

2011 Open64 Tutorial-Workshop Notes

- **C++0x thread support for Open64 Presentation**

- **Presenter: Hans Boehm; Scribe Eddie Gornish**

- A question was raised about the storing of bit fields, where an architecture can't write anything smaller than a byte.
- Answer - this is addressed in the standard.
-
- Question - can you still perform register promotion?
- Answer was yes, but you need to do it differently.
-
- Hans thought that most compilers don't have proper support for some/many features.
- Atomic support might be in gcc?
-
- Hans stated that:
- If you're compiler/architecture/... doesn't have threads, you don't really need to support this.
- But if your compiler/architecture/... does have threads, you should follow these conventions - single threaded code will find its way into multi-threaded applications.

-

- **Open64 Phase Componentization**

- **Presenter: Min Zhao; Scribe Feng Zhou**

- Question: Has the componentization framework been used/tested by third-party developers?
- Answer: Yes. One master student in University of Houston has been using it to implement some new optimizations.
-
- Question: Has the framework been used for existing optimization phase?
- Answer: No. Our current focus is to build and consolidate the infrastructure. We are using the framework for writing new optimizations. We haven't got resources to retrofit existing optimizations to this framework.
-
- Question: What happens if some optimization using the framework is outside the scope of WOPT and WSSA where some of the information (e.g. control flow graph) is not constructed yet?
- Answer: The framework will create the control flow graph.
-

- Question: Can the framework update CFG, WSSA etc. on the fly?
- Answer: Yes. The framework provides interfaces for incrementally update the CFG, etc. On the other hand, there is a flag in the component that can signal the framework to rebuild CFG etc. after current optimization. This is handy for phases that do a lot of changes to the IR or do not want to deal with incremental update.
-
- Question: Can the framework used for CG optimizations?
- Answer: Yes. It can be used.

•

•

- **WHIRL SSA enhancements**

- **Presenters: Jain-Xin Lai, Jaewook Shin; Scribe: Xiaomi An**

- Question1: Is the ssa form generated from HSSA equivalent to that built by the standalone builder?
- Answer: Since the standalone wssa builder uses the new alias analysis, they may not be equivalent.
-
- Question2: Is it possible that at some point during program transformation, the whirl tree is not synchronized with the WSSA?
- Answer: The WSSA should be maintained up to date during any program transformation. And we need to rebuild the whole SSA if needed.

•

- **Open64 performance optimizations**

- **Presenters: Michael Lai, Micahel Berg; Scribe Teresa Johnson**

- **OpenMP 3.0 future features/compiler/runtime**

- **implementation**

- **Presenters: Deepak Eachempati; Scribe Xiaohau Zhang**

- **Open64 Discussion on Future Areas of Development**

- **Presenters: Michael Lai, Luis Lozano, Min Zhao; Scribe Jaewook Shin**

- Michael Lai announced he's the Open64 Release Manager for the next release.
- We need feedback to promote collaboration in the Open64 community
- Min Zhao gave a CGSSA presentation to share the work that is currently taking place in this area
 - Questions: Register SSA based on TN?

- Answer: yes
- Comments shared:
 - What is good to do and useful to do as a hobby for those in the Open64 community?
 - Where does Open64 stand in terms of performance and robustness? Where are we going?
 - Open64 is not viewed only as a performance tool. Robustness is important in AMD's view.
 - A page is sourceforge 2000 downloads/month. People are asking how robust Open64 is but no answer. CG code is hard to read. 173 compiler directives. Part of robustness is also the code quality. We should make it cleaner. GCC has no directives but is hard to work on.
 - Gatekeepers should look for these things when reviewing codes.
 - We like to use Open64 for runtime compilation (JIT). Trend is toward the runtime compilation, architecture neutral, shifting toward dynamic optimization.
 - Request to Open64 community is to send what you'll be working on for next Open64 release to new Release Manager.
 -
 -

- **OpenCL**

- **Presenter: Professor Jenq-Kuen Lee, Yu-Te Lin, Ming-Yu Hung; Scribe Min Zhao**

- 1. How soon to converge in reducing the PPA graph?
- A: It depends on the number of nodes in the graph, since in each iteration, one node is reduced.
-
- 2. Why choose open64 for OpenCL, instead of LLVM as others?
- There is some historical reason - we have been using open64 since 2005. Also we think open64 has a better backend and we want to provide OpenCL community another compiler choice.
-
- 3. Suggestion to the problem that EBO does not consider register component:
- Have different TNs for register components inside EBO (this will require no change to EBO) and have a synthesis phase afterwards to reconstruct the register components.

- **SSA-based Algorithm for Optimal Speculative Code Motion under an Execution Profile**
- **Presenter: Hucheng Zhou; Scribe Eddie Gornish**
 - Question - Did you do a comparison of your PRE algorithm vs. no PRE at all?
 - Answer no - rather, compared to other PRE variants.
 - The question was asked because in embedded space PRE has been turned off - because it can have an impact on register pressure.
 - Reply - haven't really analyzed register pressure
 -
 - Request - Can you explain how to build an EFG?
 - Answer - An EFG is a flow network built from the vector redundancy graph. You form the FRG from the CFG, then delete any unnecessary loads, then build the EFG.
 - The author then showed slides that showed how to build an EFG.
 -
 - Question is every expression a candidate for your algorithm?
 - Answer - yes, this is similar to current PRE algorithms
 - A follow-up question - Shouldn't you only consider expressions, that are used more than once, as candidates?
 - Answer - that's why MCSSAPRE is better... - not sure if question was understood, answered...
 -
 - Hans - concern over speculative stores
 - Reply - in theory, there will be legality checks
 -
 - Comment that this code was not in Open64
 - Reply - after returning to Beijing, will submit patch
 -
 - Question could this algorithm be implemented using WHIRL SSA?
- Answer - Implementation was done in WOPT, but author saw no reason why it couldn't be implemented in WHIRL SSA.