МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ УНИВЕРСИТЕТ имени академика С.П.КОРОЛЁВА

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Методики оценки эффективности параллельных вычислений и производительности суперкомпьютера «Сергей Королёв» и алгоритм-инструкция эффективного использования параллельных вычислений с применением имеющегося и вновь приобретаемого программного обеспечения

Электронное методическое пособие

Работа выполнена по мероприятию блока 2 «Развитие и повышение эффективности научно-инновационной деятельности» и блока 3 «Развитие информационной научно-образовательной среды и инфраструктуры» Программы развития СГАУ на 2009 – 2018 годы по проекту «Разработка комплекса технологий использования ресурсов суперкомпьютера «Сергей Королёв» в целях развития инновационной и научно-образовательной среды университета»

Соглашение № 2_5 от 3.06.2013 г.

САМАРА

2013

УДК004(075)

ББК32.9я7

М 545

Составители: Баскаков Александр Викторович, Симановский Евгений Аркадьевич.

Методики оценки эффективности параллельных вычислений и производительности суперкомпьютера «Сергей Королёв» и алгоритм-инструкция эффективного использования параллельных вычислений с применением имеющегося и вновь приобретаемого программного обеспечения.

[Электронный ресурс]: электрон. метод. пособие / М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т); сост.: А. В. Баскаков, Е.А. Симановский - Электрон. текстовые и граф. дан. (181 Кбайт). - Самара, 2013. - 1 эл. опт. диск (CD-ROM).

В методическом пособии приведен материал, необходимый для проведения оценки производительности суперкомпьютера “Сергей Королёв”. Приведены распространенные средства оценки производительности процессоров, кластеров, суперкомпьютеров. Изложены методики проведения оценки производительности для различных операционных систем.

Пособие предназначено для студентов, инженеров, научных работников, занимающихся разработкой аппаратно-программного обеспечения суперкомпьютерной техники.

Разработано в УИТ.

Оглавление

1 Оценки производительности супер-ЭВМ

Большинство оценочных характеристик производительности супер-ЭВМ связано с вычислениями над вещественными числами [1]. К ним относится пиковая производительность (ПП), измеряемая в млн. операций с плавающей точкой, которые компьютер теоретически может выполнить за 1 с (MFLOPS).

ПП - величина, практически не достижимая. Это связано с проблемами заполнения функциональных конвейерных устройств. Чем больше конвейер, тем больше надо "инициализационного" времени для того, чтобы его заполнить. Такие конвейеры эффективны при работе с длинными векторами. Поэтому для оценки векторных супер-ЭВМ было введено такое понятие, как **длина полупроизводительности** - длина вектора, при которой достигается половина пиковой производительности.

Более реальные оценки производительности базируются на временах выполнения различных тестов. Самыми хорошими тестами являются реальные задачи пользователя. Однако такие оценки, во-первых, весьма специфичны, а во-вторых, часто вообще недоступны или отсутствуют. Поэтому обычно применяются более универсальные тесты.

Поскольку большую часть времени выполнения программ обычно занимают циклы, иногда именно они применяются в качестве тестов, например, известные ливерморские циклы.

Наиболее популярным тестом производительности является **Linpack**, который представляет собой решение системы N линейных уравнений методом Гаусса. Так как известно, сколько операций с вещественными числами нужно проделать для решения системы, то, зная время расчета,

можно вычислить выполняемое в секунду количество операций.

Имеется несколько модификаций этих тестов. Обычно фирмы-производители компьютеров приводят результаты при N = 100.

Свободно распространяется стандартная программа на Фортране, которую надо выполнить на суперкомпьютере, чтобы получить результат тестирования. Эта программа не может быть изменена, за исключением замены вызовов подпрограмм, дающих доступ к процессорному времени выполнения.

Другой стандартный тест относится к случаю N = 1000, предполагающему использование длинных векторов. Эти тесты могут выполняться на компьютерах при разном числе процессоров, давая также оценки качества распараллеливания.

Для MPP-систем более интересным является тест Linpack-parallel, в котором производительность измеряется при больших N и большом количестве процессоров. Здесь лидером является 6768-процессорный Intel Paragon (281 GFLOPS при N = 128600). Что касается производительности процессоров, то при N = 100 лидирует Cray T916 (522 MFLOPS), при N = 1000 и по пиковой производительности - Hitachi S3800 (соответственно 6431 и 8000 MFLOPS). Для сравнения, процессор в AlphaServer 8400 имеет 140 MFLOPS при N =100 и 411 MFLOPS при N = 1000.

Для высокопараллельных суперкомпьютеров в последнее время все больше используются тесты NAS parallel benchmark, которые особенно хороши для задач вычислительной газо- и гидродинамики. Их недостатком является фиксация алгоритма решения, а не текста программы

Как известно, FLOPS – это единица измерения вычислительной

мощности компьютеров в операциях с плавающей точкой. Поэтому для выяснения возможностей супер- и просто компьютеров существуют чуть более приближенные к реальным вычислительным задачам бенчмарки, например, SPEC [2]: SPECint и SPECfp. И, тем не менее, FLOPS активно используется в оценках производительности и публикуется в отчетах. Для его измерения давно уже использовали тест Linpack, а сейчас применяют открытый стандартный бенчмарк из LAPACK.

