this project is to integrate these tools and services to create a robust, secure, and scalable data exploration platform that supports data aggregation, integration and distribution beyond its traditional strengths in Astronomy and Medicine, by expanding support to use and developer communities.

All glue code is Open Source, at github.com/glue-viz.

The goal of the NSF SSE funding is to expand glue’s functionality into domains beyond its traditional strengths in Astronomy and Medicine, by broadening both its user and developer communities.

All glue code is Open Source, at github.com/glue-viz.

"InfoVis" & "SciVis" TOGETHER

GIS compatible

Medical Imaging

New! Jupyter Lab functionality

User-defined "Dimensions"
Massively Parallel Solvers for Computational Fluid Dynamics on Block Structured Cartesian Grids
Jaber Hasbestan, Scott Aiton, Brenton Peck, Donna Calhoun, Inanc Senocak, Grady Wright*
https://github.com/GEM3D

Block Structured Cartesian AMR

Schur complement domain decomposition method for the Pressure Poisson equation on block structured grids

Solve: \[
\begin{align*}
\nabla^2 p &= f & \text{on } \Omega \\
n \cdot \nabla p &= 0 & \text{on } \partial \Omega
\end{align*}
\]

for \( p \) given \( f \),

Timing (solve)

<table>
<thead>
<tr>
<th>Refinement level</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red-black tree</td>
<td>3.3</td>
<td>21.14</td>
<td>69.75</td>
<td>145.87</td>
</tr>
<tr>
<td>Z-curve enhanced hash function</td>
<td>3.7</td>
<td>26.85</td>
<td>87.85</td>
<td>180.01</td>
</tr>
<tr>
<td>C++ STL default hash function</td>
<td>4.89</td>
<td>39.95</td>
<td>138.19</td>
<td>272.97</td>
</tr>
</tbody>
</table>

Highly scalable red-black binarized-octree for generating and managing the adaptively refined grids
Automated Detection and Repair of Errors in Event-Driven Applications

Frank Tip

College of Computer and Information Science, Northeastern University
www.franktip.org

- modern applications rely on event handling for, e.g., user input, network communication
- key operations: register event handlers, emit events, call-back to event handler
- programmer errors are common, and lead to hard-to-debug failures
  - e.g., event race errors depending on nondeterministic scheduling of event handlers
- research goal: provide programmers with better tools to detect and repair such errors
  - based on static & dynamic program analysis
GeoVisuals Software: Capturing, Managing, and Utilizing GeoSpatial Multimedia Data for Collaborative Field Research

Ye Zhao
**Collaborative RAPID**

**BUILDING INFRASTRUCTURE TO PREVENT DISASTERS LIKE HURRICANE MARIA**

<table>
<thead>
<tr>
<th><strong>OBJECTIVE 01</strong></th>
<th><strong>OBJECTIVE 02</strong></th>
<th><strong>OBJECTIVE 03</strong></th>
<th><strong>Expected Science Outcomes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Water Quality Sampling Campaign</strong></td>
<td><strong>Data Archive</strong></td>
<td><strong>Cyberinfrastructure Advances</strong></td>
<td><strong>DISASTER:</strong> Contamination, drought, landslides, bio-diversity</td>
</tr>
<tr>
<td>Drinking water samples from public streams</td>
<td>Baseline assessment: Population Health Data, Healthcare Providers and supporting organizations, natural system environmental variables, Public Water System location and infrastructure status.</td>
<td>LANDLAB raster model grid and diverse data formats</td>
<td><strong>DRINKING WATER:</strong> Geographic location and use data</td>
</tr>
<tr>
<td>Spatially aggregated anonymized information of the impact zone</td>
<td>Hurricane Maria health and environmental data from public data repositories and Luquillo CZO instruments in El Yunque National Park</td>
<td>Observation Data Model (ODM2)</td>
<td><strong>HUMAN IMPACT:</strong> Spatial distribution of contamination or drought</td>
</tr>
<tr>
<td><strong>PUBLIC ACCESS INFORMATION</strong></td>
<td><strong>PRIVAICY PROTECTED INFORMATION</strong></td>
<td><strong>PRIVAICY PROTECTED INFORMATION</strong></td>
<td><strong>HUMAN IMPACT:</strong> Spatial distribution of contamination or drought</td>
</tr>
<tr>
<td>PRASA Utility, community operated tank system, household data</td>
<td>Water samples with personal information</td>
<td>Population health researcher user-testing</td>
<td></td>
</tr>
<tr>
<td>Teacher collection of student health data (IRB)</td>
<td>De-identified water samples that can be geo-located</td>
<td>Water quality professionals and researchers user testing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Individual data owners user testing</td>
<td></td>
</tr>
</tbody>
</table>
PAPI-EX
Performance Application Programming Interface for Extreme-scale Environments
SI2-SSI-1450122

