Research Report

Monday 11/11/2013

Wei Wang

review of the issues to solve/What I needed to do

- Needed to understand the memory trace and let PoCC read in the trace and help memory analysis.
- Needed to work on a two pager describing the work I have been doing in the last 2 or 3 months (finished and sent)
- Others
  - Plos ONE (cardiac)
  - Lulesh (using INRIA PPCG)
  - PNNL benchmark

Progress on PoCC Memory Trace

- Ran the instrumented gemverOut.cpp
  - if N=4000, the output file becomes large (more than 2G)
  - Changed to smaller N for now
  - Tristan suggested the shape/dimension of the arrays should be considered as well
- Two related work from Clauss's group
  - Profiling Data-Dependence to Assist Parallelization: Framework, Scope, and Optimization (MICRO'12)
  - Online Dynamic Dependence Analysis for Speculative Polyhedral Parallelization (Euro-par'13)
  - Just got the idea, need to read in detail
- PoCC Code
  - Figured that Dependency Analysis should be the focus (CAnDL component of PoCC)
- No substancial progress achieved yet

The plan

- Getting to know how existing programs perform the memory/dependency analysis
- Getting to know its weakness
- Think where the memory trace should step in and help

Research Report

Thursday 11/07/2013

Wei Wang

review of the issues to solve/What I needed to do

- Needed to prepare and ship to you "one-icon click" version of Cygwin PoCC
- Needed to make MinGW work
- Others
  - Plos ONE (cardiac)
  - Lulesh (using INRIA PPCG)
  - PNNL benchmark

Progress on Windows PoCC

- Standalone Cygwin PoCC successfully prepared
  - This involved finding out what dlls and exe files (providing bash system on windows) are needed to enable a standalone version
  - Tested with different Windows platform on gemver.c (default test file)
- MinGW PoCC: not successful
  - Existing components of PoCC rely on POSIX compliant implementations offered by fork, wait, and pipe
  - MinGW, being minimalist GNU for Windows, just doesn't support the calls
  - PoCC has many places of calling posix functions like fork,pipe,and wait

The plan

- Focus on writing the 2-pager summarizing the work in the past 2-3 months.
  - Mini version of IMPACT submission
  - Brief mention of Windows version Polyhedral compiler
- Need also to work on PloS ONE re-submission (hopefully can put an end to the project)
  - Spend a few time on improving OpenACC performance on MIC (if not successful, should also be fine)
  - Text Edits

Research Report

Monday 11/04/2013

Wei Wang

review of the issues to solve/What I needed to do

- Needed to successfully build PoCC on CygWin
- Needed to build PoCC using mingw
- Whichever of the above two that provides the most easy way for others to try should be the focus
- Others
  - Plos ONE (cardiac)
  - Lulesh (using INRIA PPCG)
  - PNNL benchmark

Progress on Windows PoCC

- Cygwin successfully built
  - Update: it is possilbe to ship the cygwin dlls (only 4 involved) with PoCC.exe
  - Update: a little issue (though easy to solve) is PoCC.exe would call LINUX SCRIPTS to postprocess and assembel the output together
  - Planned action: also ship the SCRIPTS -- this needs a BASH system to also be shipped (bash.exe, find.exe, grep.exe from mingw can be a good choice)
- MinGW built
  - less library than Cygwin caused undefined variable
  - Still under investigation.
  - Set MinGW priority lower than making Cygwin-PoCC easy to try
  - After making Cygwin-PoCC work, immediately go back to this

The plan

- Cygwin "one-icon click" version ship to John soon
- MinGW-PoCC: try to make it work
- Others : depends on whether the above getting.

Research Report

Thursday 10/31/2013

Wei Wang

review of the issues to solve/What I needed to do

- Needed to make progress on Polyhedral compilers on Windows
- Needed to prepare some slides accompaning SC'13 UD Booth poster
- Others
  - Plos ONE (cardiac)
  - Lulesh (using INRIA PPCG)
  - PNNL benchmark

Windows Polyhedral Compilers

- No working version yet (Thursday 10/31)
  - 1. Cygwin Full Installation
  - 2. PoCC 1.2
  - 3. Fixed only two of three issues, when compiling PoCC in Cygwin (stumbling at the third issue)
    - Default Perl version in Cygwin too high, resolved by installing perl 5.10
    - two similar components have the same trivial variable array declared (resolved by manual renaming)
    - THIRD one: not able to resolve yet. Building clasttool components needed dynamic libraries (e.g. -lcloog-isl, -lpast), however PoCC compilation in Cygwin cannot generate dynamic libraries (dll?), native linux generated libcloog-isl.so and libpast.so

Others

- Prepared slides for SC13 UD Booth that go with the poster
- Others: no progress except cleaned up all results relating to IMPACT submission

The plan

- Figure out how to generate .so counter part in Cygwin for components like cloog, past etc.
- Try to compile component by component
- Learn how to develop DLLs in Windows Environment and apply to all components of PoCC?
- Turn to PPCG and use Visual Studio 2013?
- Need to work for the resubmission of PlosONE (30 days passed, recommended 45 days resubmission)

