

Бенчмарк LINPACK для архитектур Intel: существующие варианты, сравнительный анализ возможностей и демонстрируемой производительности

Григорий Речистов*

17 октября 2011 г.

Аннотация

The paper describes three variants of Linpack benchmark available for Intel IA-32 architecture. The results of the runs and comparative analysis of performance is presented.

Key words: Linpack, performance, x86, smp, mpi

В статье описаны три варианта бенчмарка Linpack, доступные для архитектуры Intel. Представлены результаты их запусков и проводится сравнительный анализ продемонстрированной производительности.

Ключевые слова: Linpack, производительность, x86, smp, mpi

Содержание

1	Введение	3
2	Работы по теме	4
3	Библиотеки и утилиты, необходимые для сборки HPL	4
4	Три варианта теста	4
4.1	Netlib HPL	4
4.2	Intel MP Linpack	5
4.3	Intel SMP Linpack	5

*Московский физико-технический институт.

5	Конфигурация исследуемой системы	6
5.1	Сборка тестов	6
5.2	Конфигурации для запусков тестов	6
6	Результаты измерений и их анализ	7
7	Заключения	9
	Список литературы	10

Список иллюстраций

1	Результаты измерений для первой машины.	8
2	Асимптотическая производительность бенчмарков для второй машины.	9

Список таблиц

1	Параметры из HPL.dat.	7
2	Полученные значения P_{max} , N_{max} , $N_{1/2}$, P_{reak} для второй машины.	8

1 Введение

Конструкторам и пользователям супер ЭВМ необходимо иметь возможность измерять и сравнивать производительность вычислительных машин. Для этих целей используются т.н. программы-бенчмарки, как правило представляющие из себя адаптированный вариант алгоритма решения какой-либо практической задачи; полезным результатом их работы являются числа, характеризующие производительность ЭВМ, использовавшейся для запуска бенчмарка. Существует огромное множество таких программ, каждая из которых предназначена для оценки какого-то определённого аспекта работы ЭВМ.

Начиная с 1979 года одним из таких бенчмарков является LINPACK [11]. Под этим общим названием объединены несколько программ, предназначенных для решения случайной системы линейных уравнений методом Гаусса на многопроцессорных системах. Различаются они размером задачи (матрицы) и допустимыми оптимизациями кода. Отдельная группа тестов LINPACK, т.н. High Performance Linpack (сокращённо HPL) представляет для нас наибольший интерес, так как размер задачи и типы оптимизаций для них неограничены; кроме того, HPL используется для составления списка TOP500 [16] суперкомпьютеров.

Правилами проведения измерений разрешается модифицировать или даже полностью переписывать код HPL для оптимизации его под конкретную архитектуру при условии, что вычисленные решения проходят проверку на корректность¹. Поэтому неудивителен факт, что каждый производитель микропроцессоров стремится выпустить свою версию теста, способную «выжать» всю производительность из их аппаратного обеспечения.

Данная работа описывает три публично доступных варианта HPL для процессоров фирмы Intel: Netlib HPL [7], Intel MP Linpack [8] и Intel SMP Linpack [9], их общие черты и различия, границы применимости. Затем описываются результаты запуска всех трёх бенчмарков на двух многопроцессорных компьютерах и сравниваются полученные результаты прогонов.

¹Отметим, что для так называемого бенчмарка LINPACK 100, предшествовавшего HPL, модифицировать код и запускать его на распределённых системах не разрешалось, поэтому все оптимизации сводились к возможностям использованного компилятора генерировать оптимальный код.

2 Работы по теме

Описание истории появления, измеряемых характеристик и деталей работы бенчмарка Linpack можно найти в [5].

Регулярно обновляемый отчёт о значениях производительности различных вычислительных комплексов, измеренных с помощью LINPACK, содержится в [4].

В работах [15] и [1] описываются решения, применённые для оптимизации производительности суперкомпьютера Intel TFLOPS для распределённого исполнения Linpack.

3 Библиотеки и утилиты, необходимые для сборки HPL

Для построения исполнимых файла бенчмарков из исходного кода используются следующие библиотеки.

- Библиотека линейной алгебры — реализация математических примитивов, входящих в алгоритм решения задачи. Известные альтернативные реализации: BLAS¹ [3], ATLAS² [2], Intel MKL³ [9].
- Библиотека MPI — реализация стандарта для параллельных вычислений путём передачи сообщений (message passing interface) [12]. Доступные для архитектуры Intel реализации: MPICH [13], Open MPI [14], Intel MPI [10].
- Библиотека OpenMP — реализация стандарта параллельных вычислений с помощью структурированной многопоточности на системе с общей памятью. Доступные для архитектуры Intel реализации: GNU OpenMP [6], OpenMP в составе Intel compiler [9].
- Компилятор — для сборки HPL из исходных кодов необходим компилятор языка Си с поддержкой стандарта C89 и опционально Fortran 77. Доступные альтернативы: GNU Compiler, Intel Compiler.

