О ВЕРХНИХ И НИЖНИХ ГРАНИЦАХ ОЦЕНОК ПРОИЗВОДИТЕЛЬНОСТИ МНОГОМАШИННЫХ МНОГОПРОЦЕССОРНЫХ КОМПЛЕКСОВ НА НАУЧНЫХ ПРИЛОЖЕНИЯХ

Г.С. Речистов

Введение

Данная работа расширяет методику оценки производительности многоядерных приложений по методу «линии крыши» (*англ*. Roofline), описанную в работе [1], расширяя её применимость на класс распределённых кластерных систем и вводя несколько возможных узких мест производительности, тогда как в оригинальной модели присутствует только одно. Дополнительно описывается оценка снизу для скорости работы приложения, дополняющая метод «линии крыши». С помощью одновременного использования обеих моделей представляется возможным получить интервал, в котором будет лежать экспериментальное значение искомой величины производительности параллельного приложения.

Оценка сверху

Эта методика исходит из наблюдения, что вычислительная производительность систем определяется «узким местом» — одной из подсистем передачи данных с наибольшей утилизацией при исполнении конкретной задачи. Для различных задач и конфигураций аппаратуры лимитирующими факторами могут быть различные подсистемы. Нас будет интересовать пиковая производительность приложений, проводящих вычисления с числами с плавающей запятой, измеряемая во FLOPS (*англ.* floating point operations per second, операции над числами с плавающей запятой). Всюду в этой статье будут подразумеваться операции с числами шириной в 64 бита, т.н. двойной точности.

Выделим три возможные точки (далее называемые каналами) образования «узкого места» — кэш-память, оперативную память (ОЗУ) и сеть передачи сообщений между узлами. Для каждой из них введём характеристику c – удельную интенсивность операций, равную среднему количеству операций над числами с плавающей запятой, отнесённому к одному переданному байту. Эта величина будет различна для всех трёх каналов; например, в ОЗУ попадают только те запросы данных, которые не были удовлетворены системой кэшей. Удельная интенсивность зависит как от алгоритма приложения, так и от архитектурных особенностей используемой системы и свойств канала передачи данных.

Вторая характеристика — это пропускная способность X каждой подсистемы, измеряемая в байт/с.

Последняя величина — это пиковая производительность вычислительного ядра T в случае, если ни одна из подсистем не ограничивает производительность; в таком случае все вычислительные узлы ядра полностью утилизированы каждый такт, не простаивая в ожидании поступления новых данных по каналам.

Если приложение в своей работе полностью использует некоторый канал передачи данных со скоростью X байт/с, при этом имея удельную интенсивность операций на каждый переданный байт, равную c, то показываемая производительность будет равна Xc операций/с. С другой стороны, эта величина не может превышать заложенное микроархитектурой процессора (шириной векторных команд, частотой процессора и т.п. факторами) значения T. В результате производительность приложения при условии, что узким местом является подсистема i, равна минимуму из этих двух значений:

$$FLOPS_i = \min(Xc, T)$$

Если в системе существует несколько потенциальных узких мест, то необходимо выбрать минимальное значение, даваемое моделью для каждого из них

