Argo AML: Explicit Management of Deep Memory for Exascale Applications

SciDAC Institute RAPIDS

Scientific Achievement

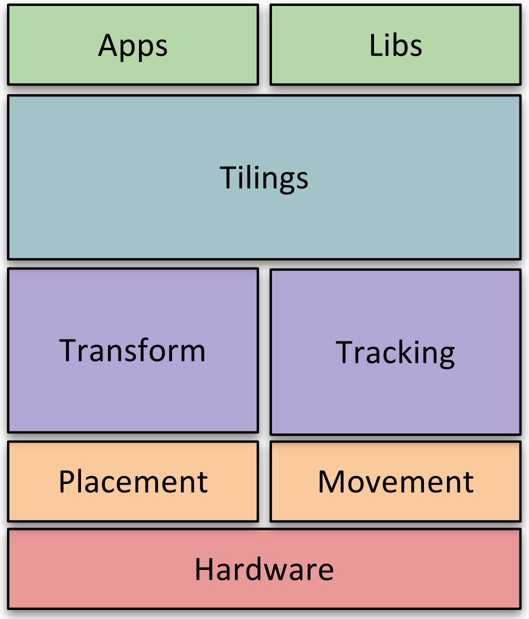
Developed AML, a framework to explicitly organize, place, move, and transform data across complex memory topologies.

Significance and Impact

Demonstrated ability to reproduce and exceed vendor-optimized linear algebra codes on very large inputs; programming runtime agnostic; directly targeting exascale memory architectures.

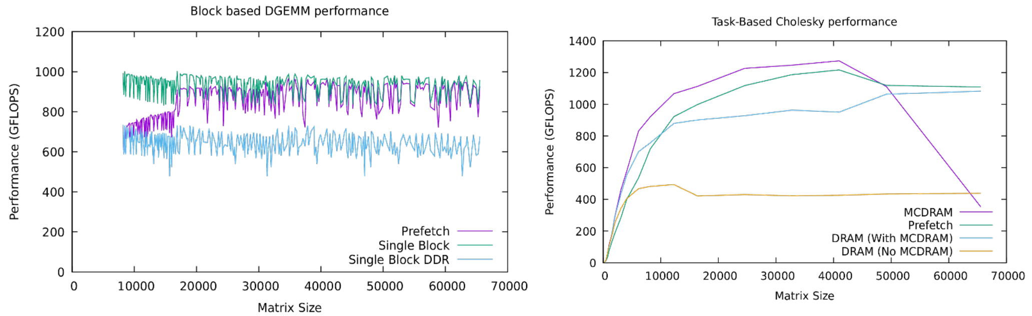
Research Details

- Explicit, application-aware memory management:, descriptive API for application-level data access, explicit placement and movement of data, abstract topology and memory device complexity.
- Collection of building blocks: few assumptions about user application, hardware-oblivious, application users can specialize the inner implementation of each offered abstraction, mix and match as needed.
- Locality optimizations for current and future hardware generations: static allocations with application insights, asynchronous (re-)mapping to optimize data locality on the go (data layout, HBM management, NUMA locality).



Major components of AML:

- Topology & hardware management (dependent on NUMA, hwloc, and SICM)
- Data layout descriptions (application-specific)
- Tiling schemes (including ghost areas)
- Data movement facilities (currently primarily memcpy or move_pages)
- Pipelining helpers (scratchpad, asynchronous requests)



Multi-memory, pipelined, out-of-core like implementation on Xeon Phi architectures compared to vendor implementation.

P. Beckman (PI), N. Denoyelle, S. Perarnau, B. Videau

Workflows for Active Search and Learning for Exascale Science

Scientific Achievement

Advanced workflow programming system (Swift/T) enables automated search and optimization for scientific problems in cancer as well as co-design problems in computer systems