Итак, FLOPS – это количество вычислительных операций или инструкций, выполняемых над операндами с плавающей точкой (FP) в секунду. Здесь используется слово «вычислительных», так как микропроцессор умеет выполнять и другие инструкции с такими операндами, например, загрузку из памяти. Такие операции не несут полезной вычислительной нагрузки и поэтому не учитываются.

Значение FLOPS, опубликованное для конкретной системы, – это характеристика прежде всего самого компьютера, а не программы. Ее можно получить двумя способами – теоретическим и практическим. Теоретически мы знаем сколько микропроцессоров в системе и сколько исполняемых устройств с плавающей точкой в каждом процессоре. Все они могут работать одновременно и начинать работу над следующей инструкцией в конвеере каждый цикл. Поэтому для подсчета теоретического максимума для данной системы нам нужно только перемножить все эти величины с частотой процессора – получим количество FP операций в секунду.

Практическое измерение заключается в запуске бенчмарка Linpack. Бенчмарк осуществляет операцию умножения матрицы на матрицу несколько десятков раз и вычисляет усредненное значение времени выполнения теста.

Так как количество FP операций в имплементации алгоритма известно заранее, то разделив одно значение на другое, получим искомое FLOPS. Библиотека Intel MKL (Math Kernel Library) содержит пакет LAPACK — пакет библиотек для решения задач линейной алгебры. Бенчмарк построен на основе этого пакета. Считается, что его эффективность находится на уровне 90% от теоретически возможной, что позволяет бенчмарку считаться «эталонным измерением».

Очевидно, что разработчики высокопроизводительных приложений хотели бы оценить эффективность имплементации своих алгоритмов, используя показатель FLOPS, но уже померянный для своего приложения. Сравнение измеренного FLOPS с «эталонным» дает представление о том, насколько далека производительность их алгоритма от идеальной и каков теоретический потенциал ее улучшения. Для этого нужно знать минимальное количество FP операций, требуемое для выполнения алгоритма, и точно измерить время выполнения программы (или ее части, выполняющей оцениваемый алгоритм). Такие результаты, наряду с измерениями характеристик шины памяти, нужны для того, чтобы понять, где реализация алгоритма упирается в возможности аппаратной системы и что является лимитирующим фактором: пропускная способность памяти, задержки передачи данных, производительность алгоритма, либо системы.

У нас есть три оценки/измерения FLOPS: теоретическая, бенчмарк и программа. Рассмотрим особенности вычисления FLOPS для каждого случая.

2 Теоретическая оценка FLOPS для системы

Чтобы понять, как подсчитывается количество одновременных

операций в процессоре, надо учесть характеристики устройства блока out-of-order в конвеере процессора Intel Sandy Bridge.

Здесь у нас 6 портов к вычислительным устройствам, при этом, за один цикл (или такт процессора) диспетчером может быть назначено на выполнение до 6 микроопераций: 3 операции с памятью и 3 вычислительные. Одновременно могут выполняться одна операция умножения (MUL) и одна сложения (ADD), как в блоках x87 FP, так и в SSE, либо AVX. С учетом ширины SIMD регистров 256 бит мы может получить следующие результаты:

8 MUL (32-bit) и 8 ADD (32-bit): **16 SP FLOP/cycle**, то есть 16 операций с плавающей точкой одинарной точности за один такт.

4 MUL (64-bit) и 4 ADD (64-bit): **8 DP FLOP/cycle**, то есть 8 операций с плавающей точкой двойной точности за один такт.

Теоретическое пиковое значение FLOPS для доступного мне 1-сокетного Xeon E3-1275 (4 cores @ 3.574GHz) составляет:

