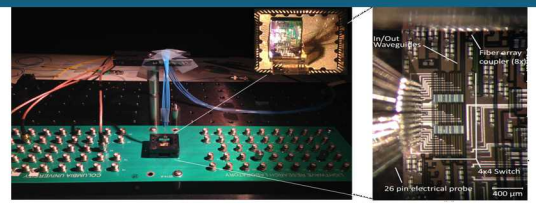
SAND2020-8056C

Putting compilers in the simulation co-design loop with surrogate performance models



PRESENTED BY

Jeremiah Wilke, Sandia National Labs, Livermore, CA

Collaborators: Cannada Lewis

Modsim, Virtual Seattle, WA, 2020

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Diverse space of AI/ML problems driving proliferation of architectures, programming frameworks, and compilers

Algorithms*

- Conventional HPC
- Mixed ML–HPC workloads
- Deep Learning

Even HPC (non-ML) workloads will either want to leverage ML architectures or accelerate kernels with ML

Architectures*

- "More Moore's" scaling of current architectures
- Domain-specific conventional architecture, FPGAs
- Deep learning chips
- Analog neural networks
- Neuromorphic, spiking neural networks

Programming Frameworks*

- TensorFlow/JAX, PyTorch
- TVM, Tensor Comprehensions
- C++, Kokkos, Raja
- Julia, R, Python
- Fugu (SNNs)

Compiler Tools, Representations*

- Glow
- ONNC
- Chill/Polly
- MLIR
- NNEF IR
- Halide IR

Autotuners*

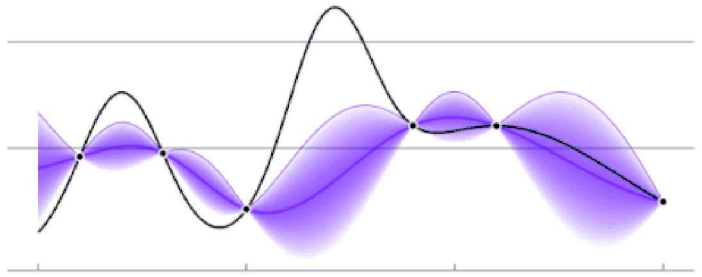
- GPTune
- Ytopt
- OpenTuner
- CLTune
- TVM
- Milepost GCC

*Not an exhaustive list!

Performance auto-tuning (particularly compilers) falls into category of "most difficult" optimization problems

- Huge search space of compile and runtime parameters
  - LLVM has 60+ compiler passes, many of which have parameters (e.g. vectorization, unroll)
  - Many scientific kernels have 5 or more runtime parameters
- No closed-form expressions or simplifying assumptions on performance function
- Noisy measurements, "chaotic" dependence on parameters
  - No derivative-based or gradient-descent methods

