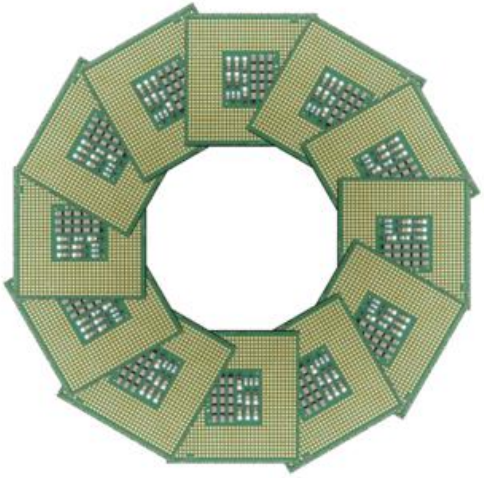
Оценка высокопроизводительных систем

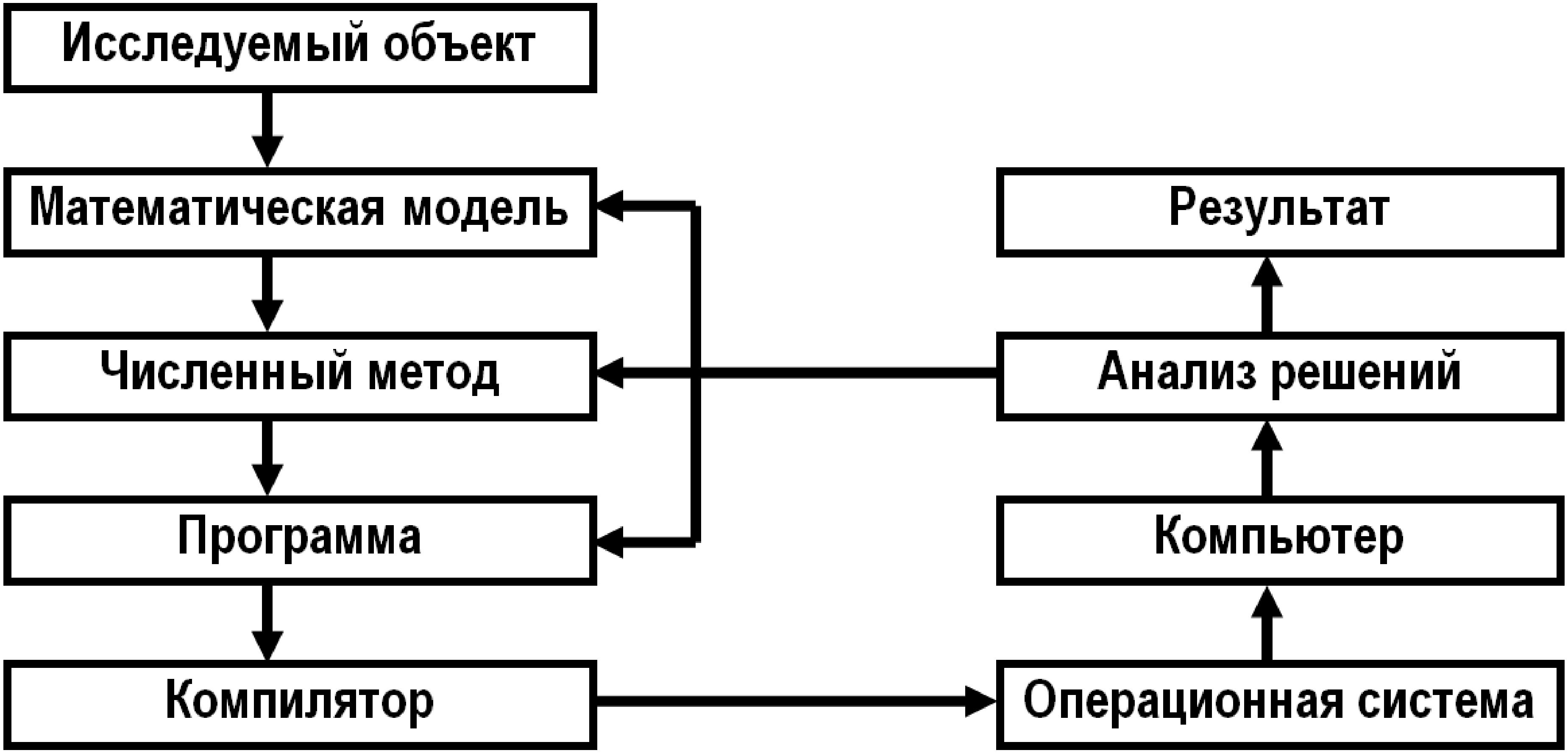


План изложения:

- Численный эксперимент и параллельная форма алгоритма
- Схемы параллельного выполнения алгоритма
- Показатели эффективности параллельного алгоритма
- Оценка достижимого параллелизма. Закон Амдала
- Тест Linpack

- Естественным способом решения проблем параллельного программирования стало создание стандартов как для вычислительной техники, так и для программного обеспечения.
- Требуется выделить класс задач, которые необходимо решать.
- Оптимально, следующим шагом должен быть выбор или конструирование системы для выделенного класса задач.
- Необходимо создать матобеспечение подходящее для конкретного класса задач на выбранной системе.
- Написать программу для данной конкретной задачи с учетом перечисленных факторов.

