Auto-tuning for Energy Usage in Scientific Applications

Ananta Tiwari, Michael A. Laurenzano,

Laura Carrington and Allan Snavely

San Diego Supercomputer Center

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

PMaC
Performance Modeling and Characterization

Background

- "Power Wall"
  - "… power is *the* scarce resource in the design of a modern processor." – Exascale Computing Study, Kogge et. al.

- Power-aware HPC research
  - Develop power characterizations via direct measurement or via explanatory models
  - Minimize the amount of energy required to solve various scientific problems

Our Work

- Tackle the “power wall” problem from software side
- Identify and understand software tunables that have high impact on power draw
- Tweak the tunables to optimize for energy usage while maintaining satisfactory levels of performance

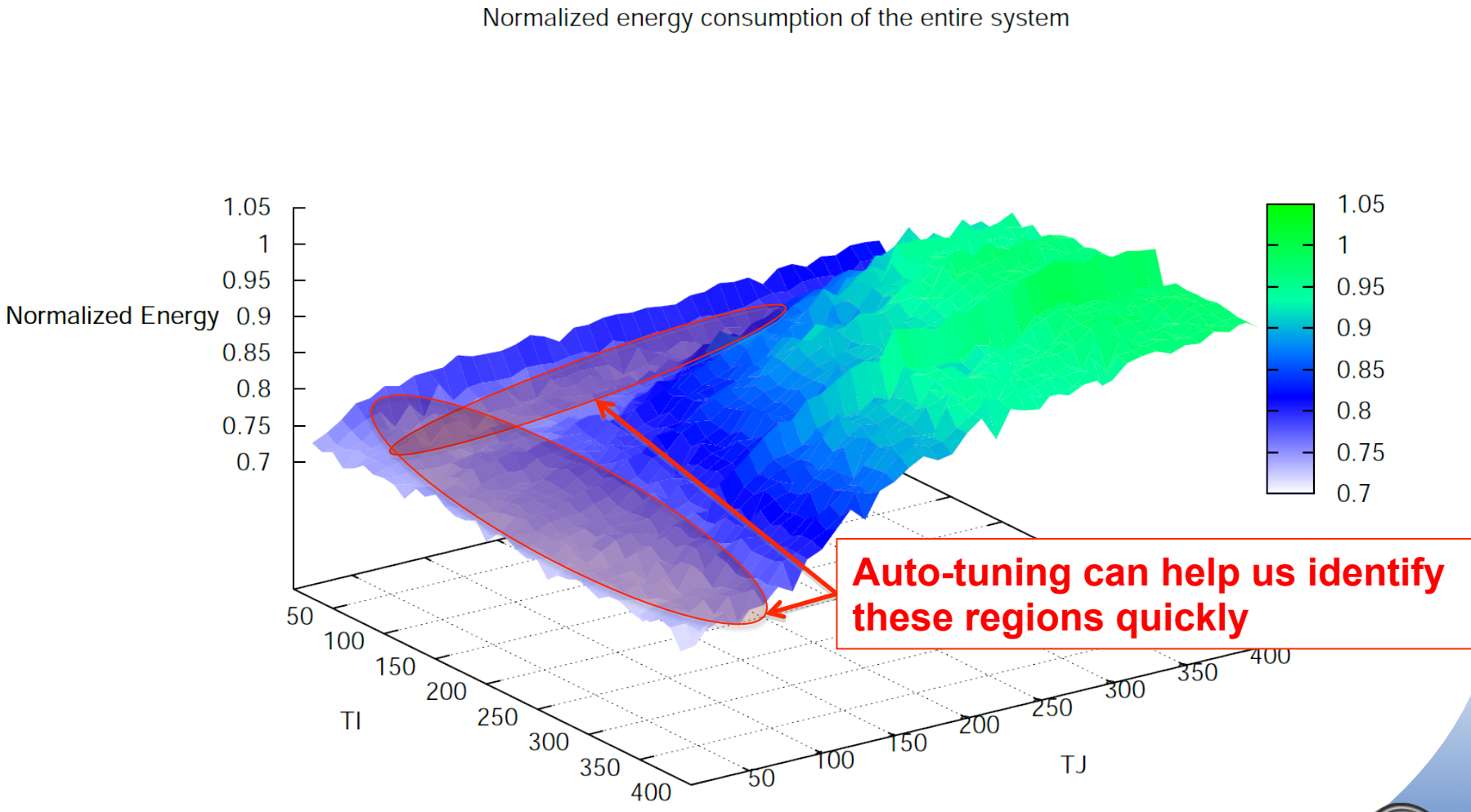
Motivating Questions

- Can simple loop transformations (blocking, unrolling, etc.) help reduce power draw?
  - How similar is this to performance tuning?
- Can we use search heuristics to find a code-variant that performs better in terms of energy consumption?
  - Leverage performance tuning work?

Compiler-Based Methodology

- Generate and evaluate variants of application hot-spots
  - Exclusive focus on stencils in this paper
- Tunables in this work – cache tiling, loop unrolling, clock frequency
- Treat CPU clock frequency as one of the tunables

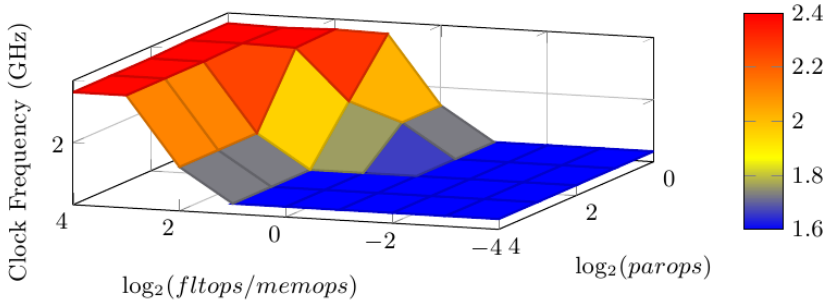
Sample Optimization Surface



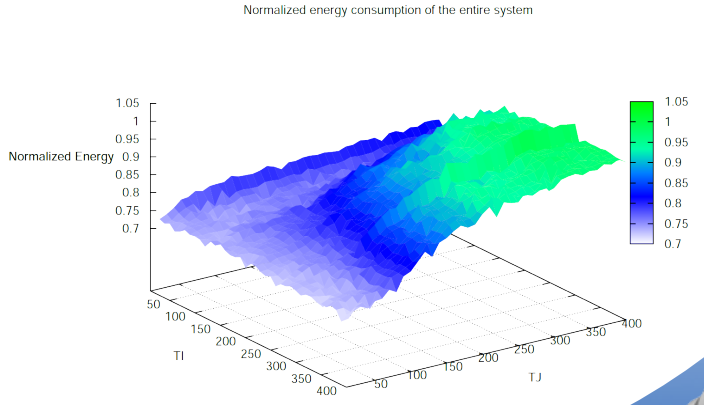
Integrated Hardware/Software Approach

- Hardware motivated by software
  - Customized hardware
  - Co-designed hardware
- Software tuned to hardware
- Our approach uses both

