# СТОХАСТИЧЕСКИЙ АНАЛИЗ ПАРАЛЛЕЛЬНОЙ ПРОИЗВОДИТЕЛЬНОСТИ ПРОГРАММНЫХ СИСТЕМ

**А.В. Бухановский, С.В. Ковальчук**

Важнейшей характеристикой программной системы (ПС), исполняемой на $p$ вычислителях (ядрах, процессорах, узлах), является ее производительность. Она определяется как общим временем работы программы, так и безразмерными характеристиками – параллельным ускорением и эффективностью. Классическое определение параллельного ускорения, как отношения времени выполнения программы на одном и нескольких вычислителях при одинаковых условиях [1] в большинстве случаев дает достаточно надежный способ для его измерения на основе экспериментальных данных. Однако существует ряд задач, в которых время работы программы в общем случае меняется от запуска к запуску в силу ряда факторов, обусловленных как особенностями алгоритма (наличием случайных ветвлений, точек останова и пр.), так и спецификой вычислительной системы (прежде всего, конкуренцией за ресурс с другими вычислительными процессами). В этих условиях экспериментально определить, и, тем более, спрогнозировать параллельную производительность ПС, пользуясь традиционными подходами, оказывается затруднительно. Это обусловлено тем, что значения времени работы последовательной и параллельной версий программы могут быть случайными величинами. Как следствие, каждое измерение времени работы программы соответствует в общем случае *разным* условиям эксперимента, что не позволяет для ее измерения напрямую воспользоваться классическими определениями ускорения и эффективности.

В докладе предлагается метод косвенных статистических измерений характеристик параллельной производительности, основанный на формализме детерминированной функции случайного аргумента. Предположим, что случайный характер времени работы программы определяется структурой самого алгоритма (например, характеристиками псевдослучайных величин, составляющих источник стохастизации) и (или) характерными особенностями вычислительной системы (распределением загрузки коммуникационной сети). Это дает основание представить время работы программы на $p$ вычислителях в форме детерминированной функции $T_p = T(p,R)$, зависящей от набора аргументов $R$, которые являются случайными величинами. Если известна плотность распределения $f_R(x)$, то плотность распределения времени выполнения программы $f_{T_p}(x)$, а также соответствующие плотности распределения параллельного ускорения $f_S(x)$ и эффективности $f_\varepsilon(x)$ выражаются через нее однозначно [2]. Таким образом, имея информацию о распределении источника стохастизации (алгоритмического или системного) и выражение, определяющее зависимость времени выполнения от количественного значения этого источника, можно оценить вероятностные характеристики параллельной производительности при произвольном количестве вычислителей. Область применения подхода ограничивается степенью адекватности модели $T_p = T(p,R)$.

Использование предложенного метода статистических измерений параллельной производительности в докладе будет проиллюстрировано на различных примерах программ со случайным временем выполнения, обусловленным как стохастическими эффектами алгоритма, так и особенностями вычислительной архитектуры.

***Стохастические алгоритмы со случайным выходом*** являются наиболее частым источником стохастизации времени работы приложения. Обычно они основаны на итерационном процессе, управляемом последовательностью псевдослучайных чисел, выход из которого выполняется по достижении определенного условия. Этот эффект порождает случайный характер времени выполнения, отмечаемый, в частности, в работах [3-5]. При этом относительный разброс времени выполнения не зависит от характеристик вычислительной системы и является специфической особенностью конкретной задачи.

В докладе стохастические алгоритмы со случайным выходом рассматриваются в качестве простейшего иллюстративного примера для измерения вероятностных характеристик параллельной производительности на системах с общей памятью. Случайная величина $R$ суть количество итераций алгоритма, а зависимость $T_p = T(p,R)$ для простоты представляется в форме закона Амдала. В результате плотность распределения параллельного ускорения задается выражением:

$$f(S) = f(R(S))\left|R'(S)\right|, \tag{1}$$

где функция $R(S)$ является обратной к закону Амдала и выражает связь между числом итераций и параллельным ускорением при каждом конкретном значении $S$. На рис. 1(а) приведены результаты расчета плотности распределения ускорения для алгоритма адаптивного случайного поиска с линейной тактикой (распараллеливание по целевой функции). Эксперименты, по которым определялась величина $f_R(x)$, проведены на рабочей станции с процессором Intel Core 2 Quad.

***Стохастические алгоритмы со случайным выходом и барьером*** также достаточно часто применяются для численного решения задач оптимизации, например, при глобальном поиске или при дискретной оптимизации. Они реализуют конкурирующую (или островную) схему: параллельно выполняющиеся задачи реализуют один и тот же алгоритм над различными данными. По завершении работы всех задач их результаты сравниваются с целью выбора оптимального.