Этапы численного эксперимента



Схемы параллельного выполнения алгоритма

- Для описания информационных зависимостей в алгоритмах традиционно используется модель в виде графа «операции-операнды».
- Будем предполагать, что время выполнения любых вычислительных операций является одинаковым и равняется 1 (в тех или иных единицах измерения).
- Передача данных между вычислительными устройствами выполняется мгновенно без каких-либо затрат времени.
- Тогда представим множество операций, выполняемых в исследуемом алгоритме решения вычислительной задачи, и существующие между операциями информационные зависимости в виде ациклического ориентированного графа $G=(V, R)$. Здесь $V=\{1,\dots,|V|\}$ – множество вершин графа, представляющих выполняемые операции алгоритма, а R – множество дуг графа, при этом дуга $r=(i, j)$ принадлежит графу только, если операция j использует результат выполнения операции i.

Расписание

Для параллельного выполнения вычислений необходимо задать множество – расписание:

$$H_p = \{(i, P_i, t_i) : i \in V\}$$

для каждой операции указывается номер используемого для процессора P_i и время начала выполнения операции t_i.

Вычислительная схема алгоритма G совместно с расписанием H_p может рассматриваться как модель параллельного алгоритма $A_p(G, H_p)$, исполняемого с использованием процессоров.

Время выполнения алгоритма

Время выполнения параллельного алгоритма определяется максимальным значением времени, используемым в расписании:

$$T_p(G, H_p) = \max_{i \in V} \left(t_i + 1 \right)$$

Для выбранной схемы вычислений желательно использование расписания, обеспечивающего минимальное время исполнения алгоритма:

$$T_p(G) = \min_{H_p} T_p(G, H_p)$$

Показатели времени выполнения параллельного алгоритма

- Уменьшение времени выполнения может быть обеспечено и путем подбора наилучшей вычислительной схемы:

$$T_p = \min_G T_p(G)$$

- Оценки $T_p(G, H_p)$, $T_p(G)$ и T_p могут быть использованы в качестве показателей времени выполнения параллельного алгоритма.

Показатели эффективности параллельного алгоритма

Ускорение (speedup), получаемое при использовании параллельного алгоритма для p процессоров, по сравнению с последовательным вариантом выполнения вычислений определяется величиной:

$$S_p(n) = T_1(n) / T_p(n)$$

т.е. как отношение времени решения задач на скалярной ЭВМ к времени выполнения параллельного алгоритма (величина *n* используется для параметризации вычислительной сложности решаемой задачи и, например, - количество входных данных задачи).

Эффективность

Эффективность (efficiency) использования параллельным алгоритмом процессоров при решении задачи определяется соотношением

$$E_p(n) = T_1(n) / \left(pT_p(n) \right) = S_p(n) / p$$

Величина эффективности определяет среднюю долю времени выполнения алгоритма, в течение которой процессоры реально используются для решения задачи.

Оценка достижимого параллелизма

Оценка качества параллельных вычислений предполагает знание наилучших (максимально достижимых) значений показателей ускорения и эффективности, однако, получение идеальных величин $S_p = p$ для ускорения и $E_p = 1$ для эффективности может быть обеспечено не для всех вычислительно трудоемких задач.

Закон Амдала

Достижению максимального ускорения может препятствовать существование в выполняемых вычислениях последовательных расчетов, которые не могут быть распараллелены. Пусть f есть доля последовательных вычислений в применяемом алгоритме обработки данных, тогда в соответствии с законом Амдала (Amdahl) ускорение процесса вычислений при использовании p процессоров ограничивается величиной:

$$S_p \le \frac{1}{f + (1-f)/p} \le S^* = \frac{1}{f}$$

Закон Амдала (2)

Так, например, при наличии всего 10% последовательных команд в выполняемых вычислениях, эффект использования параллелизма не может превышать 10-кратного ускорения обработки данных.

Вывод - не любая программа может быть эффективно распараллелена.

Необходимо, чтобы доля информационно независимых операций была очень большой.

Как показывает практика, большинство вычислительных алгоритмов устроено в этом смысле достаточно хорошим образом.

