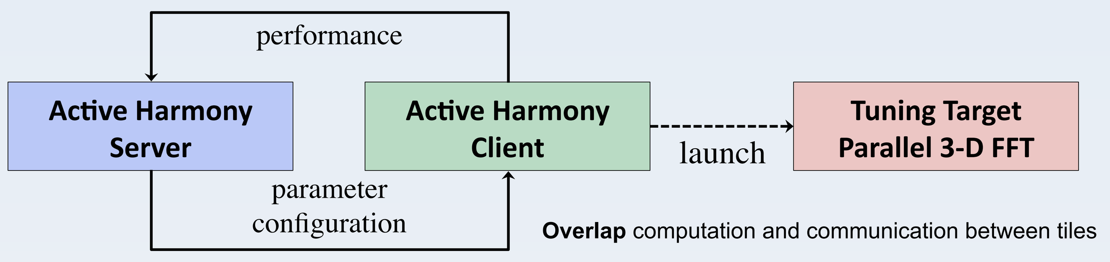
SUPER Computer Performance: Analysis and Optimization

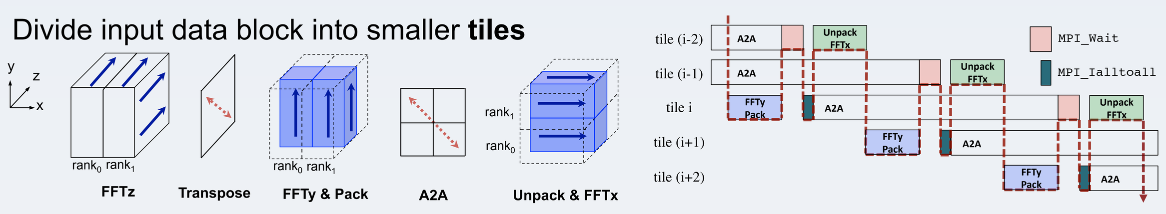
Jacqueline Chame — University of Southern California
Mary Hall — University of Utah
Jeff Hollingsworth — University of Maryland
Allen D. Malony, Kevin Huck — University of Oregon
Shirley Moore — University of Texas at El Paso
Boyana Norris — Argonne National Laboratory
Leonid Oliker, Sam Williams — Lawrence Berkeley National Laboratory
Pat Worley, Philip C. Roth — Oak Ridge National Laboratory