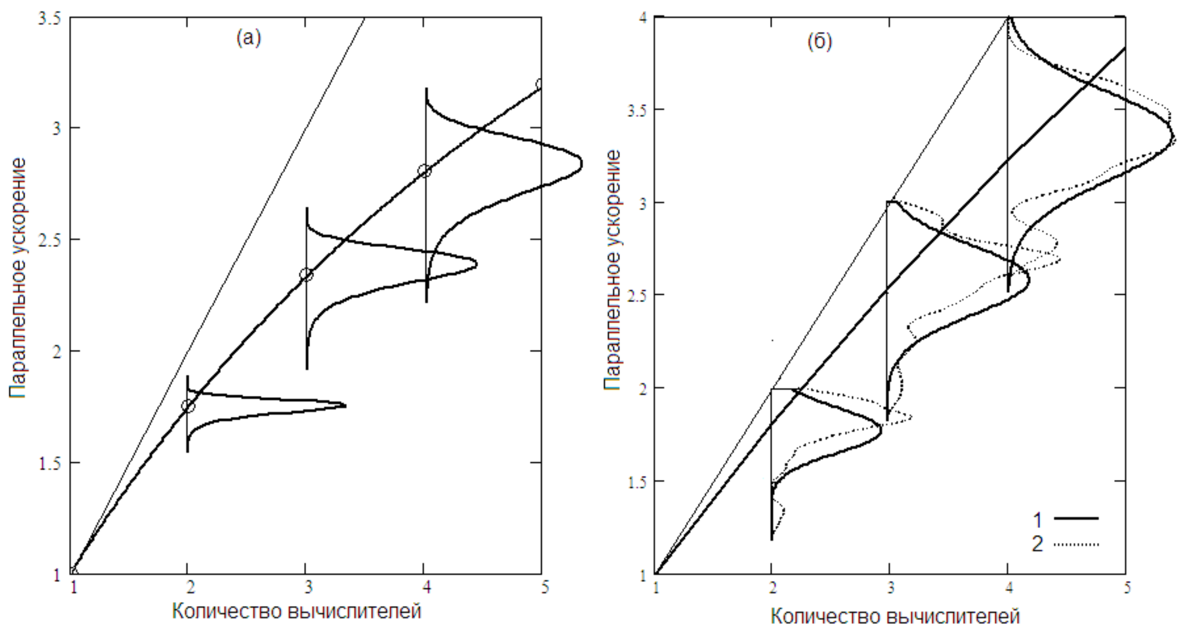


Рис. 1. Распределения параллельного ускорения в зависимости от количества вычислителей: (а) - алгоритм со случайным выходом без барьера; (б) – алгоритм со случайным выходом и барьером. 1 – теоретические значения; 2 – ядерная оценка по данным эксперимента

Особенностью применения предложенного в докладе метода к алгоритмам со случайным выходом и барьером состоит в том, что они управляются не одной, а сразу несколькими (по количеству вычислителей) последовательностями псевдослучайных чисел, в то время как итоговое время расчетов определяется только одной (соответствующей максимальной или минимальной реализации). В этом случае, имея информацию о законе распределения $f_R(x)$, можно оценить плотность распределения времени вычислений $f_{max}(x)$, опираясь на статистическую теорию экстремальных значений [6]. В частности, если закон распределения количества итераций близок к нормальному (что характерно для методов класса случайного поиска в силу центральной предельной теоремы), то распределение максимального (минимального) значения должно тяготеть к I предельному распределению (Фишера-Типпета, Гумбеля). В этом случае, используя формулу (1), выражение для плотности распределения параллельного ускорения представляется в форме модели комбинированного распределения:

$$f(S) = \int_0^\infty f_{\max}(R_{\max}(S,\vartheta))\left|R'_{\max}(S,\vartheta)\right|\varphi(\vartheta \mid m_\vartheta, \sigma_\vartheta)d\vartheta \qquad (2)$$

Здесь сумма $\vartheta = \sum_{k=1}^{p-1} R_k$ характеризует задачи, завершившиеся до момента достижения барьера (в случае задачи на максимум) или, наоборот, принудительно остановленные при достижении результата одним из вычислителей (в случае задачи на минимум). Функция $\varphi(\bullet)$ - плотность распределения величины $\vartheta$, аппроксимируемая усеченным нормальным распределением. Пример оценивания распределений параллельного ускорения по (2) для генетического алгоритма с островной моделью распараллеливания приведен на рис. 1(б). Результаты расчетов сопоставлены с ядерными оценками, полученными путем испытаний на экспериментальном стенде, когда и последовательная, и параллельная версия программы использовали одни и те же управляющие последовательности псевдослучайных чисел.