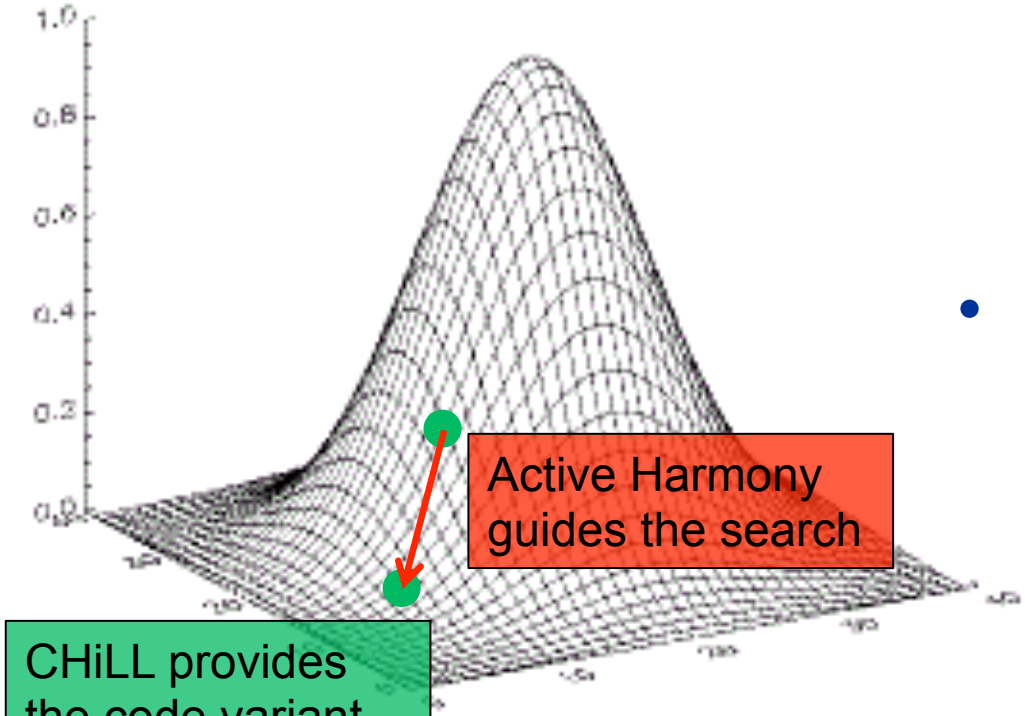
Clock Frequency Scaling



Software Tuning Parameters



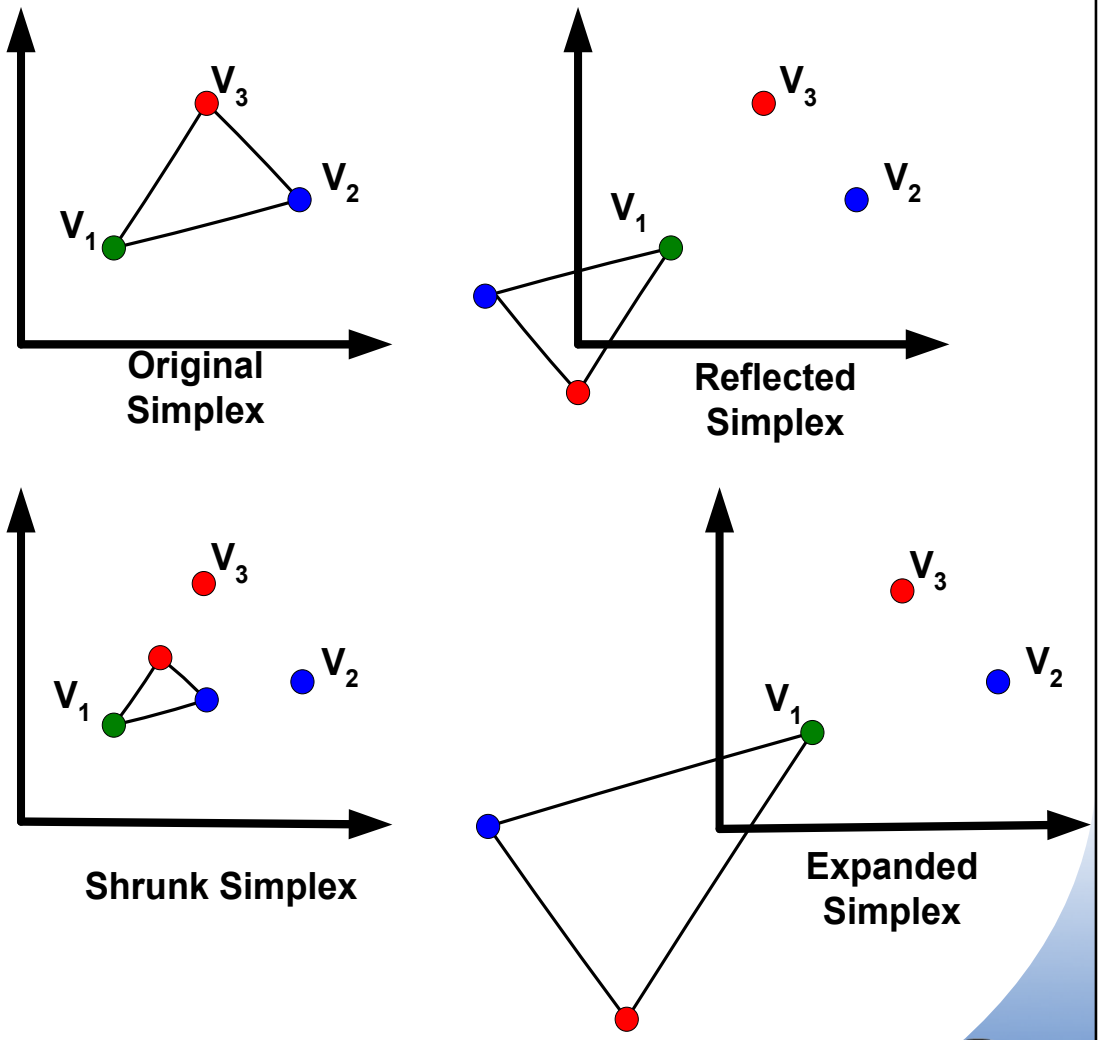
Enabling Components



- **Active Harmony (Maryland, open-source)**
  - **Search-based auto-tuner that suggests tunable parameter confs based on observed power-draw**
- **CHiLL (Utah, open-source)**
  - **Polyhedra-based loop transformation framework**

Active Harmony Parallel Rank Ordering Algorithm

- Direct Search
- All but the best point of simplex moves



Auto-tuning Feedback Metric

- **Feedback Metric:**
  - E (Power x Delay)
  - ED (Energy x Delay)
  - ED^2 (Energy x Delay x Delay)
  - D (Delay)

- **Appropriateness of metric depends on the overall goal of the tuning exercise**