Отметим, что ввиду высокой переносимости исходных кодов HPL некоторые из описанных ниже вариантов Linpack можно собрать с помощью более чем одного набора библиотек и/или компиляторов, при этом имеется возможность выбирать наиболее оптимальное их сочетание.

4 Три варианта теста

4.1 Netlib HPL

Netlib HPL ([7], последняя версия 2.0 от 2008 года) является исторически первым тестом HPL. Его отличительные особенности:

¹Basic Linear Algebra System.

²Automatically Tuned Linear Algebra System.

³Math Kernel Library.

- Полностью открытый исходный код, написанный таким образом, что позволяет портировать тест на множество архитектур ЭВМ при условии существования для них компиляторов C/Fortran и библиотеки MPI.
- Параллельное исполнение обеспечивается использованием библиотеки MPI.
- Параллелизация методами OpenMP не используется в самом тесте, но может быть обеспечена в функциях линейной алгебры, если они были собраны с необходимыми опциями.

Для определения размера задач, числа запусков и опций решения уравнений используется конфигурационный текстовый файл `HPL.dat`. Подробное описание его формата приведено здесь: <http://www.netlib.org/benchmark/hpl/tuning.html>.

4.2 Intel MP Linpack

Intel MP Linpack [8] поставляется компанией Intel в составе своих пакетов библиотек и компиляторов. Его отличительные особенности:

- Исходный код основан на коде Netlib и доступен для скачивания и изучения. Тем не менее, он открыт не полностью — в двоичном виде поставляется библиотека `libhpl_hybrid`, доступная только для 32- и 64-битных ЦПУ Intel.
- Отличие исходных кодов этого бенчмарка от варианта Netlib — использование прагм OpenMP для задействования параллелизма исполнения для многоядерных систем с общей памятью.
- Поставляемые правила сборки в `Makefile` подразумевают использование инструментария от Intel: компилятора, библиотек алгебры и MPI, однако оставляется возможность использовать утилиты GNU; После ряда нетривиальных модификаций сборочных скриптов удалось собрать этот вариант с помощью как чисто GNU-библиотек, так и с помощью «смешанного» набора — компилятором GCC с библиотекой Intel MKL. В этой работе мы не будем рассматривать производительность таких вариантов.
- Параллельное исполнение обеспечивается использованием как библиотеки MPI, так и OpenMP. Таким образом, данный тест позиционируется как «гибридный», т.е. использующий как общую память и потоки, так и систему передачи сообщений для параллелизации исполнения алгоритма.
- Файл конфигурации имеет формат, идентичный используемому Netlib `HPL`.

4.3 Intel SMP Linpack

Intel SMP Linpack поставляется компанией Intel вместе с библиотекой MKL. Его отличительные особенности:

- Поставляется только в предкомпилированном виде для 32- и 64-битных вариантов ЦПУ Intel. Скорее всего, не базируется на коде Netlib.

- Для параллелизации исполнения использует только многопоточность внутри одного приложения одной ЭВМ, т.е. в отличие от вышеописанных программ его невозможно распределить на несколько компьютеров.
- При инициализации тест самостоятельно определяет количество доступных ядер и частоту работы процессора.
- Имеет собственный формат конфигурационного файла. При отсутствии файла запрашивает данные с терминала у пользователя.

5 Конфигурация исследуемой системы

Для сравнения характеристик производительности, демонстрируемых тремя бенчмарками, необходимо было установить конфигурацию каждого из них таким образом, чтобы максимально выровнять стартовые условия для их запуска. Для этого были предприняты следующие шаги:

- Все тесты выполнялись на одном хосте (т.к. SMP Linpack не может работать на распределённых системах)
- Число потоков/процессов для исполнения выбиралось равное числу физически доступных ядер ЦПУ (т.е. было равно восьми для первой машины и четырём для второй, см. секцию 6).
- Конфигурационные файлы HPL.dat были идентичны для MP Linpack и Netlib HPL.
- Размер задач для SMP Linpack были использованы те же, что и для остальных двух тестов.