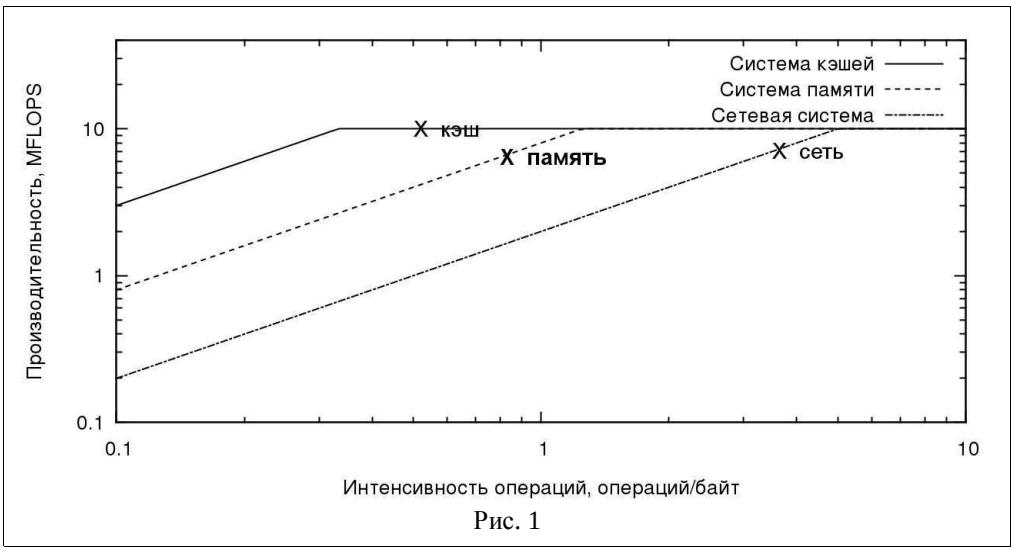

Рис. 1

$$FLOPS = \min(FLOPS_1, FLOPS_2, FLOPS_3)$$

Заметим, что это рассуждение верно для отдельного потока приложения. Для получения полной производительности мы должны умножить величину на количество параллельных процессов N:

$$FLOPS_{async}(N) = N \min(FLOPS_1, FLOPS_2, FLOPS_3)$$

Обозначение $FLOPS_{async}$ станет понятным при рассмотрении второй оценки, делающей другие предположения о характере поведения приложения.

Оценка снизу

Приведённые выше рассуждения для нахождения скорости приложения подразумевают, что оно в состоянии эффективно использовать каналы передачи данных, полностью занимая их полосу пропускания. Для этого его алгоритм должен обеспечивать, чтобы длительность отдельных актов коммуникаций не пересекались по времени (т.е. все они *асинхронны*). В противном случае, когда два или более сообщений одновременно поступают в канал передачи, каждый из них отодвигает во времени обработку остальных, тем самым уменьшая степень утилизации канала передачи (эффективно значение X).

Для получения пессимистичной оценки (снизу) производительности параллельного приложения построим другую аналитическая модель, в которой передача сообщений от всех потоков приложения всегда инициируется одновременно (синхронно), таким образом организуя «затор» в каналах передачи. Время рассасывания этого затора пропорционально количеству процессов, конкурирующих с одним за передачу данных, т.е. (N-1), и обратно пропорционально пропускной способности канала X (чем он производительнее, тем быстрее закончится передача данных). Тогда длительность интервала коммуникаций равна

$$t_{sync} = (N\text{-}1)N_{bytes}/X,$$

где N_{bytes} — среднее количество передаваемых байт одним процессом.

Длительность интервала «чистых» вычислений, на котором не происходит коммуникаций, равна

$$t_{comp} = N_{ops}/T,$$

где N_{ops} — среднее количество операций с плавающей запятой в одном процессе за этот период.

Учитывая, что по определению удельная производительность с = N_{ops}/N_{bytes} и что полный цикл работы приложения равен $t_{sync}+ t_{comp}$, а также, что полное число процессов N, мы можем вывести формулу полной производительности параллельного приложения для случая синхронной передачи данных.

$$FLOPS_{sync}(N) = \frac{TNXc}{T(N-1)+Xc}$$

Анализ соотношения двух оценок и их асимптотического поведения

Заметим, что в оба полученных выражения: $FLOPS_{async}$ и $FLOPS_{sync}$ — величины удельной интенсивности операций и пропускной способности канала входят только как произведение Xc. Таким образом, мы имеем три независимые переменные: T — пиковая производительность одного процесса, обусловленная архитектурными особенностями ядра приложения, N — количество параллельных процессов, Xc — максимальная производительность, обеспечиваемая каналом связи. Проведём краткий анализ поведения полученных выражений для различных диапазонов значений величин.

