

*Minutes of the EuroLLVM 2016

"Polly - Loop Optimization I

nrastructure" BoF

*Agenda:

- * Status update
- * Open Agenda

*Status Update

- * Stability
 - * LNT builds continue
 - * ffmpeg nightly builds <http://fate.ffmpeg.org/> (32bit and 64 bit)
 - * Qualcomm production compiler, daily testing on various benchmarks including large system builds
 - Compiler available at <https://developer.qualcomm.com/software/snapdragon-llvm-compiler-android>
 - Polly is on at -O2 and higher. Performance is good. Either way whether it helps and not your benchmark, we would like to hear from you.
 - * University of Passau is using building Gentoo Package repository with Polli (a jit-compiler based on Polly)
 - * Darwin (Jack Howarth)
 - * Compile Time
 - * Currently http://llvm.org/perf/db_default/v4/nts/64426?compare_to=64367
 - * Move Polly into LNT <http://pollybof.grosser.fastmail.com.user.fm/pollybof/performance.png>
 - * New Optimizations
 - * Outer-loop Vectorization
 - * OpenCL Versioning (see CC)
 - * GPU Code generation (see tomorrow afternoon)
 - * Since November
 - * 310 commits
 - * 5000 lines changed in Polly / 8000 lines in isl
 - * 5 contributors with more than 20 commits
 - * Major new features
 - * Moving Polly into the pass pipeline (a lot of bug fixes)
 - * Invariant Load Hoisting
 - * De-LICM
 - * Memory accesses of different Types
 - * Outer-Loop Vectorization
 - * Improved output in isl
 - * New pass manager
 - * User provided `__assume()`
 - * Pollylabs: <http://pollylabs.org>

*Agenda

- * Testing coverage
 - * Increase test coverage and daily test code quality
- * New directions
 - * Better vectorization
 - * Make Polly as analysis available to LLVM (e.g. vectorizer)
 - * User provided annotations
- * Controlling Polly by pragmas?

*Questions from the Audience

Q: Are all the execution regression caused by Polly?

A: We don't know, some of the regressions are caused by increase of code amount.

Q: Can't use the Pass Manager to put it before LICM?

A: Currently Polly runs very early, which increases compile time. You can put it before LICM, just before the inliner loop. We want to run Polly as late as possible to reduce affect of Polly. We want to run in after the inliner loop because these might sometime remove the loop.

Q: How big are the changes to the pass manager for Polly. Why not run it so every pass would benefit from it.

A: It can be enabled by default in -O3, but rarely affects anyone. There are some "chaotic" performance that are impossible to sort all out.

Question to the audience: Who tried Polly?

~Most in audience

Question to the audience: Who want's a Blackbox tool which just improves things?

~half of those bore

Question to the audience: Who want to extract information out of Polly?

~ Again about half

Who want's to use it to implement own transformation?

~ Smaller number of people

Q: Experience from Polly. Would generate a lot of store-reload sequences. Suggestion: Add a Post-Polly normalization pass.

A: May try -polly-position=before-vectorizer. Could you send us an email about the concrete problem?

Q: Try not to run both, vectorizer and and Polly at the same interest because of competing interests

A: The unrolling pass is contra-productive to Polly

Q: Add #pragma novectorizer

A: There are separate concerns. Polly can't optimize single loops and don't touch it. Once a loop is touched, the loop vectorizer cannot transform it anymore. Polly does outer loop vectorization. For inner loops it just annotates the loop for the loop vectorizer to be vectorized. The choice of vectorizing (cost model) is entirely at the loop vectorizer.

Loop vectorizer will ignore cost model if it is told to vectorized, stopped by legality concerns.

Zino: Proposed to make -polly-position=before-vectorizer by default.

Q: Targeted transformations. E.g Loop distribute etc.

A: Such transformation could be strictly beneficial, by currently require the whole Polly chain to run. Good idea if people would write such targeted passes that make use of Polly.

Q: Suggestion to leave the inner loop transformation to LLVM. Use information in Polly whether if eg. loop is interchanged, it can be vectorized, etc.