5.1 Сборка тестов

Тесты MP и SMP Linpack были взяты в уже скомпилированном виде из пакета Intel MKL 10.3. Тест Netlib HPL был скомпилирован из исходных кодов со следующими параметрами:

- Компилятор — GCC 4.3.1.
- Библиотека ATLAS — версия 3.9.44, скомпилированная с автоматической настройкой производительности, с поддержкой 8 потоков, без поддержки Fortran 77.
- Библиотека MPI — MPICH версии 1.4, собранный с настройками по умолчанию.
- Флаги для компиляции взяты по умолчанию:
-O3 -funroll-loops -fomit-frame-pointer

5.2 Конфигурации для запусков тестов

Были использованы следующие размеры задач: 1000, 2000, 4000, 8000, 16000, 32000, 64000.

Для Netlib HPL и MP Linpack использовалась сетка $P \times Q$ 1×1 , остальные параметры были взяты из поставляемого с MP Linpack конфигурационного файла и оставлены без изменений. Они приведены в таблице 1.

NBs	168
PMP	0
threshold	16.0
PFACT	2
NBMIN	4
NDIV	2
RFACT	1
BCAST	1
DEPTH	0
SWAP	2
swapping threshold	64
L1	0
U	0
Equilibration	1
Mem alignment in double	8

Таблица 1: Параметры из HPL.dat.

Для Netlib Linpack число используемых потоков было определено на этапе компиляции библиотеки ATLAS (были сделаны две разные её копии для двух тестовых компьютеров). Для MP Linpack желаемое число используемых потоков передавалось в переменной среды `OMP_NUM_THREADS`.

SMP Linpack использует меньшее число параметров конфигурационного файла, чем остальные два теста. Для каждого размера задачи запрашиваются число запусков, размер ведущего измерения задачи и граница выравнивания данных в памяти. В этой работе число запусков варьировалось от двух до четырёх, размер ведущего измерения выбирался равным размеру задачи, а выравнивание данных в памяти равно 4 кбайтам. Конфигурацию используемого ЦПУ этот бенчмарк определяет сам.

Для запуска использующих MPI вариантов использовалось окружение MPICH версии 1.4 с менеджером процессов Hydra, строка запуска: `mpirun.hydra -n 1 ./xhpl`, т.е. они запускались в одном процессе. Таким образом, использование возможностей параллелизма сводилось к многопоточности алгоритмов линейной алгебры для Netlib HPL и для OpenMP-кода для MP Linpack.

SMP Linpack не требует дополнительных опций, кроме файла конфигурации, т.к. автоматически определяет доступные ресурсы при старте.

6 Результаты измерений и их анализ

Измерения производились на двух различных компьютерах.

Для запуска первой серии бенчмарков использовался следующий компьютер:

- Процессор — Intel(R) Xeon(R) X5450, 8 ядер.
- Частота процессора — 3 ГГц.
- Объем ОЗУ — 16 Гбайт.
- Операционная система — SUSE Linux Enterprise Server 10 (x86_64)

На рис. 1 приведены значения производительностей, сообщённых бенчмарками для различных размеров тестовых задач.

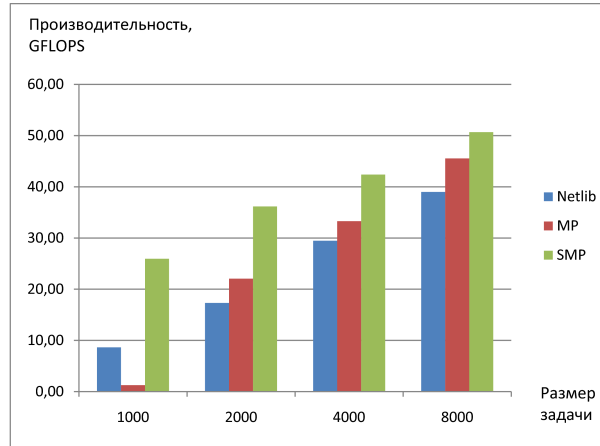


Рис. 1: Результаты измерений для первой машины.

Для запуска второй серии использовался следующий компьютер:

- Процессор — Intel(R) Xeon(R) 5150, 4 ядра.
- Частота процессора — 2,66 ГГц.
- Объем ОЗУ — 16 Гбайт.
- Операционная система — Red Hat Enterprise Linux Server release 5.4 (x86_64)

В табл. 2 приведены значения измеренных значений P_{max} , N_{max} , $N_{1/2}$, P_{peak} (их смысл подробно описан в [5]) для трёх вариантов бенчмарка. При значении $N = 64000$ все три бенчмарка сообщили о нехватке памяти.

Параметр	MP	HPL	SMP
P_{max} , Гфлопс	35,65	33,90	34,93
N_{max}	32000	32000	32000
$N_{1/2}$	1220	1800	500
P_{peak} , Гфлопс	42,56	42,56	42,56
эффективность	83,76%	79,65%	82,07%

Таблица 2: Полученные значения P_{max} , N_{max} , $N_{1/2}$, P_{peak} для второй машины.