При N = 1 формула для пессимистичной (синхронной) оценки даёт значение T, тогда как оценка сверху не даёт дополнительной информации. При росте N и фиксированных значениях T и Xc нижняя оценка

ограничена сверху значением $Xc/(N-1)$. При неизменных N, T и растущим Xc обе величины изменяются от нуля при $Xc = 0$ (соотв. отсутствию операций с плавающей запятой) до TN при больших значениях Xc (идеальный случай параллелизма при сверхбыстром канале передачи и/или суперэффективном алгоритме). Можно видеть, что оценка для синхронного режима всегда остаётся меньше величины, получаемой для асинхронного приложения при любом характере поведения входящих в формулы величин — T, N, Xc.

Экспериментальное получение данных

Полученные в данной статье формулы получили первоначальную проверку на работе приложения **mdrun (**компилятор GCC 4.4.5, опции сборки: двойная точность, поддержка библиотеки MPI**)** из состава пакета моделирования молекулярной динамики Gromacs [2], запущенного на многомашинном вычислительном кластере.

Величина T («потолок» производительности) для одного ядра процессора находится из результатов запуска программы High Performance Linpack [3] при наибольших размерах матрицы. Значения пропускных способностей X берутся из программ-тестов: для ОЗУ — STREAM [4], для кэшей — LMBench [5], для сетевой подсистемы — Netperfmeter [6]. Значения c находятся из значений аппаратных счётчиков производительности, считываемых с помощью приложения Intel VTune Amplifier XE 2011 [7].