16 (FLOP/cycle)*4*3.574 (Gcycles/sec) = **228 GFLOPS SP**

8 (FLOP/cycle)*4*3.574 (Gcycles/sec) = **114 GFLOPS DP**

2.1 Запуск бенчмарка Linpack

Запускам бенчмарк на системе и получаем следующие результаты:

```
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --  September 10, 2008
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

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 : 50400
NB : 168
PMAP : Row-major process mapping
P : 3
Q : 4
PFACT : Left
NBMIN : 4
NDIV : 2
RFACT : Left
BCAST : 1ringM
DEPTH : 0
SWAP : Mix (threshold = 256)
L1 : transposed form
U : transposed form
EQUIL : no
ALIGN : 8 double precision words

Column=000336 Fraction=0.005 Mflops=126818.41

Здесь нужно отметить, как именно учитываются FP операции в бенчмарке. Как уже упоминалось, тест заранее «знает» количество операций MUL и ADD, которые необходимы для перемножения матриц. В упрощенном представлении: производится решение системы линейных уравнений Ax=b (несколько тысяч штук) путем перемножения плотных матриц действительных чисел (real8) размером MxK, а количество операций сложения и умножения, необходимых для реализации алгоритма, считается (для симметричной матрицы) Nflop = 2*(M^3)+(M^2). Вычисления

производятся для чисел с двойной точностью, как и для большинства бенчмарков. Сколько операций с плавающей точкой действительно выполняется в реализации алгоритма, пользователей не волнует, хотя они догадываются, что больше. Это связано с тем, что выполняется декомпозиция матриц по блокам и преобразование (факторизация) для достижения максимальной производительности алгоритма на вычислительной платформе. То есть нужно запомнить, что на самом деле значение физических FLOPS занижено за счет неучитывания лишних операций преобразования и вспомогательных операций типа сдвигов.

3 Оценка FLOPS программы

Чтобы исследовать соизмеримые результаты, в качестве высокопроизводительного приложения необходимо использовать пример перемножения матриц, созданный пользователем. Пример реализации перемножения матриц, написанный на языке C, можно найти в директории Samples пакета Intel VTune Amplifier XE. Воспользуемся формулой Nflop=2*(M^3) для подсчета FP операций (исходя из базового алгоритма перемножения матриц) и померим время выполнения перемножения для случая алгоритма multiply3 при размере симметричных матриц M=4096. Для того, чтобы получить эффективный код, надо использовать опции оптимизации *–O3* (агрессивная оптимизация циклов) и *–xavx* (использовать инструкции AVX) C-компилятора Intel для того, чтобы сгенерировались векторные SIMD-инструкции для исполнительных устройств AVX. Компилятор даст узнать, векторизовался ли цикл перемножения матрицы. Для этого необходимо указать опцию *–vec-report3*. В результатах компиляции

должны быть сообщения оптимизатора: «LOOP WAS VECTORIZED» напротив строки с телом внутреннего цикла в файле *multiply.c*.

Необходимо проверить, какие инструкции сгенерированы компилятором для цикла перемножения.

$icl –g –O3 –xavx –S

По тэгу __tag_value_multiply3 надо найти нужный цикл — инструкции правильные.

$vi muliply3.s

Результат выполнения программы (~7 секунд) нам дает следующее значение FLOPS = 2*4096*4096*4096/7[s] = **19.6 GFLOPS.**

3.1 Измерение FLOPS программы

Существуют задачи в линейной алгебре, программную имплементацию которых очень сложно оценить в количестве FP операций, в том смысле, что нахождение такой оценки само является нетривиальной математической задачей. Как считать FLOPS для программы? Есть два пути, оба экспериментальных: трудный, дающий точный результат, и легкий, но обеспечивающий приблизительную оценку. В первом случае придется взять некую базовую программную имплементацию решения задачи, скомпилировать ее в ассемблерные инструкции и, выполнив их на симуляторе процессора, посчитать количество FP операций. Следует учитывать, что если ветвление исполнения задачи будет зависеть от входных данных, то вся точность оценки сразу поставится под сомнение.

Второй путь предполагает подсчет количества FP инструкций,

выполненных процессором. Счетчики производительности (PMU), умеют считать, сколько микроопераций было выполнено на том или ином вычислительном блоке. С такими счетчиками умеет работать VTune Amplifier XE.

Несмотря на то, что VTune имеет множество встроенных профилей, специального профиля для измерения FLOPS у него пока нет.

Процесс создания профиля и сбора данных:

1. Создаем новый проект и указываем в качестве target application наше приложение *matrix*.
2. Выбираем профиль Lightweight Hotspots (который использует технологию сэмплирования счетчиков процессора Hadware Event-based Sampling) и копируем его для создания пользовательского профиля. Обзываем его My FLOPS Analysis.
3. Редактируем профиль, добавляем туда новые процессорные счетчики событий процессора Sandy Bridge (Events). На них остановимся чуть подробнее. В их названии зашифрованы исполнительные устройства (x87, SSE, AVX) и тип данных, над которыми выполнялась операция. Каждый такт процессора счетчики складывают количество вычислительных операций, назначенных на исполнение. На всякий случай мы добавили счетчики на все возможные операции с FP:

- ✓ FP_COMP_OPS_EXE. SSE_PACKED_DOUBLE – векторы (PACKED) данных двойной точности (DOUBLE)
- ✓ FP_COMP_OPS_EXE. SSE_PACKED_SINGLE – векторы данных

одинарной точности

- ✓ FP_COMP_OPS_EXE. SSE_SCALAR_DOUBLE – скалярые DP
- ✓ FP_COMP_OPS_EXE. SSE_ SCALAR _SINGLE – скалярные SP
- ✓ SIMD_FP_256.PACKED_DOUBLE – векторы AVX данных DP
- ✓ SIMD_FP_256.PACKED_SINGLE – векторы AVX данных SP
- ✓ FP_COMP_OPS_EXE.x87 – скалярные данные x87

В полученных результатах надо переключиться в Hardware Events viewpoint и копировать количетво events, собранных для функции *multiply3*: 34,648,000,000.

Далее необходимо подсчитать значения FLOPS по формулам. Операции с данными двойной точности выполняются одновременно над четырмя 64-битными DP операндами в 256-битном регистре, поэтому надо умножить на коэффициент 4. Данные с одинарной точностью, соответственно, умножаем на 8. В последней формуле не умножаем количество инструкций на коэффициент, так как операции сопроцессора x87 выполняются только со скалярными величинами. Если в программе выполняется несколько разных типов FP операций, то их количество, умноженное на коэффициенты, суммируется для получения результирующего FLOPS.

FLOPS = 4 * SIMD_FP_256.PACKED_DOUBLE / Elapsed Time

FLOPS = 8 * SIMD_FP_256.PACKED_SINGLE / Elapsed Time

FLOPS = (FP_COMP_OPS_EXE.x87) / Elapsed Time

В программе выполнялись только AVX инструкции, поэтому в результатах есть значение только одного счетчика SIMD_FP_256.PACKED_DOUBLE.

