

Сергеев А.А.

Институт прикладного системного анализа НТУУ “КПИ”, Киев, Украина

Параллельное генетическое программирование на графических процессорах

При поиске оптимального решения с помощью генетического программирования (ГП) для оценки новых решений (программ) требуются значительные вычислительные ресурсы. Задача усложняется, когда оценку программ необходимо проводить в течении многих поколений (в контексте эволюционного алгоритма).

Поскольку входные данные в процессе поиска остаются неизменными, оценка новых решений может быть осуществлена параллельно [1]. Можно выделить следующие стратегии реализации параллельного ГП: на уровне команд, одна команда – множество данных (ОКМД); на уровне программ, одна программа – множество данных (ОПМД); на уровне поколений, множество программ – множество данных (МПМД).

К аппаратным решениям, поддерживающим параллельную реализацию ГП, относятся центральные и графические процессоры. Центральные процессоры имеют сложную архитектуру (кэширование, конвейер команд), которая сокращает время выполнения арифметических и логических операций для последовательных данных. Ускорение на графических процессорах достигается за счет увеличения количества вычислительных блоков (пропускной способности), и как следствие, количества обработанных данных в единицу времени [2].

Увеличение количества процессоров в современных графических ускорителях (ГУ), а также возможность их коммуникации между собой [3], позволяют достичь выигрыша по времени при параллельной оценке программ в ГП. На рисунке 1, изображена абстракция архитектуры ГУ.

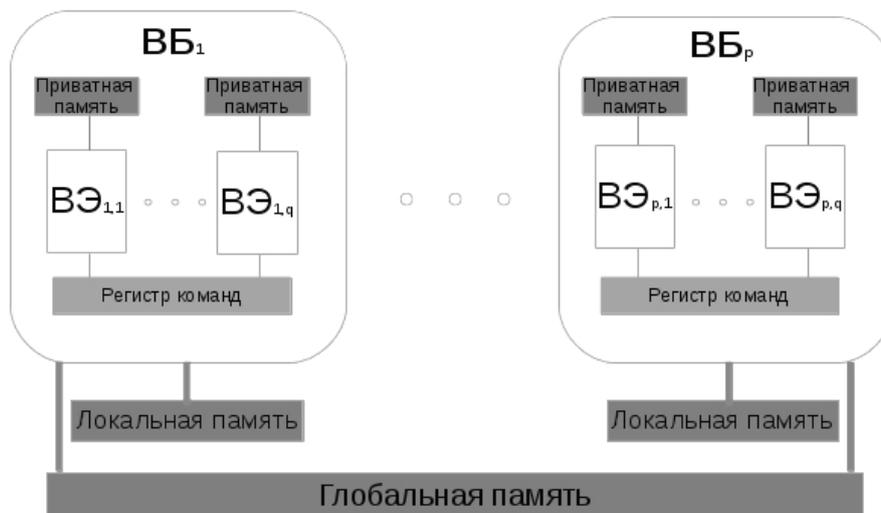


Рис. 1. Абстракция архитектуры ГУ

Из рисунка 1 следует, что ГУ состоит из массива вычислительных блоков (ВБ), каждый из которых содержит массив вычислительных элементов (ВЭ). Данная архитектура позволяет разделить выполняемую программу на меньшие составляющие, обеспечив этим масштабируемость эволюционного алгоритма. В самом простом случае (ОКМД) все вычислительные элементы ВБ выполняют одну команду.

В таблице 1 приведено сравнение стратегий распараллеливания ГП для набора входных данных $x[1] \dots x[m]$. Сокращения в таблице: ВБ - вычислительный блок (графический процессор состоит из массива ВБ), ВЭ - вычислительный элемент (ВБ состоит из массива ВЭ).

Выбор стратегии распараллеливания должен быть обусловлен его эффективностью по сравнению с последовательной реализацией ГП. Следует обратить внимание на соотношение количества входных данных и количества генерируемых программ. В таблице 2 приведен

Таблица 1. Сравнение стратегий распараллеливания ГП

Название стратегии	Входные данные	Выполнение программ
ОКМД	В каждый момент времени все данные $x[k]$ распределяются между всеми $V_{i,j}$, где $k \in [1, pq]$, $i \in [1, p]$, $j \in [1, q]$	В каждый момент времени во все $V_{i,j}$ записывается одна команда программы $P[k]$, где $k \in [1, pq]$, $i \in [1, p]$, $j \in [1, q]$
ОПМД	В каждый момент времени на все $V_{i,j}$ подается одно значение $x[k]$, где $k \in [1, pq]$, $i \in [1, p]$, $j \in [1, q]$	В каждый момент времени во все ячейки $V_{i,j}$, записывается по одной команде из всех программ $P[k]$, где $k \in [1, pq]$, $i \in [1, p]$, $j \in [1, q]$
МПМД	В каждый момент времени все данные x параллельно распределяются между $V_{i,j}$ заданного VB_k , где $k \in [1, pq]$, $i \in [1, p]$, $j \in [1, q]$	В каждый момент времени одна команда программы $P[k]$ записывается во все $V_{i,j}$ заданного VB_k , где $k \in [1, pq]$, $i \in [1, p]$, $j \in [1, q]$

анализ стратегий распараллеливания ГП.

Таблица 2. Анализ стратегий распараллеливания ГП

Название стратегии	Преимущество	Недостаток
ОКМД	Позволяет параллельно обрабатывать огромный массив данных	При малом количестве данных часть ВЭ остается незадействованной
ОПМД	Оценка каждой программы производится последовательно в заданном ВЭ, что упрощает её пересчет	При малом количестве программ в поколении часть ВЭ остается незадействованной. При выполнении разных команд в ВБ может возникнуть коллизия команд.
МПМД	Параллельная оценка программ (на ВБ), параллельная подача данных (на ВЭ). При малом количестве данных, все ВБ задействованы.	Архитектурное ограничение на количество программ (ВБ) в поколении. При малом количестве программ часть ВБ остается незадействованной.

Для реализации на ГУ подходят две из трех вышеупомянутых стратегий: ОКМД для большого количества данных и малого количества программ в поколении; МПМД для любого количества данных и любого количества программ в поколении. При этом в указанных стратегиях исключается возможность расхождения интерпретации команд ВБ, поскольку во всех ВЭ (в пределах каждого ВБ) выполняются одинаковые команды.

Литература. 1. D. Augusto, H. Bernardino, H. Barbosa. Parallel Genetic Programming on Graphics Processing Units. Genetic Programming - New Approaches and Successful Applications, InTech, pages 95-114, Oct 2012 2. D. Kirk, W. Hwu. Programming Massively Parallel Processors: A Hands-On Approach, Applications of GPU Computing Series, Morgan Kaufmann Publishers, 2010. 3. S. Ryoo, C. Rodrigues, S. Baghsorkhi. Optimization Principles and Application Performance Evaluation of a Multithreaded GPU Using CUDA. Symp. on Parallel Programming, pages 73–82