Implementation of Additional Optimizations for VPO on the Power Platform

Chungsoo Lim, Gary A. Smith, Won So
{clim, gasmith, wso}@ncsu.edu
http://www4.ncsu.edu/~gasmith/csc791a_proj

Spring 2005 CSC791A Project Progress Report
11 April 2005

Work Completed To Date

This section contains a detailed analysis of our progress to-date. Each of our three assigned optimizations was investigated individually by team members. The team met to discuss high-level issues involving cooperation of optimizations. We also shared insights regarding how our individual implementations operate. Our group met at regular intervals to evaluate team progress and ensure that the workload breakdown was fair to all team members. Group member contributions are summarized in the following subsections.

Chungsoo Lim

Chungsoo investigated the common subexpression elimination compiler pass, especially local common subexpression elimination. The existing local CSE code within VPO is located in powerpc-vpo/vpo/lib/cse.c.

To verify the correctness of local CSE, Chungsoo employed two approaches. The first approach was to read the corresponding code and try to find any errors. The second one is to make up some test codes, compile them, and compare RTLs before the optimization with RTLs after the optimization. By just reading the code, it was hard to determine if the code is correct because of the two reasons. First of all, it was hard to understand the code because exact algorithm used in local CSE was not known. The second reason is that there may still be some odd cases unhandled. However, through this process, Chungsoo obtained some insight about the code and it will help in the later part of the project. To test the local CSE, Chungsoo made up several test cases, each of which consist of only one basic block. Chungsoo also modified the existing code to have RTLs printed out, so that RTLs before and after the local CSE can be compared with each other. Chungsoo compared both RTLs and assembly codes and found evidence proving that local CSE worked fine at least for the test codes.

Gary Smith

Gary is responsible for the liveness analysis optimization. Existing VPO source code was evaluated to discover the status of existing functionality. Opt-compatible liveness source code, provided by Dr. Mueller, and Gary's past Opt-compatible liveness implementation were evaluated to determine if their integration into the existing Power VPO package would be beneficial.

The existing live_variable_analysis code within VPO, located in powerpc-vpo/vpo/lib/dataflow.c, is more comprehensive than the aforementioned Opt-compatible examples. The VPO implementation addresses liveness when delay slots have been filled. This action requires that the last instruction in predecessor blocks be examined in addition to the code in the current block. From a high level, the component parts of all three implementations are identical. All three clear any old dataflow accounting information in basic blocks, examine each RTL, sets use/def vectors, and performs the iterative liveness analysis algorithm. The VPO implementation creates use/def information based on an intermediate representation instead of assembly instructions, as is the case with Opt.

Gary learned how to emit RTL and assembly files from the VPO compiler suite. Although debugging output was available from the VPO compiler tool, Gary created custom functions to output liveness information. For instance, writing custom routines to output control flow information, RTL instructions, and liveness data was more efficient than utilizing the existing code in the VPO package. Test codes were authored and utilized to ensure that liveness analysis is functional.

Gary is responsible for coordinating the project website and technical documentation.

Won So

Won is investigating the evaluation order determination (EOD) compiler pass. The test results by enabling EOD with "-VV" option shows that EOD routine have some problems. For some benchmarks, compilation does not finish. Won is responsible of fixing this bug. The existing EOD code within VPO is located in powerpc-vpo/vpo/lib/eod.c. EOD processes are composed of two processes: The first is to construct links between RTLs and compute costs of RTLs. The second is to reorder RTLs so that the operations with higher costs are evaluated first.

By writing the custom function, which emits RTLs before and after performing this optimization, Won examined the problems in current EOD implementation. Testing with simple operations such as series of summations and multiplications does not expose the problem and the changes in RTLs before and after EOD look right. By testing with the files in the benchmark, which end up with infinite compilation, 2 problems were found in the code and resolved. The first problem was the wrong link constructed by ruage() function in the 1st process. Some RTLs have wrong links to future instruction. Won fixed this bug by adding a condition before calling add_link() function. The second problem happened while reordering instructions. The while loop calling reoder() could not escape from it because it was looping with the same RTL. This problem might be cause by a bug in the reorder() function. It was fixed by adding one condition inside this loop so that it can avoid traversing the same item infinitely.

Though infinite looping problems were fixed, the compiled binary is suffering segmentation fault in some cases. This does not seem to be an easy problem. The reorder routine is still suspected and will be examined thoroughly.

Open Issues

Our project will be completed once the evaluation order determination and global CSE compiler are completely functional. Since Gary has completed the live variable analysis investigation, he can help with the other tasks on the project. Chungsoo and Won will continue in their present roles. The specific problems encountered for each item are as follows.

Global CSE

- An investigation of this routine was started during this phase of the project; however, emphasis was placed on local CSE. As a result, global CSE is still an open issue.

Evaluation Order Determination

- A bug in the reorder routine seems tricky to fix.

Project Completion Milestones

Our proposed plan for project completion is summarized in Table 1.

Date	Milestones
11 April 2005	Completion of This Progress Report
18 April 2005	Estimated 80% on Global CSE and Eval Order Determination
25 April 2005	100% on All Tasks, Complete Final Testing

Table 1: Project Completion Plan

Disclaimer

This paper and associated software changes are intended for **informational purposes only**. Therefore, any use of the information presented in this student work is at your own risk. Chungsoo Lim, Gary A. Smith, and Won So provide **no warranties of any kind** surrounding the use of this material.