Significance and Impact

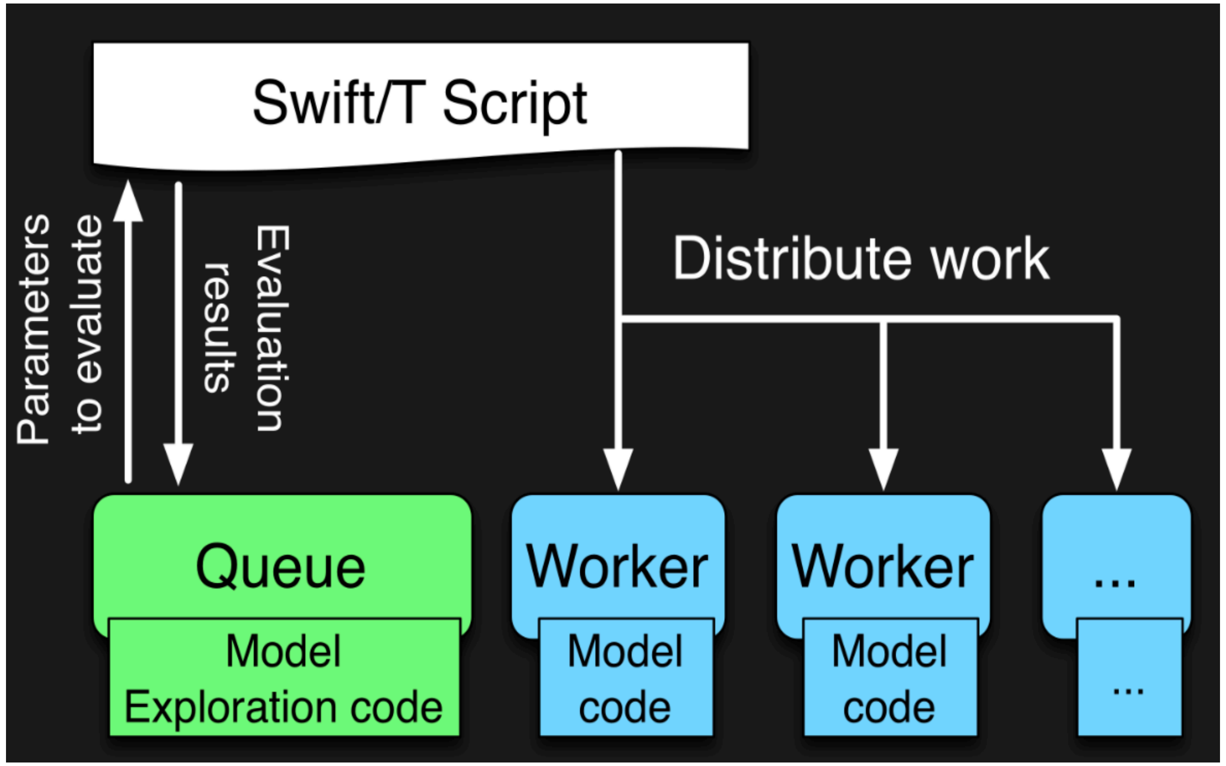
Demonstrates preliminary integration of machine learning to control scientific workflows; impacts design problems in neural networks for cancer research and use of exascale computers

Research Details

- Workflows built with the Swift/T workflow language can easily delegate control to 3rd party libraries that perform optimization, active learning, or other advanced algorithms
- Application successes include optimizing the design of deep learning systems for cancer pilot applications, optimizing simulation-analysis pipelines, and active classification for uncertainty reduction in influenza models
- This work was recognized with an R&D 100 in Fall 2018

Wozniak et al., BMC Bioinform. 2018; Ozik et al., IEEE Trans. Comput. Social Syst., 2018

Work was performed at Argonne National Laboratory



Sensitivity Analysis Tools for PDE-Constrained Optimization

Scientific Achievement

Developed new composable and scalable discrete-adjoint solvers in PETSc that are capable for local sensitivity analysis.

Significance and Impact

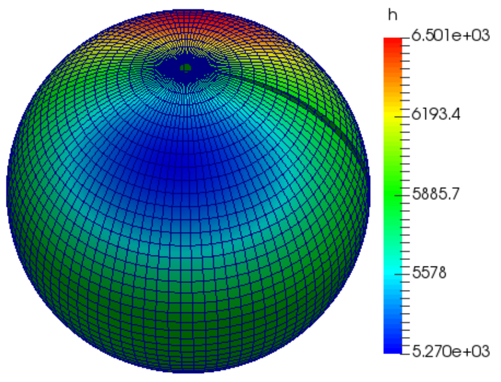
There is no ready-to-use second-order discrete-adjoint solver for large-scale time-dependent simulations. The new capability enables efficient calculation of gradient and Hessian-vector product for PDE-constrained optimization.

Research Details

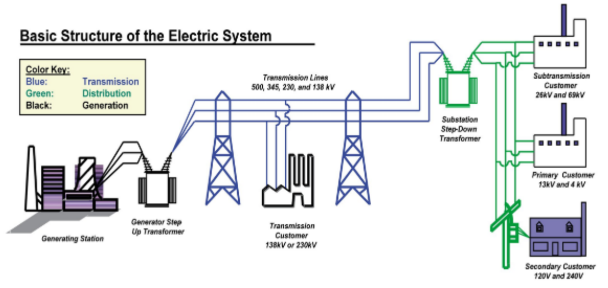
- The discrete-then-differentiate approach provides consistent gradients that are critical to the convergence of optimization
- Optimal adjoint checkpointing is developed to utilize the hierarchical memory on supercomputers (e.g. Theta)
- The implementation allows for multiple objective functions and integration over multiple time intervals as needed by applications in climate and power grid

$$
\begin{tikzcd}
f(y) \arrow[r, "\text{TLM}"] \arrow[d, "\text{Discretize}"] \arrow[rrrd, dashed] & f's \arrow[r, "\text{ADJ}"] & (f')^T\lambda \arrow[r, "\text{ADJ}"] & \Lambda^T(f'')\Lambda \arrow[d, "\text{Discretize}"] \\
f_n(y_n) \arrow[r, "\text{TLM}"] \arrow[rrr, bend right, "\text{Automatic Differentiation}"'] & f'_n s_n \arrow[r, "\text{ADJ}"] & (f_n)^T\lambda_n \arrow[r, "\text{ADJ}"] & \Lambda_n^T(f'')\Lambda_n
\end{tikzcd}
$$