- Performance Measurement Library
- Cross-platform
- Widely used in Supercomputing Environments
- Find Bottlenecks in your code!
- Measure raw performance, architectural effects (Cache, Branch Predictor, etc.), Power and Energy
- Supports most modern computing hardware
- Companion tools: PAPI-ex, Counter Inspection Toolkit

Jack Dongarra, Heike Jagode, Anthony Danalis
University of Tennessee

Vince Weaver
University of Maine
What is MATPOWER?

- Set of free, open-source, Matlab language tools
  - compatible with MATLAB® and GNU Octave
- For steady-state power system simulation and optimization, including:
  - power flow (PF)
  - extensible, optimal power flow (OPF)

MATPOWER’s Unique Combination

- free, open-source license (BSD)
- code that is easy to understand, customize
- state-of-the-art, high performance solvers
- ready-to-use realistic data included

MATPOWER boosted to de facto standard

- benchmark platform for power systems research
- educational tool for power systems engineers and optimization

Project Overview

Expand MATPOWER’s future impact as a successful research-enabling tool for the problems of the power systems of the future by providing the project infrastructure and core software architecture needed to facilitate ongoing community-supported growth.

MATPOWER Project Infrastructure

- Transition to fully collaborative open development paradigm with, public code repository, issue tracker, user and developer forums, contributor guidelines, public list of project descriptions

MATPOWER Core Software Architecture

- Redesign core software around a general modular architecture to enable more flexible user customization and facilitate significant user contributions, while retaining and enhancing the simplicity that makes it attractive in education
CRII: OAC: A Hybrid Finite Element and Molecular Dynamics Simulation Approach for Modeling Nanoparticle Transport in Human Vasculature

Ying Li
Summary

A software architecture challenge to design a unified set of high performance tree abstractions.
Modern programming languages provide atomic operations

Atomic operations:
- Make it possible to build faster, more scalable data structures with stronger guarantees
- Expose developers to complex behaviors that arise from CPU & compiler optimization
- Are extremely difficult to use correctly

C11Tester project is building tools to help developers effectively test code with atomic operations
Bringing higher end computational tools to the bench scientist to accelerate the discovery process.

Current grant coming to a close.

Working on transitioning to Community efforts and how to coordinate new efforts going forward.

Several new projects being proposed -- plus
- Networking/workshop grants
  - COST action proposal in Europe
  - looking at various US opportunities
  - Nurture involvement from major facilities

OTHER IDEAS WELCOME
The project includes (left) growth and characterization, (middle) iterative modeling, and (right) design training and validation. Single-frame red boxes represent experimental samples and data, while double-framed blue boxes represent computational products. The shaded region in the middle represents the application of particle swarm optimization.

The general flow can be understood as:

1. Growth of samples varied by composition and growth procedures.
2. Experimental structural characterization.
3. Iterative model simulation using characterization data.
4. ANN training to link simulation and growth parameters followed by predictive application of the ANN.

Future work: Use Particle Swarm Optimization to Find Simulation Parameters that Optimize Target Properties.
Real-time GW searches are plagued by “glitches”. E.g., GW170817 - a binary neutron star merger - had a delayed alert because we had to deal with data quality issues.

Goal: use machine learning to classify glitches in real-time in based on auxiliary information like seismometers, magnetometers, etc.

Currently can reject \( \frac{2}{3} \) of the glitches with a 1% false dismissal. Working to make it even better.
Overview

Motivation: Sophisticated and scalable workflows have become essential for advances in computational science. In spite of the many successes of workflow systems, there is an absence of a reasoning framework for end-users to determine which systems to use, when and why. Workflows are increasingly a manifestation of the algorithmic and methodological advances; workflow users and workflow system developers are often the same. Workflow systems must be easily extensible so as to support diverse functionality and the proverbial "last mile customization".