***Детерминированные алгоритмы в стохастической вычислительной среде*** представляют собой специфический объект исследований, актуальность которого связана с необходимостью планирования вычислений в гиперсетях (включающих пиринговые архитектуры, Грид и пр.). В данном случае стохастизация времени вычислений обусловлена, в первую очередь, «коммунальными» эффектами, связанными, например, с использованием для передачи команд и данных общих каналов Интернет и конкуренцией за ресурсы удаленных вычислительных систем с другими пользователями. В отличие от стохастических алгоритмов, описанных выше, в этих условиях даже для однородных «в среднем» вычислительных архитектур при идеальной статической балансировке нагрузки вычислителей будет возникать дисбаланс, обусловленный изменением характеристик вычислительной системы. В этом случае общее время выполнения $T_{total} = \max_{i=1,p}[T_i]$, где $T_i = T_{calc} + T_{comm}$. Величины $T_{calc}, T_{comm}$ соответствуют времени выполнения задачи на удаленном узле и времени передачи данных на узел, используя для этого общий канал. Учитывая, что $T_{total}$ по определению является крайним членом выборки из $p$ случайных величин, для формализации его распределения может быть также применена теория экстремальных значений и построена процедура косвенных измерений вероятностных характеристик ускорения в форме (2). На рис. 2 приведен пример определения характеристик параллельного ускорения на тестовом полигоне Грид на основе Intel Grid Programming Environment (Intel GPE) с разной степенью влияния коммуникационного фактора ($T_{calc} / T_{total}$) [7]. Видно, что при увеличении доли коммуникационных затрат возрастает и разброс параллельного ускорения.

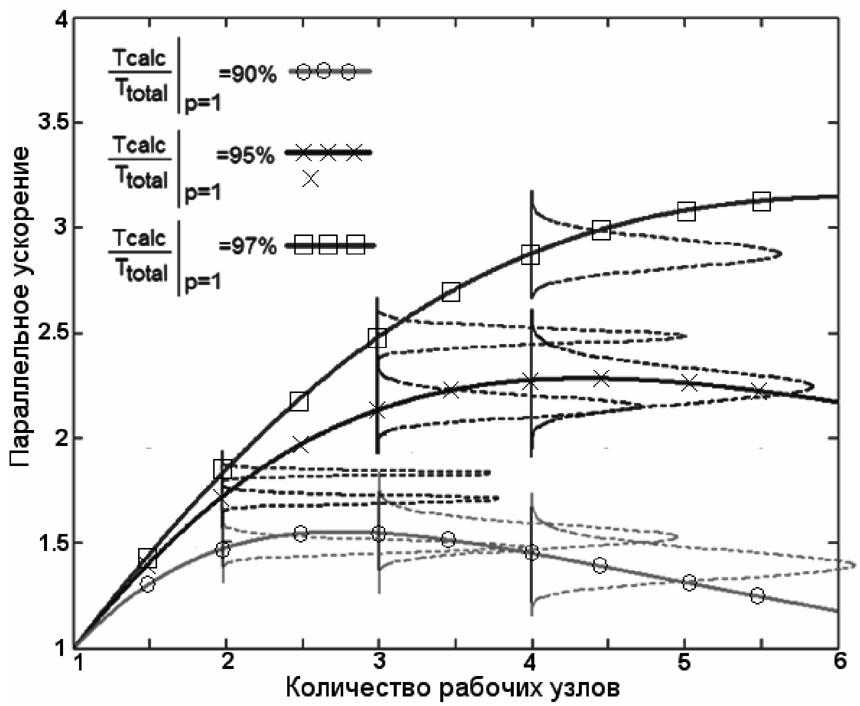


Рис. 2. Изменение вероятностных характеристик параллельного ускорения приложений в Грид в зависимости от количества узлов на основе косвенных измерений (2).

***Стохастические алгоритмы в стохастической вычислительной среде*** являются наиболее сложным объектом исследования в силу того, что изменчивость времени выполнения программы обусловлена сочетанием двух независимых групп случайных факторов, характеризующих как сам алгоритм, так и среду, в которой он выполняется. Изучение производительности в этом случае требует специфических методов дисперсионного анализа, которые будут анонсированы в докладе.

ЛИТЕРАТУРА:


1. *Ортега Дж.* Введение в параллельные и векторные методы решения линейных систем. М.: Мир, 1991.
2. *Лившиц Н. А., Пугачев В. Н.* Вероятностный анализ систем автоматического управления. Т.1. М., Советское радио, 1963
3. *Uresin A., Dubois M.* Effect of asynchronism in the convergence rate of iterative algorithms // J. Parallel and Distributed Computing, 34. 1996. P. 66—81.

4. *Casanova H.* Stochastic models for performance analysis of iterative algorithms in distributed environment // Ph.D. thesis Univ. Tennessee, Knoxville, 1998.
5. *Ларченко А. В., Дунаев А. В., Бухановский А. В.* Анализ и моделирование производительности параллельных стохастических алгоритмов, адаптированных к особенностям многоядерных вычислительных архитектур // Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ: Труды Всероссийской научной конференции (24-29 сентября 2007 г., г. Новороссийск). М.: Издательство МГУ, 2007. с. 156.
6. *Лидбеттер М., Линдгрен Г., Ротсен Х.* Экстремумы случайных последовательностей и рядов. М., Мир, 1989.
7. *Дунаев А.В., Ларченко А.В.,. Бухановский А.В* Моделирование параллельных вычислительных процессов в среде Грид на примере Intel Grid Programming Environment // Параллельные вычислительные технологии (ПаВТ'2008): Труды международной научной конференции (Санкт-Петербург, 28 января – 1 февраля 2008 г.). – Челябинск: Изд. ЮУрГУ, 2008. – с. 283-289.