Необходимо удостовериться, что данные события собраны для нашего цикла в функции *multiply3* (переключившись в Source View):

FLOPS = 4 *34.6Gops/7s = **19.7 GFlops**

Значение вполне соответствует оценочному, подсчитанному в предыдущем пункте. Поэтому с достаточной долей точности можно говорить о том, что результаты оценочного метода и измерительного совпадают. Однако, существуют случаи, когда они могут не совпадать.

4 Тесты производительности

В настоящее время разработано множество различных тестов для оценки производительности процессоров, вычислительных систем, суперкомпьютеров. Некоторые из них приведены в таблице:

Название	Краткое описание
C LINPACK	Тест LINPACK, переписанный на языке C.
CPU2	Набор из 34 фортрановских программ, интенсивно работающих с вещественной арифметикой. Производительность измеряется в единицах, эквивалентных производительности MicroVAX II.
Dhrystone	Тест целочисленной арифметики, показателен в системном программировании. Не учитывает производительности кэш-памяти.
Flops	Вычисляет производительность в MFLOPS на определенных последовательностях инструкций FADD, FSUB, FMUL и FDIV. Работает как на скалярных, так и на векторных машинах.
Heapsort	Целочисленная программа, сортирующая 2MB-массив из целых чисел.
LFK (Livermore Loops)	Состоит из 24 циклов, представляющих собой характерные вычисления из различных областей прикладной физики. Вычисляет эффективность соответствующих фрагментов в MFLOPS с тремя

	наборами длин векторов.
LINPACK	Тест состоит в решении системы линейных уравнений с помощью LU-факторизации. Основное время затрачивается на векторные операции типа FMA (умножение и сложение). Производительность определяется как количество "полезных" вычислительных операций над числами с плавающей точкой в расчете на 1 секунду, и выражается в Мфлоп/сек (миллионах операций в секунду). Число выполненных операций с плавающей точкой оценивается по формуле $2n^3/3 + 2n^2$ (здесь n - размер задачи, т.е. матрица имеет размеры n x n). Таким образом, при увеличении размера матрицы в 2 раза, объем используемой памяти увеличивается примерно в 4 раза, а объем вычислений - примерно в 8 раз. Есть версии, работающие с матрицами 100x100 и 1000x1000, а также с варьируемым размером матрицы (LINPACK HPC) [3]. Автор теста – Jack Dongarra. Результаты теста используются при составлении рейтинга Top500. HPL - реализация на языке Си, причем обмены между процессорами выполняются через процедуры интерфейса MPI, а вычисления на каждом процессоре - с помощью вызовов процедур BLAS. В качестве BLAS можно использовать библиотеку ATLAS.

	SLbench (1.2MB) - параллельная версия LINPACK, использующая ScaLAPACK и BLACS (необходимо наличие реализации MPI). Доступна Java-версия.
Matrix Multiply (MM)	Тест содержит 9 различных программ умножения матриц (размером 500x500). Оценивается работа кэш-памяти и уровень оптимизации компилятора.
NAS Kernels	Последовательная версия NAS CFD (computational fluid dynamics)
NPB (NAS Parallel Benchmarks)	Состоит из 8 различных программ для определения производительности параллельных компьютеров. Программы взяты из реальных аэрокосмических расчетных пакетов.
PERFECT	Представляет собой комплект из 13 прикладных Fortran-программ, представляющих четыре типа вычислительных задач - аэро- и гидродинамики, моделирования химических и физических процессов, инженерного проектирования, а также обработки сигналов. Выполняется дважды - до и после оптимизации исходных текстов.
SLALOM	Масштабируемый тест производительности для суперкомпьютеров. Оценивает объем вычислений, который может произвести компьютер за одну минуту.
Stanford	Тестовый набор, состоящий из 8 целочисленных

	тестов (умножение матриц, сортировка 3 методами, перестановки, ханойская башня, растановка 8 ферзей, головоломка) и 2 тестов на вещественные вычисления (быстрое преобразование Фурье, перемножение матриц)
STREAM	Синтетический тест*, оценивающий скорость работы с памятью с простой арифметикой и без. Основан на измерении времени выполнения больших векторных операций: копирование по памяти, умножение на константу, сложение, умножение и сложение. Доступны версии на Фортране и Си. Результатами теста пользуются все ведущие разработчики высокопроизводительной техники.
Whetstone	Синтетический тест*, ориентированный на численное программирование (с плавающей запятой). Не учитывает кэш. Компилятор легко можно оптимизировать под Whetstone.

4.1 Тест производительности для суперкомпьютера «Сергей Королёв»

Суперкомпьютер «Сергей Королёв» построен на базе линейки оборудования IBM BladeCenter с использованием блейд-серверов HS22 и обеспечивает пиковую производительность 15 триллионов операций с плавающей точкой в секунду [4].

Технические характеристики суперкомпьютера "Сергей Королёв":

- Общие характеристики:
    - Общее число процессоров / вычислительных ядер: 272/1184;
    - Общее число графических процессоров/ядер: 4/1720;
    - Общая оперативная память: 2442 Гб;
    - Тип системной сети: QLogic/Voltaire InfiniBand DDR, QDR;
    - Тип управляющей вспомогательной сети: Gigabit Ethernet.
    - 112 BladeCenter HS22 вычислительных блейд-серверов:
    - Процессоры: 2x Intel Xeon X5560, 2.80GHz, 8MB Cache;
    - Оперативная память: 12 Гб;
    - Жесткий диск: 76Гб или 146Гб SAS 10K 16Mb Cache;
    - Тип системной сети: InfiniBand DDR, QDR;
- 14 HS22 блейд-серверов с памятью 24 Гб на сервер:
    - Процессоры: 2x Intel Xeon X5670, 2.93GHz, 12MB Cache;
    - Оперативная память: 24 Гб;
    - Тип системной сети: InfiniBand QDR;
- 8 HS22 блейд-серверов с памятью 96 Гб на сервер:
    - Процессоры: 2x Intel Xeon X5670, 2.93GHz, 12MB Cache;
    - Оперативная память: 96 Гб;
    - Жесткий диск: 244Гб RAID0 из 2-х дисков SAS;
    - Тип системной сети: InfiniBand QDR;
- 2 HS22 блейд-сервера с графическими картами Nvidia Tesla 2070:
    - Процессоры: 2x Intel Xeon X5670, 2.93GHz, 12MB Cache;

    - Оперативная память: 24 Гб;
    - Тип системной сети: InfiniBand QDR;
    - GPGPU: 2x Nvidia Tesla 2070, 448 CUDA cores, 6Gb GRAM
- 2 IBM x3650M управляющих сервера:
    - Процессоры: 2x Intel Xeon X5560, 2.80GHz, 8MB Cache;
    - Оперативная память:12 Гб.
- 2 IBM x3650M3 сервера для работы файловой системы GPFS:
    - Процессоры: 2x Intel Xeon X5620, 2.40GHz, 12MB Cache;
    - Оперативная память:12 Гб.
- Сетевой интерконнект:
    - QLogic 9080 Infiniband DDR 20Гбит/с;
    - QLogic 12800 Infiniband QDR 40Гбит/с.
- Операционная система: Red Hat Enterprise Linux 5.6

Лог-файл теста призводительности LINPACK одного вычислительного сервера (блейд-сервера):