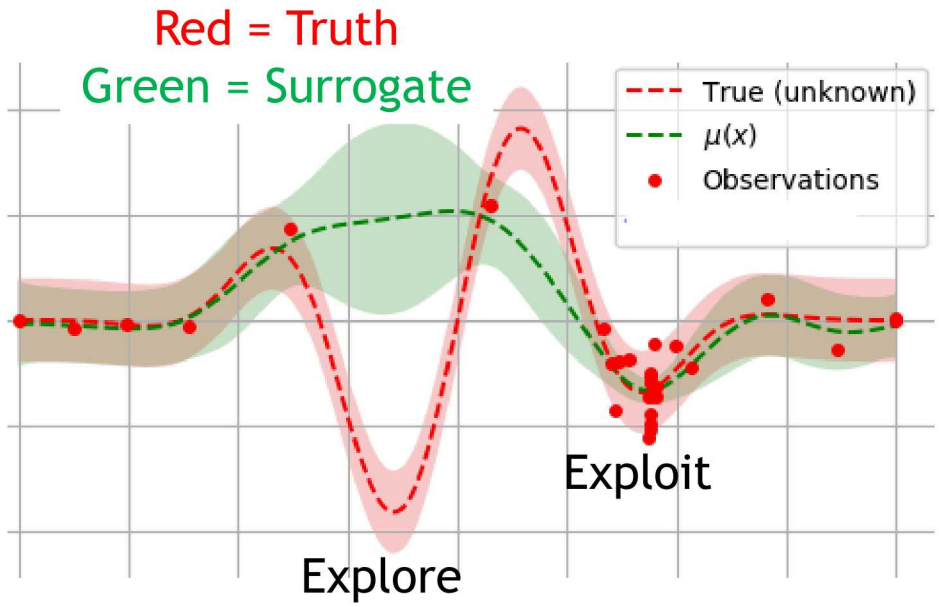
Surrogate model with simpler properties



These traits define a "Black Box" optimization

Performance tuning as a black-box optimization problem in fixed search space

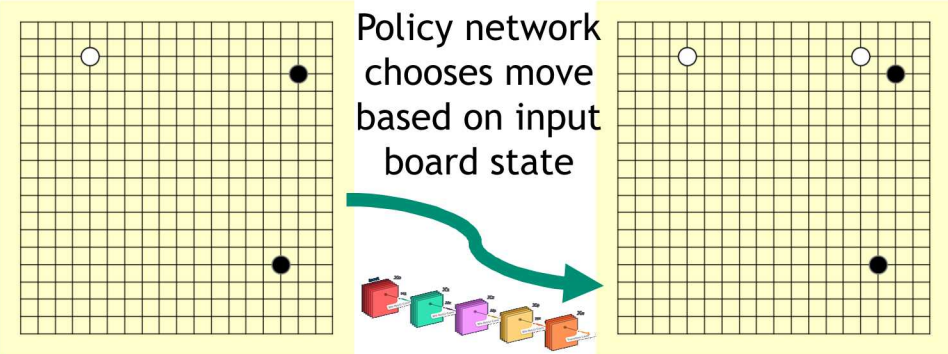
- Well-defined set of tuning dimensions
  - Categorical parameters (CPU vs GPU)
  - Integer parameters (Tile Sizes)
  - Continuous parameters (Not common in autotuning)
- Every method driven by exploration/exploitation tradeoffs
  - Exploration: Try new possibilities that might be better
  - Exploitation: Focus on areas that seem most promising
- "Model-free methods" directly search the space without attempting to build a performance model
  - Genetic algorithms, e.g. prefer "traits" with high "fitness"
- "Model-based" methods guide search with simple performance models that are easy to optimize
  - Bayesian optimization commonly uses Gaussian Processes to select search points most likely to improve performance
  - Any surrogate model (deep neural networks/ensemble methods) could guide search to most promising candidates



Surrogate models used for fast rollout/lookahead in frameworks e.g. TVM

Performance tuning as a reinforcement learning problem: teaching compilers to play Go

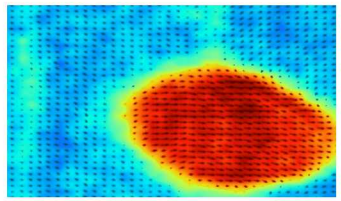
- Compiler passes (iterative compilation) for individual kernels, whole graph optimization are a sequence of actions with varying rewards
- Reinforcement learning of optimal actions is most challenging when:
  - The number of possible actions is large (absolutely true of compiler optimizations)
  - Need to take many actions until reward is received, i.e. "sparse" rewards (possibly true of optimization)
- Many lessons can be learned from (maybe) most impressive reinforcement learning achievement: AlphaZero on Go



Success more a matter of brute-force data collection on TPUs than clever algorithms?

- Moves selected based on policy/value network over actions (a), state(s)
  - Pi(s,a) -> next move
  - V(s) -> value of a given state
- Networks trained over millions of games of self-play *without* training on human experts
- A "player" is defined by their policy network and exploration/exploitation preference

ModSim challenge: Develop "grandmaster" compiler that exceed human performance



Map to some model IR

Input app "board state"

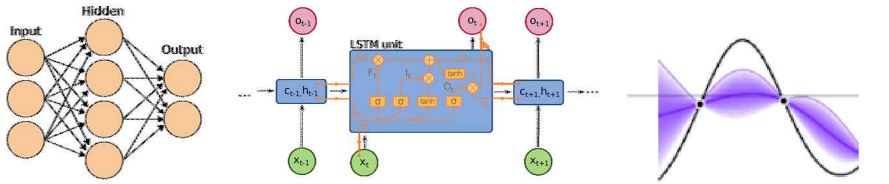
Collect performance results

Challenge #1: Tons of data needed to train these models. Can we harvest all the performance data generated daily on DOE platforms?

	n	time	m	gflops
0	100	0.012	100	2.6
1	200	0.210	200	5.2
2	400	0.290	400	10.4
3	800	0.350	800	20.8

Challenge #2: Many possible representations for building models, which are constantly changing. Can we begin exploring even if IR space constantly changing?

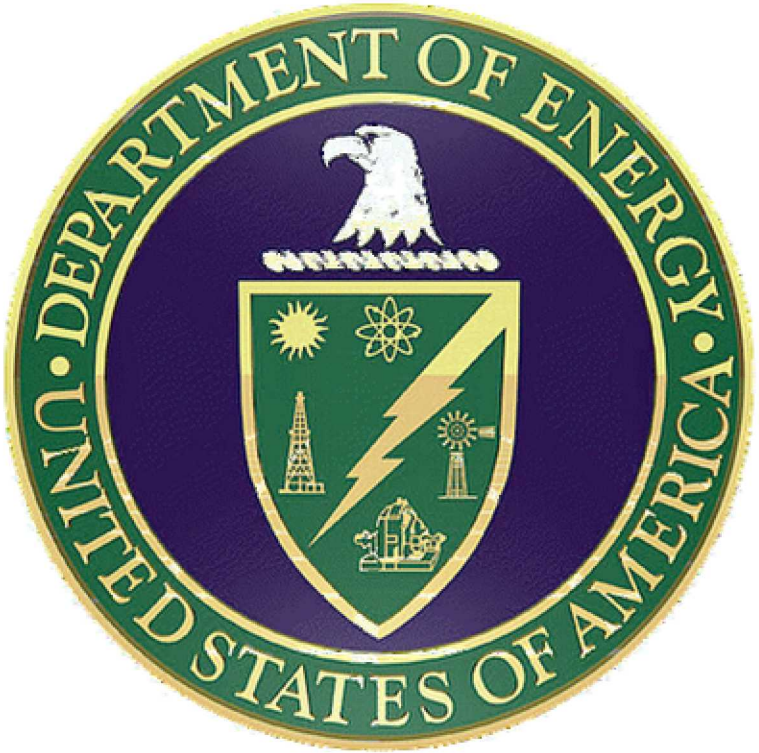
Build surrogate Performance models



NNs RNNs GPs

Acknowledgments

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.



NNSA
National Nuclear Security Administration

Sandia National Laboratories