

Статический анализ Си программ в системе SAPFOR для их распараллеливания

Н.А. Катаев¹, Ю.А. Лапенко²

¹ИПМ им. М.В. Келдыша РАН, ²МГУ им. М.В. Ломоносова

Ключевые слова и фразы: анализ и преобразование программ, автоматизация распараллеливания, LLVM, DVM, SAPFOR

При разработке параллельных программ приходится выбирать, какие технологии параллельного программирования использовать, чтобы задействовать все доступные возможности современных высокопроизводительных вычислительных систем. Низкоуровневые модели параллельного программирования (MPI, CUDA, OpenCL, POSIX Threads) позволяют достичь максимальной эффективности выполнения параллельных программ, но существенно усложняют их разработку и отладку, заставляя пользователя разбираться как в деталях каждой из используемых моделей, так и в особенностях вычислительной системы. Существующие же на данный момент автоматически распараллеливающие компиляторы в подавляющем большинстве случаев не позволяют добиться приемлемого распараллеливания для реальных вычислительных комплексов.

Возможным решением может быть применение смешанного подхода [1], который опирается на использование высокоуровневых моделей параллельного программирования для описания параллелизма в программе в сочетании с автоматизацией наиболее трудоемких и приводящих к ошибкам этапов распараллеливания. Высокоуровневые модели (OpenMP, OpenACC, DVMH) снижают сложность разработки параллельных программ, но в тоже время обеспечивают высокую эффективность их выполнения. При этом пользователь получает возможность осуществлять контроль над ходом автоматизированного распараллеливания программ, в том числе задавая свойства анализируемых программ и подсказывая необходимость автоматического выполнения того или иного преобразования.

В качестве высокоуровневой модели параллельного программирования может выступать DVM [2, 3], которая предоставляет директивные расширения для стандартных последовательных языков программирования C и Fortran и позволяет описать параллелизм как между узлами вычислительной системы, так и внутри каждого узла, в том числе, используя возможности ускорителей.

Система SAPFOR (System FOR Automated Parallelization) [4, 5] может выступать в качестве средства автоматизации параллельного программирования, обеспечивая возможность автоматического распараллеливания «хорошо» написанных потенциально параллельных программ. При этом SAPFOR предоставляет пользователю возможности для автоматизированного преобразования программ и приведения их к потенциально параллельному виду. Возможность выполнять статический анализ программ является неотъемлемой частью системы SAPFOR, необходимой как для автоматического распараллеливания и преобразования программ, так и для предоставления пользователю описания свойств программ, влияющих на возможность их распараллеливанию.

Применение системы SAPFOR для распараллеливания некоторых приложений из пакета NAS Parallel Benchmarks [6] выявило ограничения реализованных в системе SAPFOR техник анализа программ [5]. В докладе будут рассмотрены новые техники анализа программ, реализованные в SAPFOR и позволяющие справиться с выявленными ограничениями. Важной особенностью является то, что предложенные техники опираются на возможность системы SAPFOR модифицировать внутреннее представление программы в виде LLVM IR [7] в процессе ее анализа для уточнения свойств исходной непретобразованной программы [4]. В докладе будут рассмотрены: (1) подход к анализу зависимостей по данным в многомодульных программах за счет выборочной подстановки функций в точках их вызова; (2) подход к определению редуцированных операций в циклах программы, для случаев, когда имеющихся возможностей LLVM по их анализу оказывается недостаточно; (3) подход к определению приватизируемых массивов в циклах про-

граммы. Все предварительные преобразования программы, необходимые для ее анализа, выполняются над LLVM IR, скрыты от пользователя и не изменяют исходный код программы на языке высокого уровня. Необходимость выполнения данных преобразований определяется системой SAPFOR автоматически на основе предварительного анализа распараллеливаемой программы.

Расширенные возможности статического анализа программ будут продемонстрированы на примере некоторых приложений из пакета NAS Parallel Benchmarks.

Список литературы

- [1] Kataev N., Kolganov A. (2021) Additional Parallelization of Existing MPI Programs Using SAPFOR. In: Malvshkin V. (eds) Parallel Computing Technologies. PaCT 2021. Lecture Notes in Computer Science, vol 12942. Springer, Cham, 2021 — P. 41-52 — doi:10.1007/978-3-030-86359-3_4
- [2] Konovalov, N.A., Krukov, V.A., Mikhajlov, S.N., Pogrebtsov, A.A. Fortan DVM: a Language for Portable Parallel Program Development // Programming and Computer Software. Vol. 21, No. 1. 1995. — P. 35--38.
- [3] Бахтин В.А., Клинов М.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Сазанов Ю.Л. Расширение DVM-модели параллельного программирования для кластеров с гетерогенными узлами // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика, 2012. № 18(277). — С. 82-92.
- [4] Kataev N.A. Application of the LLVM Compiler Infrastructure to the Program Analysis in SAPFOR // Voevodin V., Sobolev S. (eds) Supercomputing. RuSCDays 2018. Communications in Computer and Information Science, vol 965. — Springer, Cham, 2018 — P. 487-499. — doi:10.1007/978-3-030-05807-4_41
- [5] Kataev, N.: LLVM based parallelization of C programs for GPU. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2020. CCIS, vol. 1331, — Springer, Cham, 2020 — P. 436-448. — doi:10.1007/978-3-030-64616-5_38
- [6] NAS Parallel Benchmarks. URL: <https://www.nas.nasa.gov/publications/npb.html>
- [7] Lattner, C., Adve, V. LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation //Proc. of the 2004 In-

ternational Symposium on Code Generation and Optimization (CGO'04). Palo Alto, California, 2004. — doi: 10.1109/CGO.2004.1281665