SUPER Optimization of Power, Energy, Performance, and More!

INSTITUTE FOR SUSTAINED PERFORMANCE, ENERGY, AND RESILIENCE

Prasanna Balaprakash, Paul Hovland (lead), Stefan Wild
Argonne National Laboratory

Ray Chen, Jeff Hollingsworth
University of Maryland

Ananta Tiwari
San Diego Supercomputer Center

Abstract

When a single objective, such as execution time, is available, the autotuning search problem can be posed as a numerical optimization problem. Increasingly, multiple metrics are of interest simultaneously, such as execution time, energy consumption, resilience to errors, power demands, and memory footprint. When the relative weights or constraints on these objectives are not known at search time, autotuning becomes a *multiobjective optimization problem*. We provide formalism for multiobjective optimization studies of broad applicability in autotuning, architecture design, and other areas of HPC. We discuss some of the potential tradeoffs among multiple objectives, and provide empirical evidence that such tradeoffs do exist in practice.

Multiobjective Optimization

The problem of empirically optimizing a code can be posed as the mathematical optimization problem:

$$\min_{x \in \mathcal{X}} F(x) = [F_1(x), \ldots, F_p(x)]$$

THE OBJECTIVE

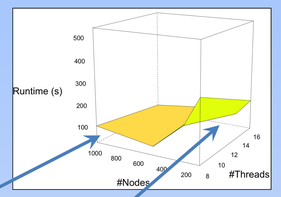
- $F_1(x),\ldots,F_p(x)$ are p **possibly conflicting objectives** that need to be optimized simultaneously
- Can capture average, median, quantile (e.g., worst-case) empirical performance
- Often stochastic/noisy (from measurement and/or run)
- Depends on machine and input size (or distribution over inputs)
- Assumes no *a priori* weights available for the objectives
- Examples: run time, power, energy, failure rate

THE DECISIONS

- Binary (compiler type, +examples)
- Integer (unroll factor, register tiling, +examples)
- "Continuous" (algorithmic parameters, internal tolerances)
- Each *x* generates a code variant (e.g., through source-to-source or compiler-based transformation)

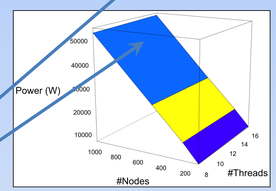
THE CONSTRAINTS

- Ensuring feasibility of transformation
- Correctness of output, maximum temperature, etc.

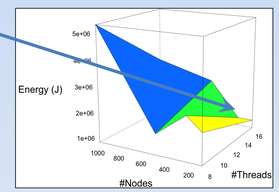


Illustrative example on three objectives on IBM BG/Q

Minimizing run time
is conflicting with
power consumption



Minimizing energy consumption
conflicts with
run time



Multi objective optimization concerns the study of optimizing two or more objectives simultaneously. Even if there is a unique optimal (software/hardware) decision when any of the objectives is considered in isolation, there may be an entire set of solutions when the objectives are considered collectively.

GOAL:

Develop multi objective optimization framework that allows exploration of the tradeoffs

Existence of these tradeoffs can motivate hardware designers to expose a richer and more appropriate set of knobs to future administrators and software designers

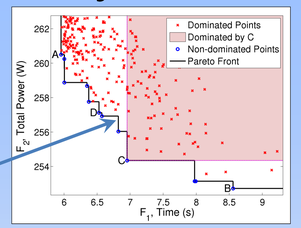
A framework that is sufficiently general and can be easily extended to incorporate new hardware-and software-based power/energy knobs as they become available

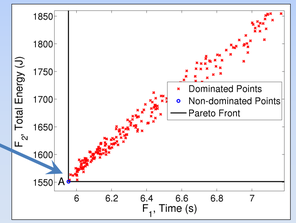
Pareto Optimality

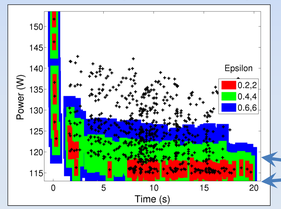
Arises when several objectives need to be to optimized simultaneously

- Sometimes objectives are correlated and satisfied simultaneously; otherwise there are tradeoffs
- Code variants now live both in a decision space and in an objective space

- Pareto front contains significantly richer information than one obtains from single-objective formulations
- Code variants for which no other variant is better in all objectives are said to be ***nondominated* or *Pareto optimal***
- For search algorithms, only certain regions of the objective space are of interest
  - The ideal and nadir point define the range of objective that include all possible optimal tradeoffs
- When multiple objectives are **not competing** the Pareto front corresponds to a single point, which simultaneously minimizes both objectives