```
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --  September 10, 2008
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

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 : 50400
NB : 168
PMAP : Row-major process mapping
P : 3
Q : 4
PFACT : Left
NBMIN : 4
NDIV : 2
RFACT : Left
BCAST : 1ringM
DEPTH : 0
SWAP : Mix (threshold = 256)
L1 : transposed form
U : transposed form
EQUIL : no
ALIGN : 8 double precision words

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
*||Ax-b||_oo / (eps * (|| x ||_oo * || A ||_oo + || b ||_oo) * N)*
- The relative machine precision (eps) is taken to be 1.110223e-16
- Computational tests pass if scaled residuals are less than 16.0

Column=000336 Fraction=0.005 Mflops=126818.41
Column=000672 Fraction=0.010 Mflops=126562.85
Column=000840 Fraction=0.015 Mflops=126792.64
Column=001176 Fraction=0.020 Mflops=126519.19
Column=001344 Fraction=0.025 Mflops=126462.26
Column=001680 Fraction=0.030 Mflops=126531.54
Column=001848 Fraction=0.035 Mflops=126348.81
Column=002184 Fraction=0.040 Mflops=126494.50
Column=002352 Fraction=0.045 Mflops=126360.06
Column=002688 Fraction=0.050 Mflops=126340.45
Column=002856 Fraction=0.055 Mflops=126399.49
Column=003024 Fraction=0.060 Mflops=126270.14
Column=003360 Fraction=0.065 Mflops=126292.26
Column=003528 Fraction=0.070 Mflops=126324.58

Column=003864 Fraction=0.075 Mflops=126226.50
Column=004200 Fraction=0.080 Mflops=126291.20
Column=004368 Fraction=0.085 Mflops=126211.77
Column=004704 Fraction=0.090 Mflops=126176.35
Column=004872 Fraction=0.095 Mflops=126198.29
Column=005208 Fraction=0.100 Mflops=126116.44
Column=005376 Fraction=0.105 Mflops=126102.20
Column=005712 Fraction=0.110 Mflops=126116.10
Column=005880 Fraction=0.115 Mflops=126053.56
Column=006216 Fraction=0.120 Mflops=126076.69
Column=006384 Fraction=0.125 Mflops=126018.74
Column=006720 Fraction=0.130 Mflops=125985.09
Column=006888 Fraction=0.135 Mflops=126005.17
Column=007224 Fraction=0.140 Mflops=125943.49
Column=007392 Fraction=0.145 Mflops=125921.62
Column=007728 Fraction=0.150 Mflops=125927.77
Column=007896 Fraction=0.155 Mflops=125886.77
Column=008232 Fraction=0.160 Mflops=125902.62
Column=008400 Fraction=0.165 Mflops=125878.28
Column=008736 Fraction=0.170 Mflops=125849.34
Column=008904 Fraction=0.175 Mflops=125869.26
Column=009240 Fraction=0.180 Mflops=125830.31
Column=009408 Fraction=0.185 Mflops=125825.25
Column=009744 Fraction=0.190 Mflops=125825.99
Column=009912 Fraction=0.195 Mflops=125800.02
Column=010248 Fraction=0.200 Mflops=125817.45
Column=010416 Fraction=0.205 Mflops=125789.38
Column=010752 Fraction=0.210 Mflops=125770.55
Column=011424 Fraction=0.225 Mflops=125739.00
Column=011760 Fraction=0.230 Mflops=125739.06
Column=011928 Fraction=0.235 Mflops=125710.88
Column=012264 Fraction=0.240 Mflops=125716.84
Column=012432 Fraction=0.245 Mflops=125694.22
Column=012768 Fraction=0.250 Mflops=125668.33
Column=012936 Fraction=0.255 Mflops=125673.79
Column=013272 Fraction=0.260 Mflops=125644.15
Column=013440 Fraction=0.265 Mflops=125632.92
Column=013776 Fraction=0.270 Mflops=125634.12
Column=013944 Fraction=0.275 Mflops=125606.74
Column=014280 Fraction=0.280 Mflops=125610.65
Column=014448 Fraction=0.285 Mflops=125591.55
Column=014784 Fraction=0.290 Mflops=125562.75
Column=014952 Fraction=0.295 Mflops=125568.38
Column=015288 Fraction=0.300 Mflops=125538.50
Column=015456 Fraction=0.305 Mflops=125525.50