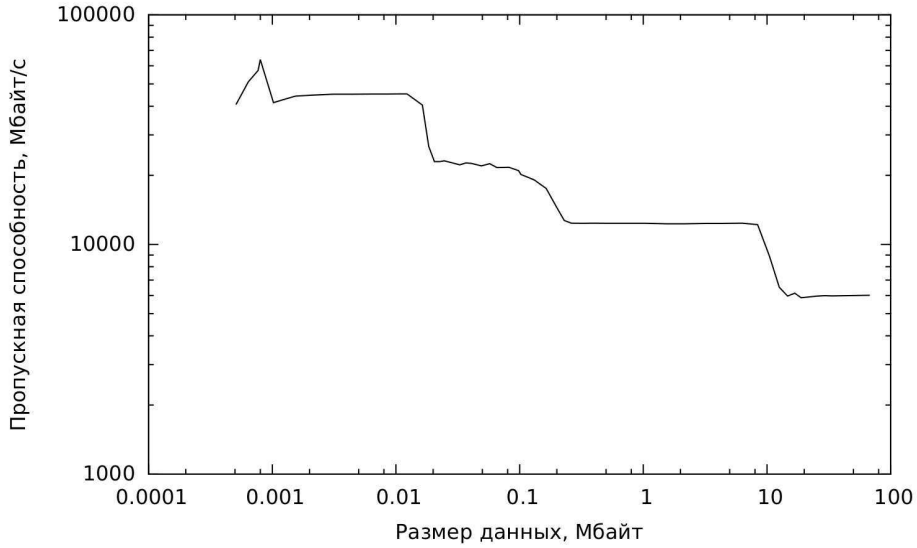

Рис. 2

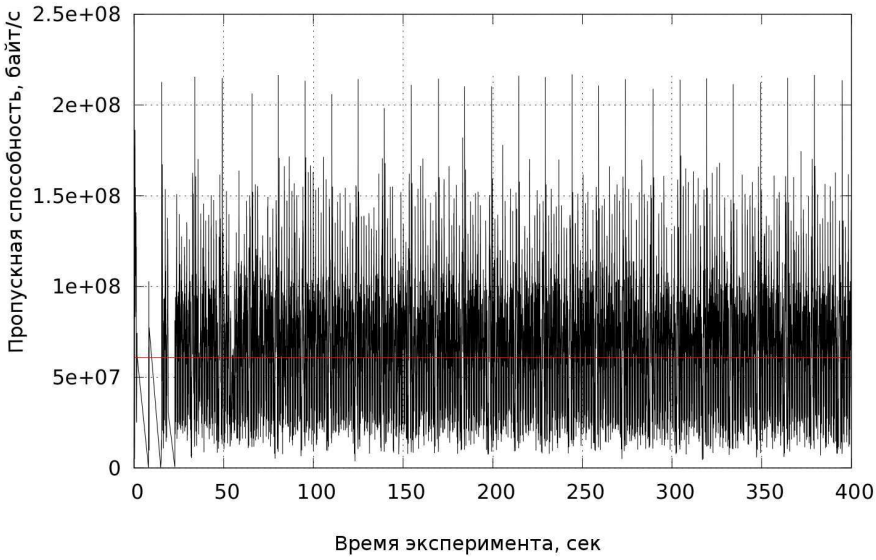

Рис. 3

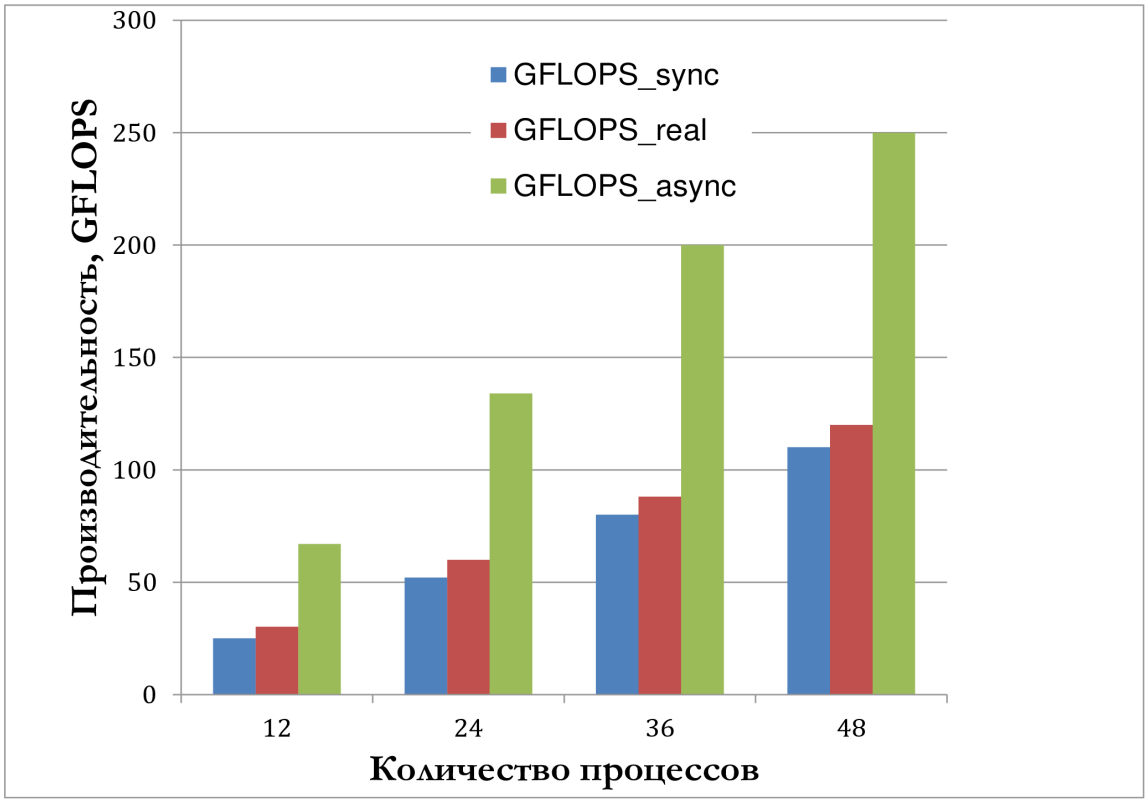

Рис. 4

На рис. 2 приведены измерения пропускной способности трёх уровней кэшей используемой системы. Каждая горизонтальная часть графика соответствует работе одного из уровней кэша системы. На рис. 3 — измеренная производительность локальной сети для 400 секунд работы приложения со средним значением. На рис. 4 сравниваются значения производительности, полученные для **mdrun** при запусках на системе, содержащей от 1 до 4 двухпроцессорных узлов (каждый из них имел по 6 ядер Intel Xeon X5680 2.8 ГГц), соединённых сетью 1 Gbit Ethernet, а также аналитические оценки, вычисленные для этих запусков. Как видно из результатов, несмотря на то, что реальное значение всегда укладывалось в предсказанные границы, сами они довольно далеко отстоят друг от друга.

Отметим, что в проведённых экспериментах не было обнаружено явления смены положения узкого места (например, от памяти к кэшам) — приложение крайне хорошо распараллеливается в исследованных пределах изменения числа процессов.

Заключение

Направления дальнейшей работы включают в себя проверку разработанных формул на широком списке параллельных приложений, включающем в себя задачи различных областей науки, разнящиеся по степени присущей им масштабируемости, используемым алгоритмам и парадигмам синхронизации совместной работы. Для возможности наблюдения эффектов смены узкого места необходимо исследовать более широкий диапазон изменения количества параллельных потоков, на которых запускается исследуемая задача. Следует ещё раз подчеркнуть, что при некоторых соотношениях входных параметров разница между численными значениями для верхней и нижней границ производительности, даваемых формулами, чрезмерно велико: пессимистичная оценка может быть меньше оптимистичной в N раз, тогда как с практической т.