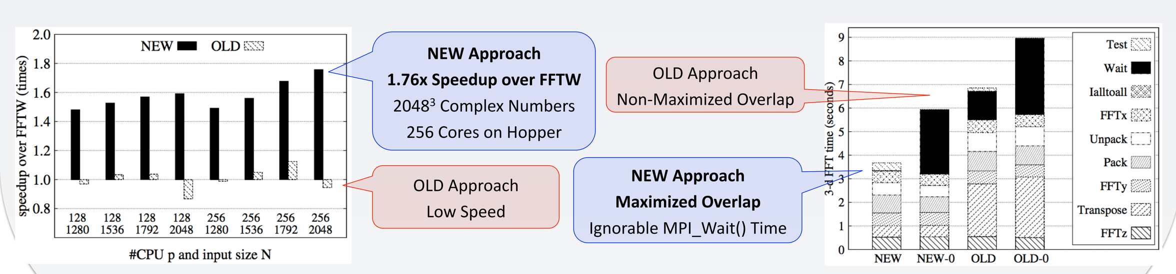
Active Harmony

Recent Results: 3-D FFT with MPI Non-Blocking All-to-All Communication



Divide input data block into smaller **tiles**



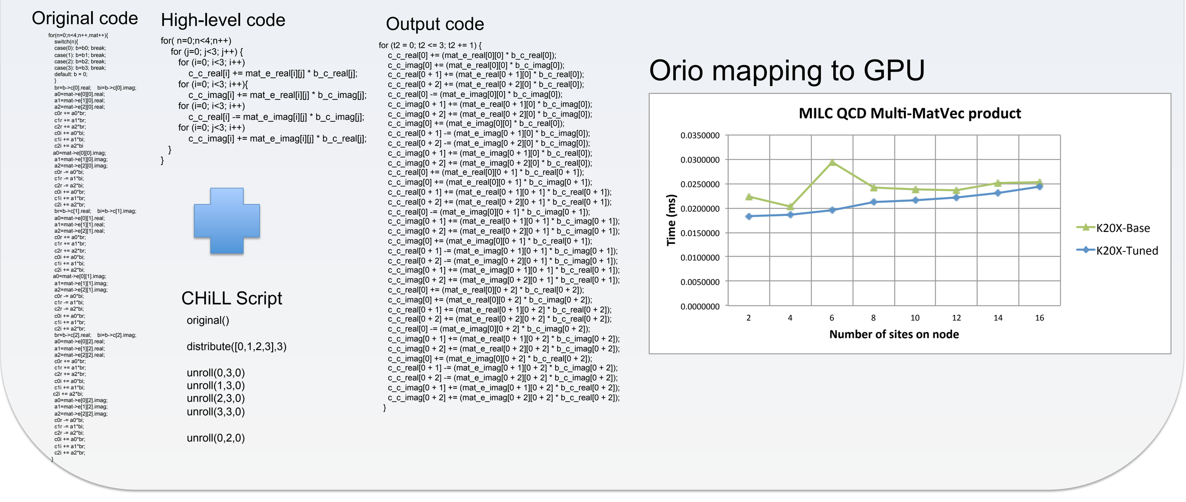


QCD + CHiLL and ORIO

CHiLL is a transformation and code generation framework designed specifically for compiler-based auto-tuning. ORIO takes as input code with embedded domain-specific optimization descriptions. Both systems permit users to guide code mapping to explore a search space of possible implementations of a computation, making it possible to achieve performance comparable to manual tuning.

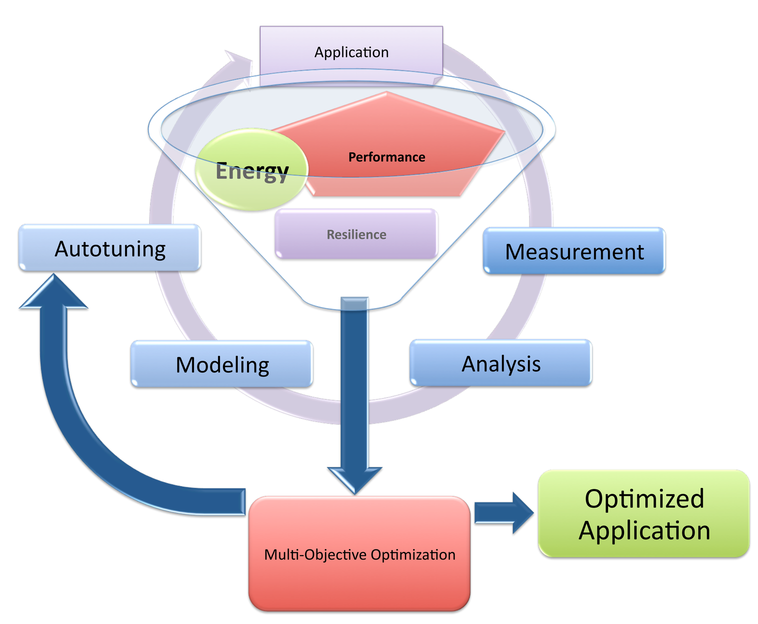
QCD (dslash operator)
- Key computation dominates execution time
- Application developers implemented low-level code that performs well across platforms
- But difficult to read, maintain, retarget
- New architectures (e.g., Xeon Phi) require different parallelization and data layout

GOALS:
- Start with high-level, readable code (requires simplification of data structures)
- Tools apply transformations to derive optimized code
- Map to different architectures automatically or semi-automatically



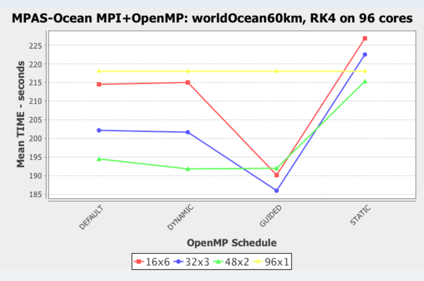
Introduction

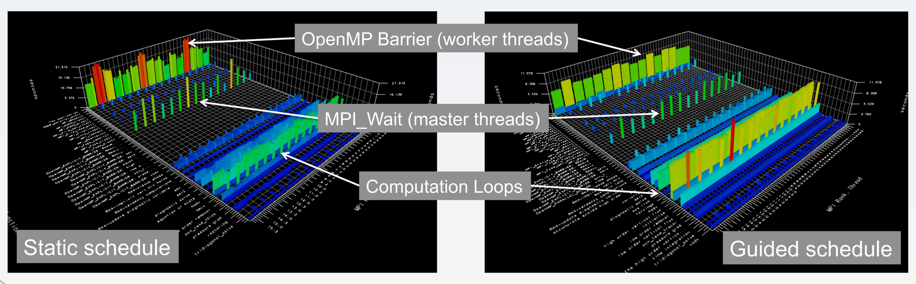
Methods for performance optimization of petascale applications must address the growing complexity of new HPC hardware/software environments that limit the ability of manual efforts in successful performance problem triage, software transformation, and static/dynamic parameter configuration. Greater performance tool integration and tuning process automation are necessary to manage and share performance information, to generate correct multiple code variants, to conduct controlled performance experiments, and to efficiently search and discover high-performant solutions, thereby improving performance portability overall. SUPER is advancing autotuning capabilities through the coupling of performance measurement, analysis, and database tools, compiler and program translators, and autotuning frameworks.



