Timothy Garnett

6.191 Preliminary Masters of Engineering Proposal

12/07/2001

A Dynamic Optimization Scheme for Windows

I intend to do my Masters of Engineering thesis with the Commit Group. The Commit Group (**Com***pilers*@**mit**) seeks to discover novel approaches to improving the performance of modern computer systems without greatly increasing the complexity faced by application developers, compiler writers, or computer architects. To this end they have a number of projects ongoing mostly in the broad discipline that can be considered compilers. Members of the group are working on diverse projects from developing compiler techniques for the unique RAW architecture to automatic compilation techniques to make full use of SIMD instructions found in many modern processors as part of multimedia extensions. They are looking for Masters of Engineering students to work on various projects including a project to develop a high level programming language specifically designed to work with streaming processes and (the area I'll be working on) a system for performing dynamic optimizations on running applications in the Windows/x86 platform. The leader of the Commit Group is Saman Amarasinghe and he will be my advisor for this project.

The idea of dynamic optimization of applications is motivated by the fact that it has become increasingly difficult to achieve high levels of efficiency during static compilation. The widespread use of object-orientated programming languages and the practice of shipping applications as collections of Dynamically Linked Libraries (DLL's) as opposed to monolithic binaries has resulted in a greater degree of runtime binding, which while advantageous from an engineering perspective, makes static compilation optimizations difficult if not impossible. On the opposite side the movement in hardware has been to offload more complexity to the compiler (CISC to RISC to SIMD etc.) asking it to take on more of the performance burden. This squeeze of increasing performance burden and increasing obstacles towards static compilation calls for new approaches. The advantage of dynamic optimizations is that they occur after runtime binding and have access to information gleaned as the program executes. The major challenge in designing dynamic optimizers is maintaining complete control over the application including over abnormal control flow such as exceptions.

Several members of the Commit Group in cooperation with Hewlett-Packard laboratories have developed a dynamic optimization framework for the IA-32 family of architectures running the Windows operating system based on Dynamo (an earlier Hewlett-Packard dynamic optimization project). The system is similar to a caching dynamic translating system, which is where most research on dynamic optimizations has occurred. The program interprets the applications code one basic block at a time and copies it into a block code cache, the copied block is then executed natively (as opposed to emulated because of the complexity of the IA-32 instruction set). The terminating branches of the basic blocks are modified to return control to the dynamic optimizations system. In frequently executed sections of the code traces are built and kept in a separate trace cache and the code in the two caches is linked together allowing longer stretches of code for optimization and improving the processor's instruction cache performance. There are a host of other complications, most specific to the Windows operating system. One of the more potentially problematic ones is with exception handlers. Exception handlers have access to the machine context from when the exception occurred, thus the system must transform the context passed to the exception handler to make it appear that the exception occurred in the original code. This is not always possible in the presence of certain types of reordering optimizations, however in practice exception handlers very rarely look at the machine context and no problems have been observed in the current system.

The current framework is complete, but not yet highly optimized. No optimizations of the cached native code are performed yet. The performance hit from running the dynamic optimization framework ranges from a couple percent to a 150 percent on SPEC2000 and office benchmarks. The addition of optimizations for the application code should speed things up significantly. The original Dynamo system was developed for PA-RISC platforms and there showed net speed improvements of up to 20%, even on highly optimized binaries. Achieving that level of speedup in the current framework is made difficult by the overhead of decoding and subsequently encoding IA-32 instructions (as compared to a RISC style instruction set). Indeed this is one largest sources of overhead in the current framework, though room for additional optimizations certainly exists.

For my thesis, I will be developing and implementing optimizations for cached application code using the existing framework. The plan is for the optimizer to run as background thread (advantageous in a multiprocessor environment since the optimizations can be run on a different processor from the main application). The focus will be on optimizations that are difficult if not impossible for a static compiler to perform such as optimizations across procedure call/return boundaries, indirect branches, and virtual method calls. The goal is to approach the performance level of the Dynamo system, but on a Windows/IA-32 framework. Systems like this are also very useful for developing virtual machines as they take care of much of the backend work.

The current plan for the schedule for the completion of my Masters of Engineering thesis is for me to work as a UROP with the Commit Group over spring semester to get aquatinted with the group and the existing framework for my project and to research potential optimizations for later in the project. Over the summer, fall, and next IAP I will work on designing and implementing optimizations for the system, with the following spring to be devoted to composing my thesis. This schedule is of course preliminary. Professor Amarasinghe said that it is very likely that I will get one semester of research assistant funding (contingent on whether some funding for the group comes through or not). For the other semester I plan to work as a TA to cover tuition costs. Additional expenses are likely to be minimal and occur mainly in the form of computer time.

There are certain risks associated with this endeavor. One, of course, is the possibility that things will not work out well between the group and I and I'll be forced to abandon the project at some point. There is also the risk that funding will fail to come through for an research assistantship in which case I would need to TA an extra term, cutting into time for working on the project. Another area of concern is that a large part of the dynamic optimization framework is adapted from code provided by Hewlett-Packard and thus they have some say on how the system will be used. A final risk is that the nature of the framework and Windows operating system will severely limit possible optimizations for some, as of yet, unforeseen reason. The project does have the advantage of being somewhat open ended, the number and scale of the optimizations implemented is not set and can scale up or down depending on how things develop without compromising the eventual thesis overly much. In the whole this is a very interesting and exciting project and I look forward to working on it.