Thoughts about Advanced Tools to Boost Productivity

Chunhua “Leo” Liao

Oct 22, 2020

LLNL-PRES-773949

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

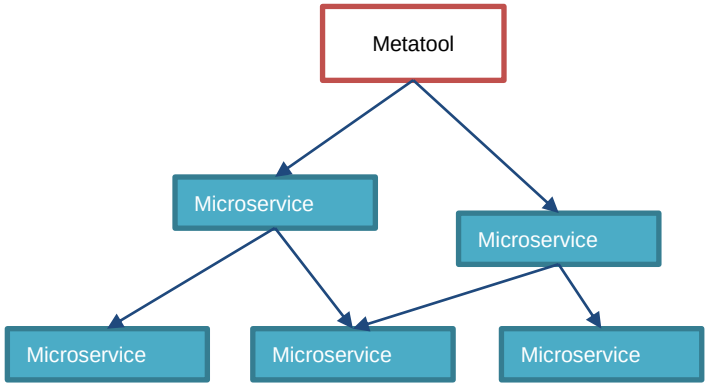
Lawrence Livermore National Laboratory

Position Statement

1. Software Ecosystem for future:
   a) Software tools as interchangeable building blocks for composable services
   b) Need common APIs, exchangeable data formats for individual compilers/tools
2. Tools:
   a) Need common benchmarks and right metrics to communicate status/progress with users/sponsors/researchers
   b) Benchmarks: regression tests = positive tests + negative tests
   c) Tool directory & Dashboard: showing available tools, compared with standard metrics
3. Education and Training
   a) Play tools online, using only a web browser
   b) Train tool users/developers online using dockerized environment embedded inside a web browser

Software Ecosystem: Program Analysis and Optimization Tools as Services

- Multiple tools needed to address any HPC challenges (e.g. productivity, performance, efficiency)
  - Making compilers and tools as interoperable, composable services
  - Enable building programming models and metatools by composing services
- Individual tools
  - Make them into microservices available 24x7: through local or web instances, possibly dockerized
  - Define standard APIs and data formats to accept input and give out output



Benchmarking of Tools with Right Metrics

If You Can't Measure it Correctly, You Can't Improve it

- Dedicated Benchmarks for Tools
  - Regression positive/negative tests
  - Collaboratively define holistic metrics
  - Auto-regenerate results if anything changes: benchmarks, tools, environments …
- Live results :
  - A scoreboard to show the state-of-the-art in objective, quantitative ways
  - For sponsors, users, and researchers

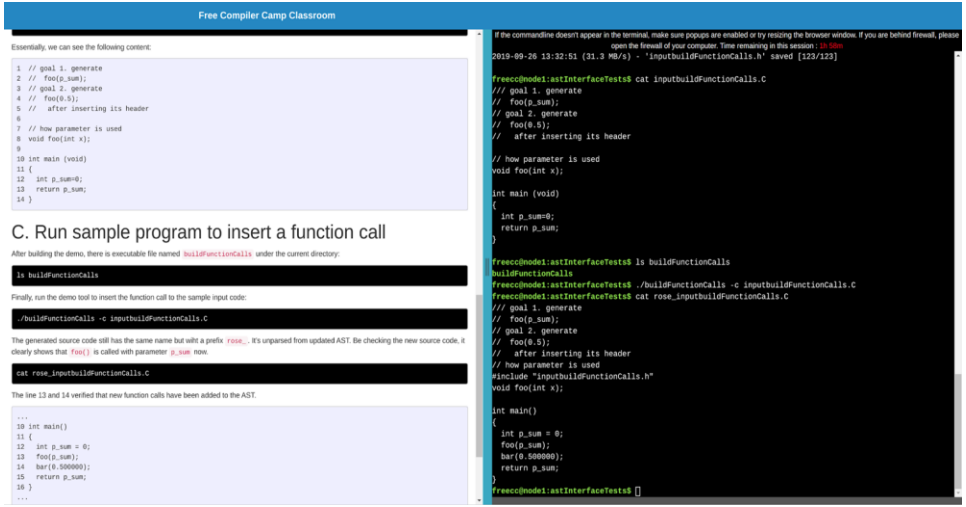
Metrics report:

Tool	Languages	TP	FP	TN	FN	Recall	Specificity	Precision	Accuracy	TSR
Archer	C/C++	63	1	80	17	0.7875	0.9877	0.9844	0.8882	0.9360
Intel Inspector	C/C++	71	40	45	8	0.8987	0.5294	0.6396	0.7073	0.9535
ROMP	C/C++	59	11	73	18	0.7590	0.8876	0.8630	0.8256	1
ThreadSanitizer	C/C++	64	1	84	15	0.8101	0.9882	0.9846	0.9024	0.9545
Coderrect	C/C++	72	2	85	9	0.8889	0.9770	0.9730	0.9345	0.9767
Archer	Fortran	53	0	63	16	0.8154	1	1	0.9063	0.7711
Intel Inspector	Fortran	63	9	65	16	0.7975	0.8784	0.8750	0.8366	0.9217
ROMP	Fortran	62	12	61	18	0.7750	0.8356	0.8378	0.8039	0.9217
ThreadSanitizer	Fortran	52	0	65	13	0.8000	1	1	0.9000	0.7831
Coderrect	Fortran	66	11	64	15	0.8148	0.8533	0.8571	0.8333	0.9398

https://github.com/LLNL/dataracebench/wiki/Tool-Evaluation-Dashboard

Education: Online Training Compiler/Tool Developers and Users

- Problems:
  - Many tools requested by app teams but only limited labors available
  - Hard for users to try out tools: download, install, configure on a right platform
- Solution: dockerized cloud-based environment through a web browser
  - Direct access preinstalled tool environment, using only a browser
  - Play compilers/tools online
  - Cloud machines in backend



http://freeCompilerCamp.org/


Lawrence Livermore
National Laboratory