Column=015792 Fraction=0.310 Mflops=125517.81
Column=015960 Fraction=0.315 Mflops=125492.65
Column=016296 Fraction=0.320 Mflops=125488.71
Column=016464 Fraction=0.325 Mflops=125470.03
Column=016800 Fraction=0.330 Mflops=125446.77
Column=016968 Fraction=0.335 Mflops=125443.67
Column=017304 Fraction=0.340 Mflops=125416.86
Column=017472 Fraction=0.345 Mflops=125405.14
Column=017808 Fraction=0.350 Mflops=125392.19
Column=017976 Fraction=0.355 Mflops=125374.52
Column=018312 Fraction=0.360 Mflops=125366.39
Column=018480 Fraction=0.365 Mflops=125345.15
Column=018816 Fraction=0.370 Mflops=125315.93
Column=018984 Fraction=0.375 Mflops=125314.24
Column=019320 Fraction=0.380 Mflops=125282.30
Column=019488 Fraction=0.385 Mflops=125272.43
Column=019824 Fraction=0.390 Mflops=125259.97
Column=019992 Fraction=0.395 Mflops=125236.95
Column=020328 Fraction=0.400 Mflops=125231.59
Column=020496 Fraction=0.405 Mflops=125211.89
Column=020832 Fraction=0.410 Mflops=125188.14
Column=021000 Fraction=0.415 Mflops=125182.40
Column=021336 Fraction=0.420 Mflops=125154.24
Column=021504 Fraction=0.425 Mflops=125143.92
Column=021840 Fraction=0.430 Mflops=125129.19
Column=022008 Fraction=0.435 Mflops=125111.44
Column=022344 Fraction=0.440 Mflops=125102.14
Column=022512 Fraction=0.445 Mflops=125082.83
Column=022848 Fraction=0.450 Mflops=125057.33
Column=023016 Fraction=0.455 Mflops=125055.66
Column=023352 Fraction=0.460 Mflops=125024.90
Column=023520 Fraction=0.465 Mflops=125013.38
Column=023856 Fraction=0.470 Mflops=124997.02
Column=024024 Fraction=0.475 Mflops=124979.34
Column=028056 Fraction=0.555 Mflops=124704.98
Column=029064 Fraction=0.575 Mflops=124641.24
Column=030072 Fraction=0.595 Mflops=124564.13
Column=031080 Fraction=0.615 Mflops=124497.06
Column=032088 Fraction=0.635 Mflops=124422.90
Column=033096 Fraction=0.655 Mflops=124360.39
Column=034104 Fraction=0.675 Mflops=124288.18
Column=035112 Fraction=0.695 Mflops=124225.62
Column=040152 Fraction=0.795 Mflops=123900.06
Column=045192 Fraction=0.895 Mflops=123649.05
Column=050232 Fraction=0.995 Mflops=123557.37

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01L2L4       50400   168     3     4             691.32             1.235e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0032228 ..... 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.
================================================================================
```

Лог-файл теста призводительности LINPACK пары вычислительных серверов:

