

Аннотация

Многие системы автоматического конструирования программ используют исполнительную систему для исполнения программ. Исполнительная система – это управляющий код, который обеспечивает работоспособность программы. Работа исполнительной системы добавляет накладные расходы, потому что использует вычислительные ресурсы компьютера, так и расходует память. Для уменьшения накладных расходов исполнительной системы применяется техника сбора и воспроизведения трассы программы. Трасса является продуктом трассировки, а трассировка - процесс исполнения программы, включающий сбор информации о последовательности исполненных команд, с последующим сохранением в файл или память. Трасса может быть воспроизведена специальной легковесной исполнительной системой, которая называется проигрыватель трасс. Балансировка трассы – процесс модификации трассы путём перераспределения нагрузки в трассе. При воспроизведении трассы из процесса исключается часть накладных расходов, связанных с исполнительной системой, что создаёт «пустоты», которые можно заполнить путём балансировки трассы. Цель работы – разработка средств оптимизации исполнения фрагментированных программ на основе трассировки путём балансировки трасс. Был разработано и реализовано средство повышения эффективности исполнения фрагментированных программ на основе трассировки. Разработанный алгоритм балансировки трассы интегрирован в систему фрагментированного программирования LuNA в качестве модуля. Тестирование алгоритма на задачах численного моделирования показало, что использование разработанного алгоритма улучшило производительность фрагментированных программ.

Тема работы

Оптимизация исполнения фрагментированных программ на основе трассировки.

Состав коллектива:

Лямин Артём Сергеевич, магистрант НГУ

Научный руководитель: Перепёлкин Владислав Александрович, старший преподаватель ФИТ НГУ, научный сотрудник Лаборатории синтеза параллельных программ ИВМиМГ

Научное содержание работы:

1. Постановка задачи

Для подсистемы проигрывателя трасс [2] в системе LuNA [1] разработать алгоритм статической балансировки нагрузки с применением профилировочной информации на основе предыдущих запусков. Ожидается, что корректировки, вносимые в трассу, позволят сбалансировать нагрузку между вычислительными узлами, что значительно ускорит исполнение трасс в подсистеме проигрывателя трасс.

2. Современное состояние проблемы.

На данный момент существуют и развиваются проекты, посвящённые автоматическому конструированию параллельных программ. Среди них - Charm++ [3], PaRSEC [4], Legion [5] и LuNA [1]. В LuNA ведутся разработки по оптимизации исполнения фрагментированных программ, в частности, статическая балансировка на основе профилирования [6]. Также прорабатывается возможность оптимизировать саму трассу, находить и устранять в ней дисбалансы нагрузки, уменьшать число коммуникаций и т.д. Данная работа прорабатывает это направление.

Список литературы:

1. Malyshkin VE, Perepelkin VA (2011) LuNA fragmented programming system, main functions and peculiarities of run-time subsystem. PaCT 2011, vol 6873. LNCS Springer, Heidelberg, pp 53–61.
2. Malyshkin, V., & Perepelkin, V. (2021). Trace-Based Optimization of Fragmented Programs Execution in LuNA System. В V. Malyshkin (Ред.), Parallel Computing Technologies - 16th International Conference, PaCT 2021, Proceedings (стр. 3-10). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Том 12942 LNCS). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-86359-3_1.
3. Kale, L. V. Parallel Science and Engineering Applications: The Charm++ Approach / L. V. Kale , A. Bhatele. – CRC Press, 2013. – 314 P.
4. Danalis, A. PaRSEC in Practice: Optimizing a legacy Chemistry application through distributed task-based execution / A. Danalis, H. Jagode, G. Bosilca, J. Dongarra // 2015 IEEE International Conference on Cluster Computing. – 2015. – P. 304 – 313.
5. Bauer, M. Legion: Expressing Locality and Independence with Logical Regions / M. Bauer, S. Treichler, E. Slaughter, A. Aiken // SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis – 2012. – Article N 66 – P. 1 – 11.
6. Лямин, А. С. Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе профилирования / А. С. Лямин // Информационные технологии : Материалы 59-й Междунар. науч. студ. конф. 12–23 апреля 2021 г. / Новосиб. гос. ун-т. — Новосибирск : ИПЦ НГУ, 2021. — С. 114.

3. Подробное описание работы, включая используемые алгоритмы

Предлагаемый алгоритм балансировки трассы состоит из следующих шагов:

1. Получить профиль исполненной LuNA программы.
2. Проанализировать профиль программы и на его основе сформировать корректирующее распределение фрагментов вычислений – список фрагментов вычислений, которые нужно переместить на другие вычислительные узлы.
3. Произвести балансировку трассы в соответствии с корректирующим распределением и сохранить корректность трассы.

В системе LuNA реализовано логирование событий при исполнении LuNA программы. Событиями являются переходы фрагментов вычислений между состояниями, пересылка фрагментов данных между вычислительными узлами. Каждое событие имеет временную метку, что позволяет определить по набору событий, когда заданный фрагмент вычислений нагружал конкретный вычислительный узел. Совокупность данных о нагрузке позволяет оценить общую вычислительную нагрузку на вычислительные узлы.