На рис. 2 отображены зависимости продемонстрированной производительности от размера задач.

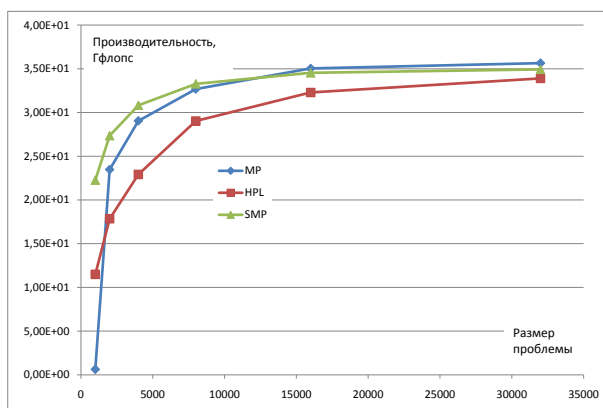


Рис. 2: Асимптотическая производительность бенчмарков для второй машины.

Почти для всех исследованных размеров задач производительность Netlib HPL оказалась меньше аналогичного показателя для MP Linpack. Провал для измерения MP Linpack для размера задачи $N = 1000$ можно объяснить следующим наблюдением: по каким-то причинам для задач малого размера данный тест не создавал дополнительные потоки и работал всего в одном. SMP Linpack же характеризуется наиболее быстрым выходом на максимальное значение для малых N . Это видно по наименьшему из всех значению $N_{1/2}$. В свою очередь, MP вариант показывал производительность несколько меньшую, чем SMP Linpack, для почти всех размеров задач. И только для наибольшего $N = 32000$ он вырвался вперед. Следует отметить, что с ростом размера задачи разрыв между всеми тремя бенчмарками сокращается, так как все они приближаются к теоретической пиковой производительности используемого железа.

7 Заключение

Несмотря на то, что SMP Linpack обладает возможностями динамического, не требующего вмешательства пользователя определения конфигурации процессоров и демонстрирует наилучшие результаты для SMP конфигураций, ему присущи недостатки, такие как закрытость кода, затрудняющее анализ его поведения, и ограниченность применимости системами с общей памятью.

Netlib Linpack с настройками по умолчанию имеет самый медленный рост производительности с увеличением размера задачи, однако показывает максимальное значение, меньше всего на 5% по сравнению с лидером. Несомненно, это разрыв может быть уменьшен правильной тонкой настройкой параметров алгоритма. MP Linpack представляет разумный компромисс между двумя вариантами.

Список литературы

- [1] *Garg, S.* Achieving large scale parallelism through operating system resource management on the Intel TFLOPS supercomputer. — 1998. <http://morden.csee.usf.edu/~chawla/ITJournal/tos.pdf>.
- [2] ATLAS linear algebra software. <http://www.netlib.org/atlas/>.
- [3] BLAS: Basic linear algebra software. <http://www.netlib.org/blas/>.
- [4] *Dongarra, J. J.* Performance of various computers using standard linear equations software / J. J. Dongarra. <ftp://netlib2.cs.utk.edu/benchmark/performance.pdf>.
- [5] *Dongarra, J. J.* The LINPACK benchmark: Past, present, and future. / J. J. Dongarra, P. Luszczek, A. Petitet // *Concurrency and Computation: Practice and Experience*. — 2003. — Vol. 15. — P. 2003.
- [6] GNU OpenMP implementation. <http://gcc.gnu.org/projects/gomp/>.
- [7] High performance Linpack benchmark. <http://www.netlib.org/linpack/>.
- [8] Intel Linpack benchmarks. <http://software.intel.com/en-us/articles/intel-math-kernel-library-linpack-download/>.
- [9] Intel Math Kernel Library. <http://software.intel.com/en-us/articles/intel-mkl/>.
- [10] Intel MPI library. <http://software.intel.com/en-us/articles/intel-mpi-library/>.
- [11] LINPACK Users' Guide / J. J. Dongarra, C. B. Moler, J. R. Bunch, G. Stewart. — 1979. — P. 320.
- [12] MPI: A Message-Passing Interface Standard. Version 2.2 / Message Passing Interface Forum. — 2009. — September. <http://www.mpi-forum.org/docs/docs.html>.
- [13] MPICH: high performance and widely portable MPI implementation. <http://www.mcs.anl.gov/mpi/mpich/>.
- [14] Open MPI: Open source high performance computing. <http://www.open-mpi.org/>.
- [15] *Henry, G.* The performance of the Intel TFLOPS supercomputer. — 1998. <http://morden.csee.usf.edu/~chawla/ITJournal/perf.pdf>.
- [16] TOP500 supercomputing sites. <http://www.top500.org>.