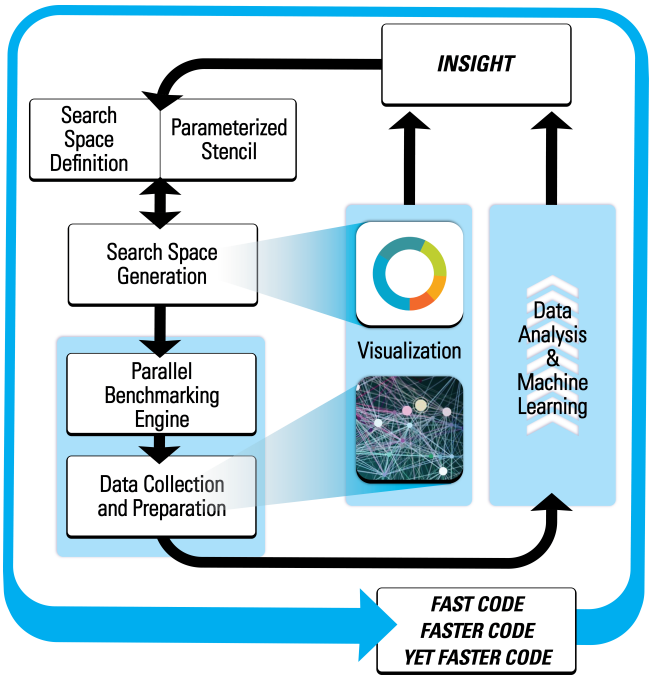
BONSAI

BENCHTESTING OPEN SOFTWARE AUTOTUNING INFRASTRUCTURE

THE PREMISE

The goal of the **BONSAI (Benchtesting OpeN Software Autotuning Infrastructure)** project is to develop a software infrastructure for using parallel hybrid systems at any scale to carry out large, concurrent autotuning sweeps in order to dramatically accelerate the optimization process of computational kernels for GPU accelerators and many-core coprocessors.



Test Many Data Sets

In the course of our work on accelerating the ALS algorithm for collaborative filtering, we discovered that the optimal parameter configuration depends heavily on the properties of the input data set, which motivates tuning sweeps over many datasets. In particular, consider tuning a sparse matrix kernel by making a sweep over all matrices in the Univ. of Florida matrix collection (currently 2,833 problems and growing).

Test Many Data Layouts

Modern hardware is increasingly sensitive to the layout of data in memory. A number of different layouts have been proposed for dense linear algebra (row and column major, tile, space-filling curves), sparse linear algebra (CSC/CSR, ELLPACK, SELL-C/SELL-P, Sell-C-Sigma, BCSR, DIA, COO), deep learning (NCHW, NHWC), PDE discretizations (structure of arrays or array of structures), etc.

Collect Lots of Performance Metrics

NVIDIA Maxwell can collect 111 different hardware counter metrics, based on 75 different events, and only a few can be collected in a single run. This forces many reruns of the kernel to collect all relevant metrics. Similarly, Intel Xeon Phi Knights Landing can collect 119 native events, using 5 counters, also making it necessary to rerun the kernel multiple times.

DELIVERABLE

We are developing a parallel, distributed benchmarking engine capable of scaling to tens of thousands of nodes, to benchmark millions of combinations of kernel configurations, problem sizes, and representative input datasets, while collecting hundreds of performance metrics such as time, energy consumption, cache misses, and memory bandwidth.

Compilation

BONSAI performs parallel compilation, both across distributed memory nodes and within each node. Compilation of a large number of kernels takes a significant time in the autotuning process. Numerical kernels are frequently heavily unrolled, which contributes to long compilation times. Also, compilation time can be extremely nonuniform. Therefore, BONSAI dynamically balances the workload.

Benchmarking

We use a standard MPI parallel job designed to work in existing batch queuing systems such as TORQUE PBS. This makes BONSAI deployable on a wide variety of systems, from small university clusters, to cloud computing, to national supercomputer centers. When available, BONSAI takes advantage of multiple accelerators within each node.

Data Collection

BONSAI will provide a framework to simplify the process of collecting hardware counters and performance data. We will leverage the various open-source and vendor-specific libraries such as NVIDIA's CUPTI API, AMD's CodeXL, Intel's VTune, and the open-source PAPI library. BONSAI will simplify the task of instrumenting the kernel and provide a simple interface for selecting the counters to be collected.

Data Analysis

We will provide a number of analytical tools and examples to guide the developer in analyzing their code. The analytical tools provided with BONSAI will include statistical and machine-learning tools in addition to a number of visualization utilities. These tools will leverage open-source data analysis libraries such as the PyData stack, R, and Spark tools such as MLlib.

SPONSORED BY

FIND OUT MORE AT
http://icl.utk.edu/bonsai