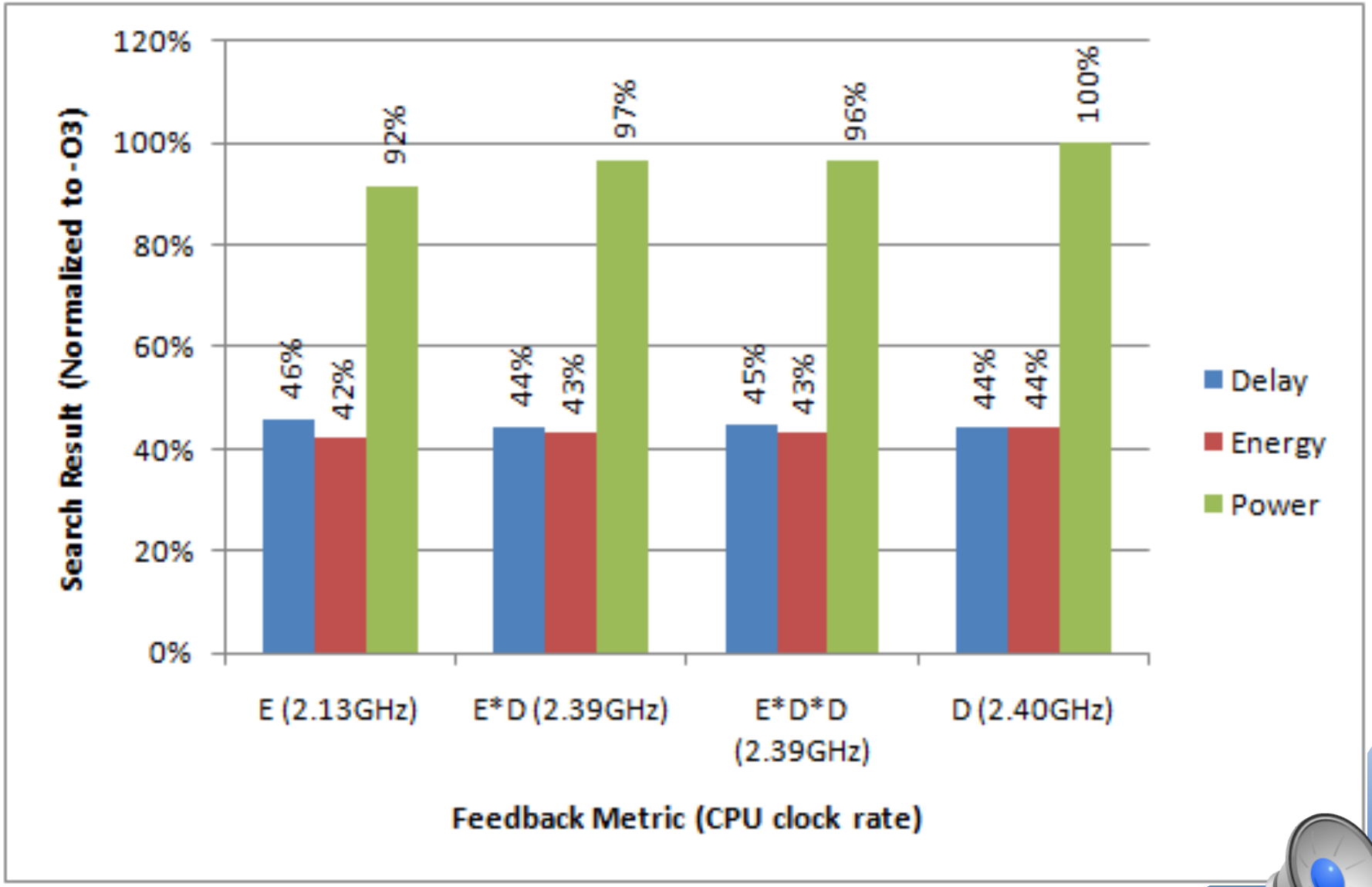
Experiment Details

- Intel Xeon E5530 workstation
- 2 quad-core processors
  - 32KB L1 cache and 256KB L2 cache per core
  - 8MB shared L3 cache per processor
- 8 available clock frequency settings
  - 1.60, 1.73, 1.86, 2.00, 2.13, 2.26, 2.39, 2.40GHz
- Processor clock frequency changed using cpufreq-utils package