Applications enabled by our sensitivity analysis tools:



Adjoint-based 4D-Var global data assimilation



Optimal economic dispatch for dynamic power system

[1] H. Zhang, S. Abhyankar, E. Constantinescu, M. Anitescu, Discrete Adjoint Sensitivity Analysis of Hybrid Dynamical Systems With Switching, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(5), p1247-1259, 2017
[2] J. Brown, E. Constantinescu, D. Ghosh, B. Smith, H. Zhang, PETSc/TS: A Modern Scalable ODE/DAE Solver Library, S. Abhyankar, arXiv:1806.01437

ANL: Hong Zhang, Emil Constantinescu, Barry Smith, Todd Munson

Multirate Time Integration Methods in Multiphysics Multiscale Simulations

ASCR Math Base (PI Smith)

Scientific Achievement

Developed multirate methods and software that allow for different step-sizes to be used for different components in the system or for different resolutions on the domain.

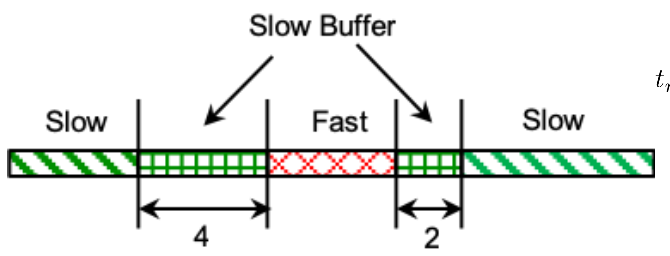
Significance and Impact

The multirate time integrator enables efficient simulation of complex systems that involve multiple physical processes and multiple time scales, which are common in climate models and material science.

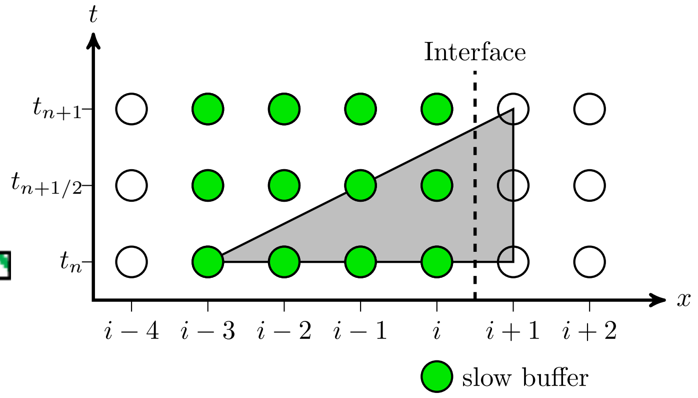
Research Details

- Differential equations are split component-wise, e.g. when adaptive mesh refinement is used
- We implemented the first multirate integrator in PETSc based on multirate partitioned Runge-Kutta methods
- The methods feature second-order accuracy, strong stability preserving property (critical for conservation laws) and minimal buffer region

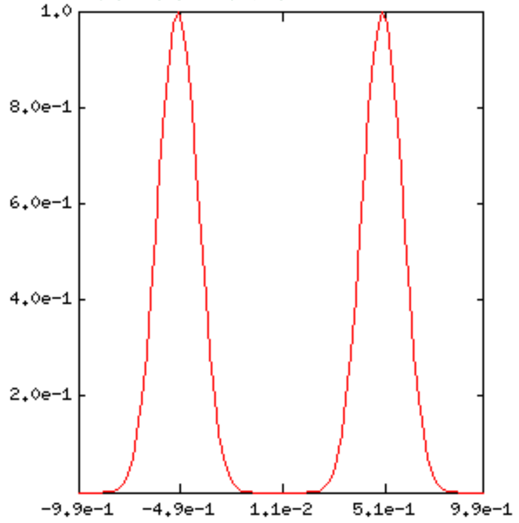
Only slow buffer is needed to satisfy order condition and flux consistency



Buffer region size depends on domain of dependence and number of stages



Results on advection equation with adaptive grids using finite-volume methods show theoretical order of convergence and mass conservation



Δt	L^1 err	conv rate	mass change
0.025	1.6243e-02		-2.73855e-16
0.0125	3.3634e-03	2.27183	-2.36848e-16
0.00625	8.2554e-04	2.02651	-1.33227e-16
0.003125	2.0017e-04	2.04411	-1.55431e-16
0.0015625	5.0982e-05	1.97317	-1.18424e-16

Second-order Multirate Partitioned RK Method