

Использование динамического анализа при распараллеливании на гибридный вычислительный кластер

Н.А. Катаев¹, А.А. Смирнов¹

¹ИПМ им. М.В. Келдыша РАН

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

Стандарт MPI широко используется при разработке программ для вычислительных систем с распределенной памятью, но его оказывается недостаточно, чтобы полностью задействовать параллелизм в узлах гетерогенного вычислительного кластера. В результате разработчик вынужден дополнительно использовать технологии параллельного программирования, такие как OpenCL, CUDA, OpenMP, DVMH [1, 2] и др., чтобы описать параллелизм внутри узла. Стоит отметить, что технология DVMH может быть также использована вместо MPI при распараллеливании последовательных программ. Использование меньшего количества технологий параллельного программирования, упрощает разработку параллельных программ, с точки зрения трудоемкости изучения используемых технологий, а также сопровождения и развития параллельной программы. При этом даже применение высокоуровневых директивных моделей требует понимания информационной структуры распараллеливаемой программы. Таким образом, все большую актуальность получает вопрос разработки инструментов автоматизации параллельного программирования. Полуавтоматические системы, такие как система SAPFOR (System FOR Automated Parallelization) [3, 4], стремятся автоматизировать как можно больше этапов разработки параллельной программы, но оставляют за пользователем возможность управлять процессом распараллеливания при необходимости.

При распараллеливании на графический ускоритель необходимо выделить участки кода (вычислительные регионы), которые могут быть выполнены на GPU. При этом помимо выполнения анализа, отвечающего за возможность параллельного выполнения того или иного фрагмента программы (например, гнезда циклов), необходимо определить набор данных, которые должны быть переданы на графический ускоритель до начала вычислений, и загружены с него в оперативную память, когда вычисления будут завершены.

Конструкции исходного языка, используемые в вычислительных регионах, выполняемых на ускорителях, часто подвергаются достаточно жестким ограничениям, что упрощает статический анализ, отвечающий за определение возможности их параллельного выполнения. Кроме того, данный анализ выполняется достаточно локально и сосредоточен в рамках ограниченной области анализируемой программы.

При этом внерегионное пространство может содержать произвольный код, что существенно усложняет статический анализ программы, необходимый для определения передаваемых данных между оперативной памятью и памятью ускорителя, а также точек программы, в которых должны быть выполнены соответствующие обмены. Основное влияние на статический анализ оказывают сложный граф потока управления программы, который может зависеть от входных данных программы, задаваемых пользователем, а также наличие адресной арифметики в программе и косвенных обращений к данным, которые должны быть переданы на устройство. В результате статический анализ должен принимать консервативные решения, что в результате приводит к наличию избыточных коммуникаций в программе между оперативной памятью и памятью вычислительных устройств. Это в свою очередь может приводить к тому, что накладные расходы на коммуникации будут перекрывать выигрыш от использований устройств, таких как графические ускорители.

В докладе будет рассмотрен подход к применению смешанного анализа программ, используемый при разработке системы SAPFOR, который позволяет оптимизировать расстановку спецификаций параллелизма в программе, за счет совместного применения как статического анализа, так и динамического анализа, выполняемого на разных наборах входных данных.

Список литературы

- [1] Бахтин В.А., Клинов М.С., Крюков В.А., Поддериюгина 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.
- [2] 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
- [3] 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
- [4] 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