Тест Linpack

Для сравнения высокопроизводительных систем используется тест Linpack. Он рассматривается как некоторое достаточно универсальное средство для измерения производительности компьютера в операциях с плавающей точкой. Его авторство принадлежит Джеку Донгарра, Пётру Лужчеку и Антуану Перитэ. Тест представляет из себя компьютерную программу, которая решает плотную систему линейных алгебраических уравнений. Решение основано на LU-разложении, матрица генерируется с помощью псевдо-случайного генератора.

Тест Linpack (2)

- Для того чтобы измерить производительность кластерной системы на тесте High Performance Linpack (HPL), потребуется сам тест и библиотека BLAS. Собственно тест HPL доступен по адресу http://www.netlib.org/benchmark/hpl/hpl.tgz. При этом для высоких значений производительности важно подобрать размер задачи так, чтобы использовалась вся оперативная память узлов.

Первый тест в Linpack

Измеряет производительность двух процедур: DGEFA(SGEFA) и DGESL(SGESL) для 64- и 32-битных версий, соответственно.

DGEFA выполняет LU-разложение с выбором ведущего элемента по столбцу, DGESL использует данное разложение для решения системы линейных алгебраических уравнений.

Большая часть времени требуется на разложение матрицы – DGEFA – *O(N3)*.

Как только оно завершено, находится решение – DGESL – *O(N2)*.

Второй и третий тест Linpack

Второй тест заботает с матрицей размерности 1000.

Третий тест создан для проверки производительности «хорошо» распараллеленных вычислений, и именно по его результату строится рейтинг Top500.

Этот тест пытается вычислить максимально-достижимую производительность системы при решении системы линейных алгебраических уравнений и представляет из себя целый набор модулей, для которого пользователем предоставляется реализация недостающих интерфейсов.

Размерность задачи и метод решения можно варьировать, чтобы добиться максимальной производительности на данном тесте.

Ограничения на решение те же, что и в предыдущем тесте. Так же запрещено пользоваться методом Штрассена умножения матриц, который хоть и даёт лучшую временную асимптотическую оценку, однако обладает в десятки раз большей численной погрешностью, которая ещё и растёт с ростом размерности примерно в 1,5-2 раза быстрее роста аналогичной погрешности привычного алгоритма умножения матриц.

Пример результата теста

Ниже представлен в виде результат запуска теста Linpack для суперкомпьютера СКИФ, установленного в Московском государственном университете:

- ==
- HPLinpack 1.0a -- High-Performance Linpack benchmark -- January 20, 2004
- Written by A. Petitet and R. Clint Whaley, Innovative Computing Labs., UTK
- ==
- An explanation of the input/output parameters follows:
- T/V : Wall time / encoded variant.
- N : The order of the coefficient matrix A.
- NB : The partitioning blocking factor.
- P : The number of process rows.
- Q : The number of process columns.
- Time : Time in seconds to solve the linear system.
- Gflops : Rate of execution for solving the linear system.

- The following parameter values will be used:
- N : 740000
- NB : 168
- PMAP : Row-major process mapping
- P : 25
- Q : 200
- PFACT : Right
- NBMIN : 4
- NDIV : 3
- RFACT : Right
- BCAST : 1ringM
- DEPTH : 0
- SWAP : Mix (threshold = 256)
- L1 : transposed form
- U : transposed form
- EQUIL : yes
- ALIGN : 8 double precision words
- --
- - The matrix A is randomly generated for each test.
- - The following scaled residual checks will be computed:
- 1) ||Ax-b||_oo / (eps * ||A||_1 * N)
- 2) ||Ax-b||_oo / (eps * ||A||_1 * ||x||_1)
- 3) ||Ax-b||_oo / (eps * ||A||_oo * ||x||_oo)
- - The relative machine precision (eps) is taken to be 2.220446e-16
- - Computational tests pass if scaled residuals are less than 16.0

Пример результата теста (окончание)

- ==
- T/V N NB P Q Time Gflops
- --
- WR01R3R4 740000 168 25 200 5709.59 4.732e+04
- --
- ||Ax-b||_oo / (eps * ||A||_1 * N) = 0.0003600 ...... PASSED
- ||Ax-b||_oo / (eps * ||A||_1 * ||x||_1) = 0.0005678 ...... PASSED
- ||Ax-b||_oo / (eps * ||A||_oo * ||x||_oo) = 0.0000950 ...... PASSED
- ==
- Finished 1 tests with the following results:
- 1 tests completed and passed residual checks,
- 0 tests completed and failed residual checks,
- 0 tests skipped because of illegal input values.
- --
- End of Tests.

Использование Linpack

Тест используется для формирования списка Top500 – пятисот самых мощных вычислительных систем мира.

Указывается Rpeak – теоретически достижимая (пиковая) производительность и Rmax – равная производительности системы на тесте Linpack.

К этим показателям и нужно относиться соответственно – они говорят о том, насколько хорошо система может решать системы линейных алгебраических уравнений с плотной матрицей указанным методом.

Для других же задач могут быть совсем другие результаты