Computing For Science: Experience and Practices from the RADICAL Lab

http://radical.rutgers.edu
".. your experiences in supporting science teams. In addition to a brief overview, we are especially interested in hearing about the best practices. How do you develop these best practices? How do you train new group members on those practices?

- Introduce RADICAL-Cybertools (RCT)
- Application Examples using RCT
- Experience and Lessons Learnt
- Best Practices:
  - Engagement
  - Performance Engineering
  - Software Development
Building blocks: integrate within existing software ecosystem

Designed for performance on HPC and in particular, leadership-class platforms

Implemented in Python

- **RADICAL-EnsembleToolkit (EnTK)**
  - Workflow engine
  - ‘Ensemble’ as first-order abstraction

- **RADICAL-Pilot (RP)**
  - Pilot system
  - Workload and task execution management

- **RADICAL-SAGA (RS)**
  - Batch-system interface
  - Interoperability layer

L4: Application Frameworks, Domain specific workflows
L3: Workflow and Workload management
L2: Task runtime system
L1: Resources
RADICAL Cybertools: Defining Features

• RCT are prototypes for advancing research in HPC, as well as tools to support S&E applications
  ○ Alignment between user requirements and research questions of interest
  ○ Not exclusive, but rich space of overlap!

• Top end of the Branscomb Pyramid
  ○ Scale is the reward; do things couldn’t do otherwise
  ○ “Making the impossible possible, and the barely possible, routine”

• Programming Model for Post-Dennard, Moore*
  ○ Ensembles, multitasks, workflows

• Heterogeneity
  ○ Type of task: executable/function
  ○ Type of resources used by tasks: nodes
  ○ Size and duration of tasks: 1-N cores/GPUs
  ○ Type of parallelism: thread/process, OpenMP, MPI

*“rumours of my demise have been greatly exaggerated”.
Combining AI with HPC: AI-driven MD simulations -- DeepDriveMD

- Weighted Ensemble MD simulations
- Deep Learning/Artificial Intelligence

"Interesting conformations", population sampled, and other features

Learning Everywhere
- Ana & Fax. In Visionary Talks: 15th International Conference eScience (2019), San Diego, California
- Ana & Fax. 15th International Conference eScience (2019), San Diego, California
IMPECCABLE: Integrated Modeling Pipeline

Ref. Aspuru-Guzik
Impacting SARS-CoV-2 Medical Therapeutics

Fig. 4. Conformational changes upon MCULE-5948770040 binding to M\textsuperscript{PRO} indicate changes within distinct regions, both close-to and farther-away from the primary binding site. (a) RMS fluctuations of the
Therapeutics: Needle in a Haystack?

● Scale of Operation:
  ○ $\sim 10^{11}$ Docking calculations
  ○ $\sim 10^3$ ML-driven MD calculations
  ○ $\sim 5 \times 10^4$ Binding Free Energy Calculations
  ○ $\sim 2.5 \times 10^6$ node-hours (~30 days, all Summit)

● Peak Performance
  ○ $\sim 8000$ nodes (Frontera, April. 2021)
  ○ ~ 4000 nodes on Summit
  ○ https://arxiv.org/abs/2103.00091

● DOE’s National Virtual Biotechnology Laboratory
  ○ Computational Infrastructure and Capabilities Beyond extend beyond COVID-19?

*US Secretary of Energy Honor Award (2021)*
Therapeutics: Needle in a Haystack?

**Scale of Operation:**
- $10^{11}$ Docking calculations
- $10^3$ ML-driven MD calculations
- $5 \times 10^4$ Binding Free Energy Calculations
- $2.5 \times 10^6$ node-hours (~30 days, all Summit)

**Peak Performance**
- ~8000 nodes (Frontera, April. 2021)
- ~4000 nodes on Summit

**Extensible Computational Infrastructure and Capabilities**
- Beyond COVID-19? Secretary Granholm during a town-hall to discuss DOE/NVBL contributions
Members of RADICAL Lab
Advancing science one computing abstraction at a time!