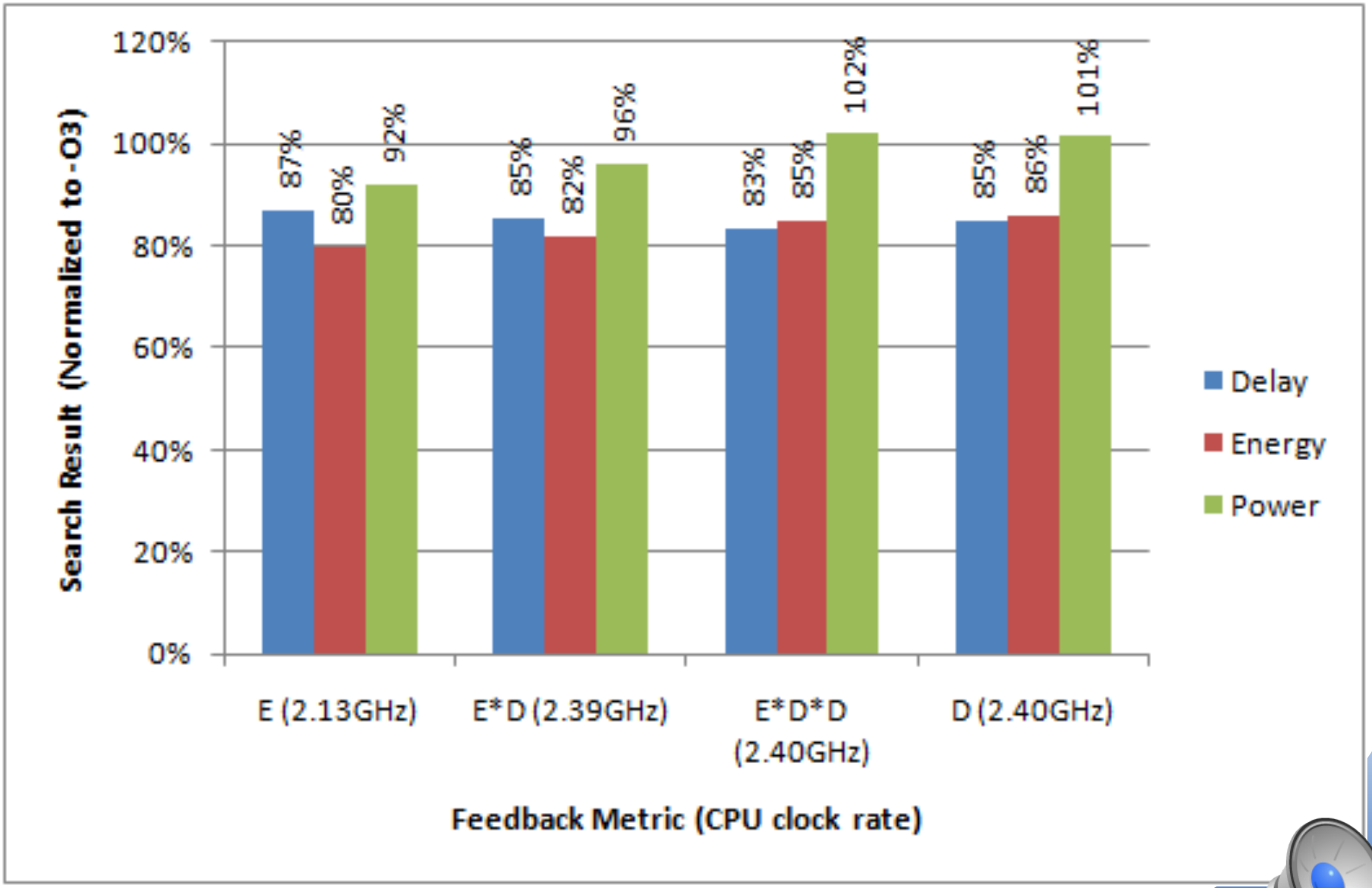
Poisson's Equation Solver

- Problem size: 640^3 grid
- Relaxation Function
  - Uses redblack successive over-relaxation method
  - 7-point stencil
  - Triply nested loop
  - Tile the two outermost loops
- Error Function
  - Sweeps though the local grid to calculate the L2-norm
  - Triply nested loop
  - Tile all loops and unroll the innermost

Parameter Search Results - Relaxation



Parameter Search Results - Error



Highlights

- **Energy consumption minimizing configurations, on average, can save:**
  - **5.8% energy with a performance loss of 4.1% for relaxation function**
  - **5% energy with a performance loss of 3.9% for error function**
- **Non-trivial interactions between compiler performance optimization strategies and energy usage**

Going Forward

- Utilize fine-grained DC power measures
- Characterize energy usage for individual system components (such as CPUs, DIMMs)
  - Build power/energy models
  - Auto-tune based on models?
- Target specific optimization techniques to reduce energy usage of various components
- Other kernel and optimization types
  - Loop fusion/split, data copy
- Online auto-tuning

Summary

- **Software based approach to energy auto-tuning for HPC applications is promising**
  - **Non-trivial interaction between software-level performance-related tunables and energy usage**
- **This work used a fairly inexpensive power meter and leveraged open source projects to explore energy and performance optimization space for stencils**

Questions

Please direct questions to tiwari@sdsc.edu and michaell@sdsc.edu