Research Report

Monday 10/28/2013

Wei Wang

review of the issues to solve/What I needed to do

- Short: submit the draft to Impact 2014
- Medium: submit plos one by November
  - This needs making the cardiac code more efficient
- Long: think about what should I do to extend the current impact 2014 draft to aim for a CONFERENCE paper and for my proposal

What progress did I make

- Submitted IMPACT 2014
- Prepared the poster for SC'13. The contents were all from IMPACT 2014 submitted paper

The Plan

- Cardiac Project
  - Work on getting more speedups on MIC (with the help of VTune)
  - Work on modifying the text
  - Get a ready version by next Monday and send out
- Energy Profiling
  - Prepare required slides for SC'13, accompanying the poster.
  - Start looking into the set of PNNL benchmarks

Research Report

Thursday 10/24/2013

Wei Wang

review of the issues to solve/What I needed to do

- IMPACT 2014 workshop on polyhedral compilation techniques due on 10/25/2013 AoE, needed to 1) finish related work and conclusion part 2) collaborate with John and Allan with the draft
- Not sure whether to include not-so-good Lulesh results to impact 2014 paper
- Manual implementation of cardiac code (for Plos ONE submission) has too much thread wait for OpenMP implementation

progress on Impact Workshop 2014

- Finished related work and conclusion before last Tuesday
- Worked with Allan improving the draft on Tuesday, Wednesday, and Today
- Thought that we better include lulesh results, even though it didn't get speedups. (at least it can support Time-Energy Correlation)
- Need to go through the draft and get it submitted before Sunday 8AM!

progress on lulesh

- Generated and ran 200 program variants with 100 maxfuse + 100 smart fuse (with different tiling size)
- Need to collect the results and add the resulting graph to the draft

progress on Plos ONE

- No progress from Monday
- Will dedicate next week to Plos ONE and aim to submit it before November.

The plan

- Short: submit the draft to Impact 2014
- Medium: submit plos one by November
  - This needs making the cardiac code more efficient
- Long: think about what should I do to extend the current impact 2014 draft to aim for a CONFERENCE paper and for my proposal....!!??

Weekly Research Report

Monday 10/21/2013

Wei Wang

review of the issues to solve/What I needed to do

- Although Lulesh can go through polyhedral compilers, no extensive experiments were done, no results (numbers /graphs) were generated
- IMPACT 2014 workshop on polyhedral compilation techniques due on 10/25/2013 AoE, need to 1) get lulesh results into paper & 2) get a ready draft to John by Monday.
- Manual implementation of cardiac code (for Plos ONE submission) is not well vectorized

progress on: IMPACT 2014 draft

- Added MIC results of Cardiac code to the draft
- Redo all parts of the draft except Related Work and Conclusion
- Sent the draft to John
- Need to work on related work and conclusion
- Depend on how many pages we want, we can add some discussion of lulesh challenges

progress on: Lulesh

- Generated program variants for one SCoP
  - that scop occupied ~12% execution in fully parallelized OpenMP
  - did not generate variants for more SCoP because the polyhedral compiler took too long to do transformation
  - The results on elo is that the variants performed as well as the original OpenMP implementation on Sandy Bridge
  - The original OpenMP program had slow down running on MIC.
  - Not planning to add lulesh results to IMPACT2014 draft

progress on: PlosONE

- Met with Will on profiling the cardiac code using Intel Vtune Amplifier
  - we found that the vectorization is at the OK range.
  - The problem is related to memory stalls and also the bandwidth is not 100% utilized.

The plan (before Thursday)

- Related work and conclusion section of IMPACT 2014 workshop paper
- Try to get good Lulesh numbers using polyhedral approach still
  - at least I will have the energy and time correlation data generated on Sandy Bridge (on MIC, the problem is aforementioned: the program is not performance portable yet: needs investigation)

Weekly Research Report

Thursday 10/17/2013

Wei Wang

What I needed to do (review)

- work on using PPCG compiler to transform Lulesh programs
  - worked on it (details in next slides)
- work on the vectorization of the cardiac code
  - Did not work on it yet.

What did I do (1)

- work on using PPCG compiler to transform Lulesh programs
  - Riyadh's PPCG could auto-parallelize (involving scalar privatization) the sequential code while tiling it. But the problem is: the time it took to auto-parallelize is at the magnitude of hours!
  - Solution: ?

What did I do (2)

- Polyhedral optimize cardiac code for input size smaller than 2048, on Xeon Phi
  - Previously, input size 2048 had barely speedups (about 3-5% improvement)
  - Found that for smaller input size, the improvement was about 15%. For example, size 1024, the 110X speedup got improved to 110*(1+15%) = 130X speedup by using polyhedral transformations. (Graphs attached in the beginning)

What do I plan to do?

- Get lulesh auto-parallelized and transformed for MIC, put results into paper draft
- Get a ready to submit version to you by Sunday (10/20/2013) for IMPACT 2014 workshop on polyhedral compilation techniques (deadline 10/25/2013 AoE)
- Get some progress on improving manual implementation of cardiac code.
- Start looking at Windows PoCC/PPCG