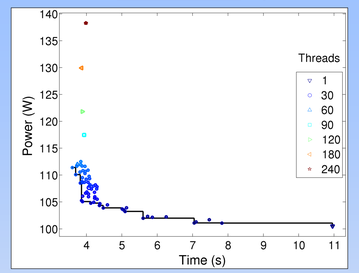


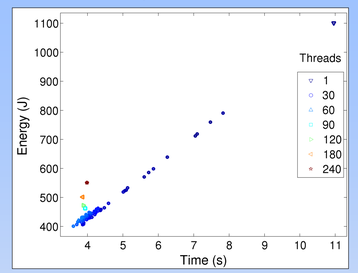


- For many time-power-energy multi-objective problems, there can be measurement error in each objective
  - Consequently, we have a **relaxed Pareto front** that potentially consists of a cloud of points

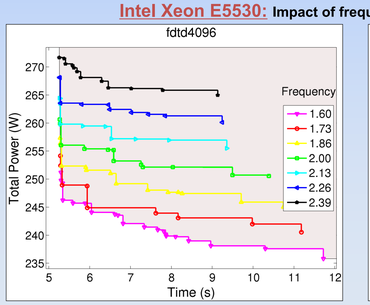
Tradeoff Studies I

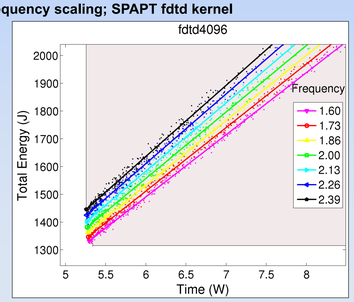
Intel Xeon Phi: **Impact of number of threads; TORCH quick sort kernel**





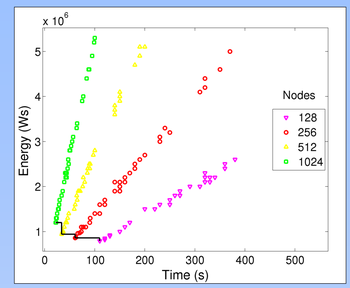
Intel Xeon E5530: **Impact of frequency scaling; SPAPT fdtd kernel**

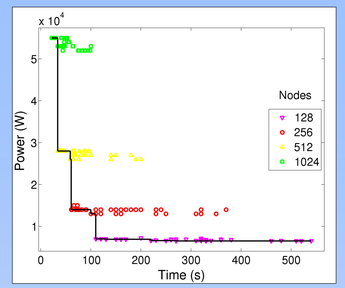




Tradeoff Studies II

Blue Gene/Q: **Impact of number of nodes; MiniFE mini-application**





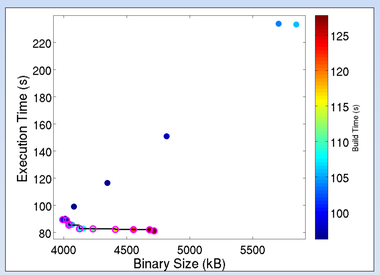
- Because of the relationship between power and energy, all points on the energy-time Pareto front have a corresponding point on the power-time Pareto front
- Number of non-dominated points for energy-time is bounded by the number of non-dominated points for power-time
- A necessary condition for *x* to be a non-dominated point on the energy-time Pareto front is

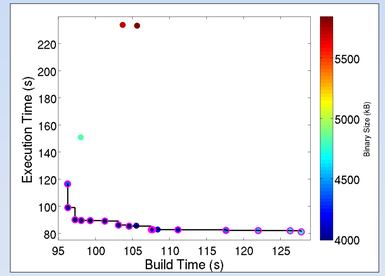
$$P(x) \leq \frac{P(x^{(1)})T(x^{(1)})}{T(x)}$$

- To observe tradeoff, the power savings must outpace the product of idle power and relative slow-down:

$$P(x^{(1)}) - P(x) \geq \frac{T(x) - T(x^{(1)})}{T(x^{(1)})} P_I$$

Three simultaneous objectives: **Build time, binary size, and execution time**





- When these three objectives are considered simultaneously, all points are Pareto optimal
- When the objectives are considered pairwise, some points are dominated

Our studies show that in some settings objectives of interest can be strictly correlated and there is a single, "ideal" decision point; in others, significant tradeoffs exist.

Future Investigations

- Develop multi objective optimization algorithms for autotuning search
- Identify appropriate use cases
- Study other tradeoffs:
  - Resilience versus memory footprint; Resilience versus execution time
  - Memory footprint versus execution time; Memory footprint versus energy

"Can search algorithms save large-scale automatic performance tuning?" Balaprakash, Wild, & Hovland, *iWAPT 2011*.
"Online adaptive code generation and tuning ." Tiwari & Hollingsworth, *IPDPS 2011*.
"SPAPT (Search Problems in Automatic Performance Tuning)." Balaprakash, Wild, & Norris, *ICCS 2012*.
"Multi-objective optimization of HPC kernels for performance, power, and energy" Balaprakash, Tiwari, & Wild, Preprint ANL/MCS-P4069-0413, 2013

Argonne NATIONAL LABORATORY
UNIVERSITY OF MARYLAND
SDSC SAN DIEGO SUPERCOMPUTER CENTER

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 Science, Advanced Scientific Computing Research.