We advance the science of workflows and prevent workflow system "vendor lockins" by formulating a building blocks approach to middleware for workflow systems grounded on four design principles of self-sufficiency, interoperability, composability, and extensibility. A building block has: (i) one or more modules implementing functionalities to operate on a set of explicitly defined entities; and (ii) two well-defined and stable interfaces, one for input and one for output.

Properties of building blocks
- Self-sufficiency: design does not depend on the specificity of other building blocks
- Interoperability: can be used in diverse system architectures without semantic modifications
- Composability: its interfaces enable communication and coordination with other building blocks
- Extensibility: its functionalities and entities can be extended to support new requirements or capabilities

RADICAL-Cybertools: An implementation of the Building Block Approach to Middleware

RADICAL-Cybertools are designed and implemented in accordance with the building block approach, spanning four functional levels:

(L4) Workflow and Application Description: Requirements and semantics of applications and workflows.

(L3) Workload Management System (WLMS): Applications devoid of semantic context are expressed as workloads.

(L2) Task Runtime System (TRS): Execution of the tasks of a workload.

(L1) Resource: Capabilities, availability and interfaces required by the tasks to be executed.

RADICAL-Cybertools are used at each level to support scalable, efficient and effective use of high-performance and distributed computing.

(L2-L1) Interface to Resource

RADICAL-SAGA (Simple API for Grid Applications): Provides an interoperability layer that lowers the complexity of using distributed infrastructure whilst enhancing sustainability of distributed workflows, services, and tools in the form of a Python API. By abstracting away the heterogeneity of the underlying systems, RADICAL-SAGA simplifies access to many distributed cyberinfrastructure such as XSEDE and OSG.

(L2) Task Runtime Management

RADICAL-Pilot: Scalable pilot system for the simple and versatile execution of concurrent and distributed many-task applications on clusters, grids, and clouds. RADICAL-Pilot offers users a lightweight Python API to handle a variety of workflows—including MPI, multiprocess, multithreaded, CPU, and GPU tasks—and scheduling (O(10k) tasks while marshaling O(10k) distributed cores).

(L3) Workload Management

Ensemble Toolkit: Provides the ability to execute flexible combinations of ensemble-based applications on high-performance distributed computing resources. Ensemble Toolkit takes charge of where and how the workload is distributed: users only have to worry about what to run and when.

(L4) Applications and Scientific Workflows

ExTASY: Enables sampling of complex macromolecules with molecular dynamics. It supports high-performance and high-throughput execution of molecular dynamics calculations, and analysis tools that provide runtime control over a simulation.

HTBAC: Enables the scalable, adaptive and automated calculation of the binding free energy on high-performance computing resources.

Integration with existing systems

SeisFlows
- Supports seismic inversion workflows on HPC machines, at scale
- We integrated Seisflow with RADICAL-SAGA (L1) to execute compute jobs
- with RADICAL-EnTK (L3) to orchestrate tasks and data staging

Atlas (Panda and Harvester)
- PanDA is a WMS designed to support the distributed execution of workflows via pilots.
- Harvester is a service which provides pilot and workload management to Panda

We integrated Panda and RADICAL-Pilot to improve its scaling on large HPC resources, and integrated Harvester and RADICAL-Pilot to provide scalable task execution on HPC machines.

Swift
- Swift is a language and a runtime system to execute workflows
- We integrated Swift with RADICAL-WLMS (L3) to execute workflows concurrently on HPC and HTC resources.

Fireworks
- Fireworks is a system that enables material science workflows
- We integrate Fireworks and RADICAL-Pilot (L2) to improve its scaling on HPC resources
Learning Directed Acyclic Graphs (DAGs)

- Generally NP-hard
- Not scalable
- Stringent assumptions

Large Network Properties

- Polynomial \(n^3\) Alg.
- Scalable to large systems
- Relaxed assumptions

Large-Scale Causal Structure Learning

CDS&E: Statistical Methods for Discrete-Valued High-Dimensional Time Series

Ali Shojaie
SI2-SSE: ShareSafe: A Framework for Researchers and Data Owners to Help Facilitate Secure Graph Data Sharing

Raheem Beyah