Input-dependent code variant selection via machine learning

Tao Wang¹  David Beckingsale²  Nikhil Jain²  Todd Gamblin²  Frank Mueller¹
¹ North Carolina State University  ² Lawrence Livermore National Laboratory

• Goal: boost runtime performance for input-dependent programs
• Problem: predict the best compiler-generated per-loop code variants for input-dependent programs via machine learning models

Assumptions

• A1: different code variants achieve the best performance for different inputs
• A2: the number of performant code variants is sufficiently small
• A3: code variants can co-exist and preserve their original performance
• A4: given an unseen input, the best code variant is predictable with low overhead

Approach

Prediction-based input-dependent code variant selection framework

Kripke: Input-sensitivity and Representativeness

• Ideal w/ O3: O3 is used as one code variants for perfect code variant prediction
• There are input-sensitivity: none of all code variants is always the best across inputs

Future work

• Validate A1~A4 on other programs
• Evaluate fine-grained code composition
• Explore tuning-based paradigm

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.