```
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

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      :   71232
NB     :     168
PMAP   : Row-major process mapping
P      :       4
```

Q : 6
PFACT : Right
NBMIN : 4
NDIV : 2
RFACT : Right
BCAST : 1ringM
DEPTH : 0
SWAP : Mix (threshold = 256)
L1 : transposed form
U : transposed form
EQUIL : no
ALIGN : 8 double precision words

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
*||Ax-b||_oo / (eps * (|| x ||_oo * || A ||_oo + || b ||_oo) * N)*
- The relative machine precision (eps) is taken to be 1.110223e-16
- Computational tests pass if scaled residuals are less than 16.0

Column=000504 Fraction=0.005 Mflops=242403.63
Column=000840 Fraction=0.010 Mflops=246923.17
Column=001176 Fraction=0.015 Mflops=248879.94
Column=001512 Fraction=0.020 Mflops=249208.78
Column=001848 Fraction=0.025 Mflops=249955.26
Column=002184 Fraction=0.030 Mflops=250643.85
Column=002520 Fraction=0.035 Mflops=250547.82
Column=012936 Fraction=0.180 Mflops=251315.25
Column=013272 Fraction=0.185 Mflops=251343.04
Column=013608 Fraction=0.190 Mflops=251274.86
Column=013944 Fraction=0.195 Mflops=251265.22
Column=014280 Fraction=0.200 Mflops=251298.56
Column=014616 Fraction=0.205 Mflops=251219.12
Column=015120 Fraction=0.210 Mflops=251241.92
Column=015456 Fraction=0.215 Mflops=251172.39
Column=015792 Fraction=0.220 Mflops=251156.93
Column=016128 Fraction=0.225 Mflops=251152.83
Column=016464 Fraction=0.230 Mflops=251137.96
Column=016800 Fraction=0.235 Mflops=251080.05
Column=017136 Fraction=0.240 Mflops=251125.99
Column=017472 Fraction=0.245 Mflops=251065.97
Column=017976 Fraction=0.250 Mflops=251049.98
Column=018312 Fraction=0.255 Mflops=251063.08
Column=018648 Fraction=0.260 Mflops=250987.67

Column=018984 Fraction=0.265 Mflops=250983.49
Column=019320 Fraction=0.270 Mflops=250985.28
Column=019656 Fraction=0.275 Mflops=250930.14
Column=019992 Fraction=0.280 Mflops=250909.12
Column=020328 Fraction=0.285 Mflops=250914.08
Column=020664 Fraction=0.290 Mflops=250853.54
Column=021168 Fraction=0.295 Mflops=250850.49
Column=021504 Fraction=0.300 Mflops=250786.82
Column=021840 Fraction=0.305 Mflops=250770.06
Column=022176 Fraction=0.310 Mflops=250754.74
Column=022512 Fraction=0.315 Mflops=250725.94
Column=022848 Fraction=0.320 Mflops=250677.24
Column=023184 Fraction=0.325 Mflops=250691.26
Column=023520 Fraction=0.330 Mflops=250635.50
Column=024024 Fraction=0.335 Mflops=250601.00
Column=024360 Fraction=0.340 Mflops=250603.01
Column=024696 Fraction=0.345 Mflops=250536.96
Column=025032 Fraction=0.350 Mflops=250521.72
Column=025368 Fraction=0.355 Mflops=250507.65
Column=025704 Fraction=0.360 Mflops=250453.10
Column=026040 Fraction=0.365 Mflops=250428.60
Column=026376 Fraction=0.370 Mflops=250427.28
Column=026880 Fraction=0.375 Mflops=250341.82
Column=027216 Fraction=0.380 Mflops=250346.40
Column=027552 Fraction=0.385 Mflops=250288.88
Column=027888 Fraction=0.390 Mflops=250259.26
Column=028224 Fraction=0.395 Mflops=250244.32
Column=028560 Fraction=0.400 Mflops=250215.73
Column=028896 Fraction=0.405 Mflops=250164.60
Column=029232 Fraction=0.410 Mflops=250168.35
Column=029568 Fraction=0.415 Mflops=250116.38
Column=030072 Fraction=0.420 Mflops=250082.43
Column=030408 Fraction=0.425 Mflops=250071.21
Column=030744 Fraction=0.430 Mflops=250010.91
Column=031080 Fraction=0.435 Mflops=249982.50
Column=031416 Fraction=0.440 Mflops=249969.28
Column=031752 Fraction=0.445 Mflops=249919.96
Column=032088 Fraction=0.450 Mflops=249889.54
Column=032424 Fraction=0.455 Mflops=249880.92
Column=032928 Fraction=0.460 Mflops=249800.14
Column=033264 Fraction=0.465 Mflops=249793.35
Column=033600 Fraction=0.470 Mflops=249744.71
Column=033936 Fraction=0.475 Mflops=249710.68
Column=034272 Fraction=0.480 Mflops=249690.71
Column=034608 Fraction=0.485 Mflops=249657.10

Column=034944 Fraction=0.490 Mflops=249609.68
Column=035280 Fraction=0.495 Mflops=249601.89
Column=036792 Fraction=0.515 Mflops=249441.77
Column=039648 Fraction=0.555 Mflops=249161.04
Column=040992 Fraction=0.575 Mflops=249026.34
Column=042504 Fraction=0.595 Mflops=248907.21
Column=043848 Fraction=0.615 Mflops=248764.29
Column=045360 Fraction=0.635 Mflops=248637.04
Column=046704 Fraction=0.655 Mflops=248508.17
Column=048216 Fraction=0.675 Mflops=248366.95
Column=049560 Fraction=0.695 Mflops=248258.00
Column=056784 Fraction=0.795 Mflops=247633.84
Column=063840 Fraction=0.895 Mflops=247149.30
Column=070896 Fraction=0.995 Mflops=246969.26

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01R2R4       71232   168     4     6             976.08              2.469e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0024091 .....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.
================================================================================
```