MULTISCALE + TAU

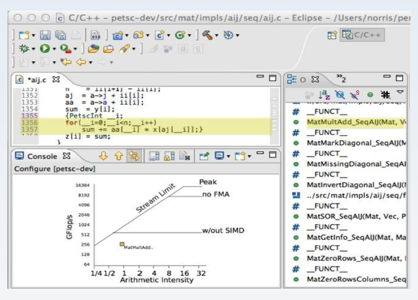
Recent Results: Adding OpenMP threads to MULTISCALE application MPAS-Ocean using TAU to analyze performance
- MPI block decomposition + OpenMP element decomposition reduces total instructions in computational regions (~10% faster than MPI alone)
- Guided OpenMP thread schedule balances work across threads (~6% faster than default)
- Weighted MPI block decomposition could further balance work across processes (~5% faster in some tests)
- Overlapping communication and computation could significantly reduce synchronization delays when exchanging halo regions (not yet implemented)





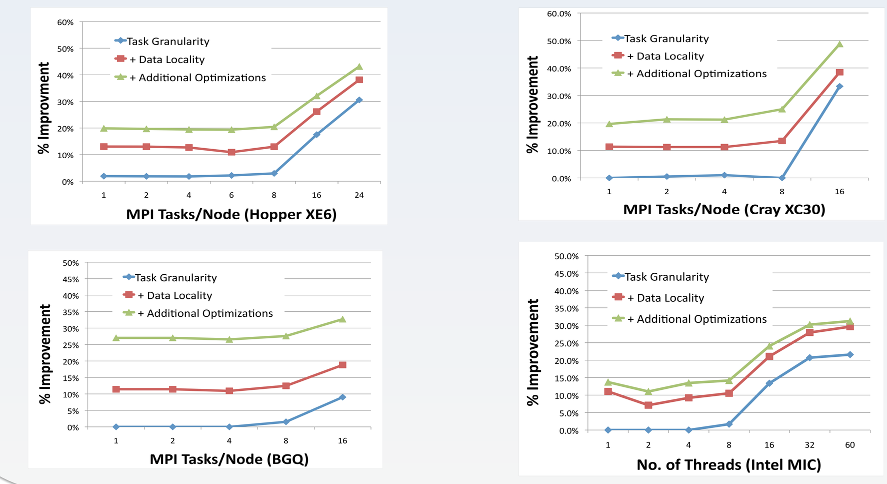
Roofline Toolkit

A downloadable collection of programs installable across a variety of systems, consisting of:
- Hardware characterization, via portable, instrumented microbenchmarks
- Software characterization via static analysis/modeling of source code, and performance counter execution instrumentation
- Data manipulation and visualization interface via TAU



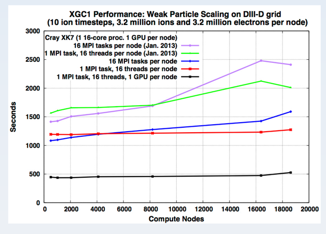
NWChem Analysis and Optimization

Recent Results: Optimize TEXAS two-electron integral package for NWChem on a variety of platforms.
- Balancing the load across MPI tasks by reducing task granularity
- Improving spatial data locality by increasing block size
- Developing new intelligent sorting algorithms
- Reducing redundant computations
- Using vectorization

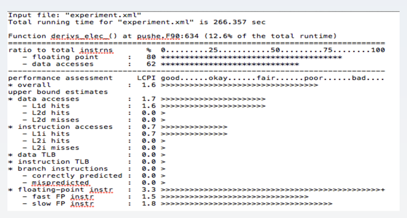


XGC1 Analysis and Optimization

Recent Collaborative Results: MPI and OpenMP optimization and GPU port of computational kernel resulted in 4X performance improvement on Cray XK7 of XGC1 particle-in-cell code for gyrokinetic simulation of tokamak plasma.

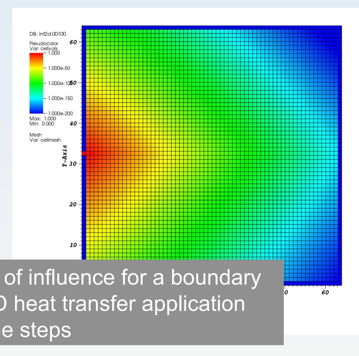


Next steps: Computational kernel takes >60% of execution time in hybrid CPU/GPU implementation, and >85% in CPU-only implementation. Continued optimization for both CPU and GPU is important. Analyses of CPU-only version using HPCToolkit and PerfExpert indicate high cycles per instruction in kernel due to expensive floating point operators (sqrt, exp, divide) and due to high number of data accesses in particle location search algorithm.



Tracking Value Influence

VIT is a tool for tracking how a value contributes to subsequent computation
- Tracking value influence can guide optimization, debugging, and fault tolerance strategies
- VIT tracks influence data for sequential, multithreaded, and multi-process (MPI) programs
- VIT tracks influence data using dynamic instrumentation (Intel Pin-based)



Propagation of influence for a boundary value in a 2D heat transfer application after 100 time steps

Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy Office of Advanced Scientific Computing Research