з. важным является минимизация интервала неопределённости при оценке скорости приложений. Требуется разработать дальнейшее улучшение формул, учитывающее тот факт, что реальное приложение скорее всего имеет частичное перекрытие периодов передачи сообщений и не сводится к какому-либо из двух рассматриваемых крайних случаев. Аналитическое описание этого факта и учёт его в формулах позволили бы значительно сблизить верхнюю и нижнюю границы аналитически получаемых значений производительности, тем самым повысив их ценность.

Работа выполнена в рамках гранта, выделенного в соответствии с постановлением Правительства России №220 от 9.04.2010 г.

ЛИТЕРАТУРА:

1. *Williams S. W., Waterman A., Patterson D. A.* Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures // Tech. rep. UCB/EECS-2008-134. EECS Department, University of

California, Berkeley. 2008.<http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-134.html> (дата обращения 16.04.2012)
2. *Van Der Spoel D. et al*. GROMACS: Fast, Flexible, and Free // Journal of Computational Chemistry. 2005. Т. 26. № 16. Pp. 1701–1718.
3. *Dongarra J. J*. Performance of Various Computers Using Standard Linear Equations Software [Электронный ресурс] // <ftp://netlib2.cs.utk.edu/benchmark/performance.pdf> (дата обращения 18.04.2012)
4. *McCalpin J. D.* STREAM: Sustainable Memory Bandwidth in High Performance Computers [Электронный ресурс] <http://www.cs.virginia.edu/stream> (дата обращения 18.04.2012)
5. *McVoy L., Staelin C.* Lmbench: portable tools for performance analysis // Proceedings of the 1996 annual conference on USENIX Annual Technical Conference. San Diego, CA: USENIX Association, 1996. С.23–28.
6. *Dreibholz T.* Netperfmeter: A TCP/UDP/SCTP/DCCP Network Performance Meter Tool [Электронный ресурс] <http://www.iem.uni-due.de/~dreibh/netperfmeter> (дата обращения 18.04.2012)
7. Intel® VTune™ Amplifier XE / *Intel Corporation* [Электронный ресурс] <http://software.intel.com/en-us/articles/intel-vtune-amplifier-xe/> (дата обращения 02.07.2012)