Shantenu Jha
(Lab Lead)

Matteo Turilli
(Assistant Research Professor)

Andre Merzky
(Senior Research Programmer)

Hyungro Lee
(Research Associate)

Mikhail Titov
(Research Associate)

Li Tan
(Scientist)

Aydin Saribudak
(PhD Student)

Aymen Alssadi
(PhD Student)

Seyma Guleryuz
(Research Assistant)
<table>
<thead>
<tr>
<th>Domain</th>
<th>Topic</th>
<th>Collaborator</th>
<th>Institution</th>
<th>HPC platforms</th>
<th>Application</th>
<th>Task Type</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biophysics</td>
<td>understanding molecular mechanisms of cell communication</td>
<td>Harel Weinstein</td>
<td>Weill Cornell Medicine</td>
<td>ORNL Summit (IBM Power9, NVIDIA V100)</td>
<td>OpenMM VMD</td>
<td>GPU based ensemble simulations</td>
<td>Python Bash shell</td>
</tr>
<tr>
<td>Biophysics</td>
<td>sampling simulation of conformational transitions</td>
<td>Abhishek Singharoy</td>
<td>Arizona State University</td>
<td>ORNL Summit XSEDE Bridges2 (AMD EPYC 7742, NVIDIA Tesla V100)</td>
<td>NAMD VMD MDFF</td>
<td>GPU/CPU mixed heterogeneous tasks</td>
<td>TCL/Tk C++</td>
</tr>
<tr>
<td>Geological Sciences</td>
<td>Combining 3D simulations of global seismic wave propagation</td>
<td>Jeroen Tromp Guido Cervone</td>
<td>Princeton Univ. Penn State Univ.</td>
<td>ORNL Summit Princeton Tiger (Intel Skylake, P100) NCAR Cheyenne (Intel Xeon Broadwell)</td>
<td>specfem3d</td>
<td>CPU based simulations</td>
<td>Python C++ R</td>
</tr>
<tr>
<td>Molecular biology</td>
<td>Pipelining ML and MD simulations</td>
<td>Rick Stevens, Arvind Ramanathan</td>
<td>Argonne National Lab</td>
<td>ORNL Summit TACC Frontera/Longhorn (Intel Cascade Lake/ IBM Power9, V100) LLNL Lassen (IBM Power9, V100)</td>
<td>OpenMM Pytorch</td>
<td>GPU/CPU mixed heterogeneous tasks</td>
<td>Python</td>
</tr>
<tr>
<td>Physical chemistry</td>
<td>Estimating physics based binding affinities</td>
<td>Peter Coveney</td>
<td>University College London</td>
<td>SuperMUC ORNL Summit TACC Frontera</td>
<td>OpenMM NAMD ESMACS/Ties</td>
<td>GPU/CPU mixed heterogeneous tasks</td>
<td>Python C++</td>
</tr>
</tbody>
</table>
RCT must provide a unique value proposition to end-users

We revolve around domain scientists
  ○ Crucial role of engagement, requirement engineering, user support, iterative development.

If you don’t have to do the repetitive things, you can innovate at higher levels:
  ○ SAGA: prevents end-user from plumbing for each machine
  ○ RADICAL-Pilot: consistent execution model for task-level parallelism
  ○ RADICAL-EnTK: pipelines as first-order abstraction for workflow specification

Role and importance of abstractions:
  ○ E.g., ensemble, pilot-job, pipelines

Some leadership class machines are easy to conquer, some more challenging:
  ○ Frontera vs Summit (early access to Summit was critical; 6-9 months lead time)

We are part of a community on which we rely and to which we contribute:
  ○ Building blocks approach to enable integration with 3rd party tools
  ○ Avoid: reinventing the wheel; end-to-end, monolithic framework and their lock-in
Best Practice: Engagement

Scale/perf., tools and HPC platforms require a **steep** learning curve; **no turnkey solutions** (Expectation Management)

- Presentation meetings:
  - Initial requirement elicitation
  - Existing and planned capabilities review
- Initial 1-on-1 ‘onboarding’:
  - Documentation (readthedocs)
  - Overview, tutorial and stock examples
- GitHub-centric user support:
  - Dedicated repository for each collaboration
  - Tickets implemented via GitHub issues
  - Discussion for feature requests
- Freq/fixed collaboration meeting with agenda and minutes
- Fine-grained support and collaboration:
  - Slack channels/private chats
Best Practice: Performance Engineering

- Performance engineering: Working closely with infrastructure teams and domain scientists to tune application and middleware
- Resource profiling to detect performance bottlenecks on the HPC platforms
- Application tracing and profiling without impacting middleware performance
- Separating tracing from profiling:
  - Online tracing at runtime via a dedicated subsystem
  - Offline profiling via dedicated library
- Stock libraries and tools for plotting and visualization
Best Practice: Software Development

- Writing and managing code:
  - Git-centric workflow (GitFlow)
  - Pull request for code reviews
  - Project-specific branches for rapid development and support
  - Taxonomy and statistical analysis of issues to drive development roadmap
  - Coding style: automatically checked (pep8)

- Testing:
  - Continuous integration
  - Github Actions unit tests
  - Code test coverage

- Distributing code:
  - Monthly releases on PyPI / Conda

- Supporting HPC platforms:
  - Early access
  - Tailored integration testing
Thank you