Существующий алгоритм формирования корректирующего распределения для LuNA программ [21] был адаптирован для данной задачи. Шаги работы алгоритма:

1. Сбор информации о промежутках исполнения фрагментов вычислений.
2. Квантование всего времени работы программы – разбиение на равные участки времени, размер которого называется величина квантования.
3. Определение суммарной нагрузки фрагментов вычислений на каждый вычислительный узел в каждый квант.
4. Определение дисбаланса нагрузки вычислительных узлов компьютера.

5. Устранение дисбаланса нагрузки на вычислительные узлы путём перемещения фрагментов вычислений с перегруженных узлов на недогруженные.

Данный алгоритм не учитывает информационные зависимости между фрагментами вычислений, поэтому теоретически полученная картина нагрузки на вычислительные узлы может отличаться от реальной, так как перемещение фрагментов вычислений между вычислительными узлами может добавить коммуникации между вычислительными узлами, которые будут увеличивать нагрузку на вычислительные узлы компьютера.

Алгоритм балансировки трассы использует полученную ранее трассу и корректирующее распределение. Шаги работы алгоритма балансировки трассы:

1. Прочитать полученную ранее трассу.
2. Переместить фрагменты вычислений между трассами вычислительных узлов в соответствии с корректирующим распределением фрагментов вычислений на вычислительные узлы.

Разработанный алгоритм генерирует корректную трассу на основе профиля исполнения программы и полученной ранее трассы.

Алгоритм не даёт гарантий, что сбалансированная трасса будет исполняться эффективнее, чем та, что была получена ранее. Предложенный алгоритм будет устранять крупные дисбалансы вычислительной нагрузки, но в более меньших масштабах может давать ухудшение производительности. Во многом это зависит от настройки алгоритма генерации корректирующего распределения для каждой конкретной задачи: необходимо правильно подобрать параметры – величину квантования и пороговое значение, при котором состояние системы будет распознано как дисбаланс. При правильно подобранных параметрах алгоритм генерации корректирующего распределения во многих случаях будет устранять крупные дисбалансы нагрузки и игнорировать более мелкие дисбалансы, что приводит если не к увеличению производительности, то к незначительному уменьшению производительности воспроизведения трассы.

Стоит отметить, что алгоритм балансировки может быть применён повторно, что приведёт к устранению дисбалансов, полученных на предыдущем шаге балансировки. Итеративное применение балансировки к трассе может постепенно приводить распределение фрагментов вычислений по вычислительным узлами в некоторую окрестность локального оптимума, в котором производительность воспроизведения будет повышаться.

Разработанный алгоритм позволяет гибко использовать в своей реализации другие алгоритмы коррекции распределения фрагментов вычислений по вычислительным узлами. При этом остальной алгоритм остаётся неизменным.

4. Полученные результаты.

Был разработан алгоритм балансировки трасс. Реализованный алгоритм реализован и встроен в систему фрагментированного программирования LuNA. Было проведено экспериментальное исследование алгоритма балансировки трасс, которое показало уменьшение время исполнения LuNA программ в следствие балансировки трасс. Результаты приведены в таблицах.

Таблица 1 – Результаты тестирования на программе без статической балансировки фрагментов вычислений

Параметры	Время исполнения, с.
-----------	----------------------

Размер фрагмента данных	Число ядер	LuNA	LuNA Trace Playback	LuNA Balanced Trace Playback (Величина квантования алгоритма генерации корректирующего распределения, с.)		
				0,1 с.	1 с.	10 с.
48	64	60,2	58,6	17,3	17,6	18,6

Таблица 2 – Результаты тестирования на программе со статической балансировкой фрагментов вычислений на все вычислительные узлы

Параметры			Время исполнения, с.			
Размер фрагмента данных	Число ядер	LuNA	LuNA Trace Playback	LuNA Balanced Trace Playback (Величина квантования алгоритма генерации корректирующего распределения, с.)		
				0,1 с.	1 с.	
48	64	17,7	17,8	18,7	18,6	

Таблица 3 – Результаты тестирования на программе, моделирующей поведение гравитирующей газовой системы методом частиц в ячейках

Параметры			Время исполнения, с.			
Размер сетки	Число частиц	Число ядер	LuNA Trace Playback	LuNA Balanced Trace Playback (Величина квантования алгоритма генерации корректирующего распределения, с.)		
				0,1	5	
150 ³	10 ⁶	64	30,2	121,1	29,3	

Таким образом, практическая ценность алгоритма балансировки трасс подтверждается результатами экспериментов – время воспроизведения сбалансированных трасс уменьшилось по сравнению с оригинальными трассами.

5. Эффект от использования кластера в достижении цели работы

Кластер был использован для проведения тестирования эффективности реализованного алгоритма балансировки трасс, а именно исполнения LuNA программ, для балансировки и воспроизведения трасс LuNA программ. LuNA система, позволяющая работать на вычислителях с распределённой памятью, поэтому важным было использование именно кластера.

6. Перечень публикаций, содержащих результаты работы

1. Лямин А. К. Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе трассировки / Лямин А. С. // Информационные технологии : Материалы 61-й Междунар. науч. студ. конф. 17–26 апреля 2023 г. / Новосиб. гос. ун-т. — Новосибирск : ИПЦ НГУ, 2023 (Принято в печать).