Лог-файл теста призводительности LINPACK суперкомпьютера «Сергей Королёв»:

```
================================================================================
HPLinpack 2.0  --  High-Performance Linpack benchmark  --   September 10, 2008
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
================================================================================

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 : 411768
NB : 168
PMAP : Row-major process mapping
P : 30
Q : 36
PFACT : Right
NBMIN : 4
NDIV : 2
RFACT : Right
BCAST : 1ringM
DEPTH : 0
SWAP : Mix (threshold = 256)
L1 : transposed form
U : transposed form
EQUIL : no
ALIGN : 8 double precision words

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
*||Ax-b||_oo / (eps * (|| x ||_oo * || A ||_oo + || b ||_oo) * N)*
- The relative machine precision (eps) is taken to be 1.110223e-16
- Computational tests pass if scaled residuals are less than 16.0

Column=002184 Fraction=0.005 Mflops=5631435.84
Column=004200 Fraction=0.010 Mflops=6368295.15
Column=006216 Fraction=0.015 Mflops=7006757.43
Column=008400 Fraction=0.020 Mflops=7580629.88
Column=010416 Fraction=0.025 Mflops=8037275.82
Column=012432 Fraction=0.030 Mflops=8344651.95
Column=014448 Fraction=0.035 Mflops=8586200.40
Column=016632 Fraction=0.040 Mflops=8811448.85
Column=018648 Fraction=0.045 Mflops=8959268.94
Column=020664 Fraction=0.050 Mflops=9103830.13
Column=022680 Fraction=0.055 Mflops=9221279.10
Column=024864 Fraction=0.060 Mflops=9317833.15

Column=026880 Fraction=0.065 Mflops=9409303.31
Column=028896 Fraction=0.070 Mflops=9488814.57
Column=030912 Fraction=0.075 Mflops=9542144.24
Column=033096 Fraction=0.080 Mflops=9607205.60
Column=035112 Fraction=0.085 Mflops=9663462.62
Column=037128 Fraction=0.090 Mflops=9699075.65
Column=039144 Fraction=0.095 Mflops=9744684.05
Column=041328 Fraction=0.100 Mflops=9788596.18
Column=043344 Fraction=0.105 Mflops=9813179.85
Column=045360 Fraction=0.110 Mflops=9849408.00
Column=047376 Fraction=0.115 Mflops=9878315.09
Column=049560 Fraction=0.120 Mflops=9899359.80
Column=051576 Fraction=0.125 Mflops=9924817.38
Column=053592 Fraction=0.130 Mflops=9946862.44
Column=055608 Fraction=0.135 Mflops=9961248.32
Column=057792 Fraction=0.140 Mflops=9981796.45
Column=059808 Fraction=0.145 Mflops=10002248.97
Column=061824 Fraction=0.150 Mflops=10011947.95
Column=063840 Fraction=0.155 Mflops=10027118.21
Column=066024 Fraction=0.160 Mflops=10044766.92
Column=068040 Fraction=0.165 Mflops=10050924.31
Column=070056 Fraction=0.170 Mflops=10066983.19
Column=072072 Fraction=0.175 Mflops=10079237.16
Column=074256 Fraction=0.180 Mflops=10086950.06
Column=076272 Fraction=0.185 Mflops=10099696.72
Column=078288 Fraction=0.190 Mflops=10109536.22
Column=080304 Fraction=0.195 Mflops=10114606.82
Column=082488 Fraction=0.200 Mflops=10126027.65
Column=084504 Fraction=0.205 Mflops=10135130.33
Column=086520 Fraction=0.210 Mflops=10138703.70
Column=088536 Fraction=0.215 Mflops=10146586.83
Column=090720 Fraction=0.220 Mflops=10155142.66
Column=092736 Fraction=0.225 Mflops=10157487.18
Column=094752 Fraction=0.230 Mflops=10165797.19
Column=096768 Fraction=0.235 Mflops=10171942.27
Column=098952 Fraction=0.240 Mflops=10173598.30
Column=100968 Fraction=0.245 Mflops=10180348.74
Column=102984 Fraction=0.250 Mflops=10185271.72
Column=105168 Fraction=0.255 Mflops=10187182.73
Column=107184 Fraction=0.260 Mflops=10192245.86
Column=109200 Fraction=0.265 Mflops=10193105.48
Column=111216 Fraction=0.270 Mflops=10197743.53
Column=113400 Fraction=0.275 Mflops=10201432.84
Column=115416 Fraction=0.280 Mflops=10201867.25
Column=117432 Fraction=0.285 Mflops=10205746.76

Column=119448 Fraction=0.290 Mflops=10209882.95
Column=121632 Fraction=0.295 Mflops=10209878.56
Column=123648 Fraction=0.300 Mflops=10213452.05
Column=125664 Fraction=0.305 Mflops=10217820.30
Column=127680 Fraction=0.310 Mflops=10216863.99
Column=129864 Fraction=0.315 Mflops=10220542.63
Column=131880 Fraction=0.320 Mflops=10223360.36
Column=133896 Fraction=0.325 Mflops=10222308.22
Column=135912 Fraction=0.330 Mflops=10225260.04
Column=138096 Fraction=0.335 Mflops=10227305.64
Column=140112 Fraction=0.340 Mflops=10225904.73
Column=142128 Fraction=0.345 Mflops=10228202.90
Column=144144 Fraction=0.350 Mflops=10229798.85
Column=146328 Fraction=0.355 Mflops=10228787.43
Column=148344 Fraction=0.360 Mflops=10230625.41
Column=150360 Fraction=0.365 Mflops=10232794.78
Column=152376 Fraction=0.370 Mflops=10230361.08
Column=154560 Fraction=0.375 Mflops=10232098.59
Column=156576 Fraction=0.380 Mflops=10233454.22
Column=158592 Fraction=0.385 Mflops=10230895.89
Column=160608 Fraction=0.390 Mflops=10233137.52
Column=162792 Fraction=0.395 Mflops=10233708.65
Column=164808 Fraction=0.400 Mflops=10231343.29
Column=166824 Fraction=0.405 Mflops=10233013.47
Column=168840 Fraction=0.410 Mflops=10233405.28
Column=171024 Fraction=0.415 Mflops=10231459.84
Column=173040 Fraction=0.420 Mflops=10232113.14
Column=175056 Fraction=0.425 Mflops=10232701.53
Column=177072 Fraction=0.430 Mflops=10231090.17
Column=179256 Fraction=0.435 Mflops=10231407.53
Column=181272 Fraction=0.440 Mflops=10232071.24
Column=183288 Fraction=0.445 Mflops=10230170.21
Column=185304 Fraction=0.450 Mflops=10230355.57
Column=187488 Fraction=0.455 Mflops=10230722.22
Column=189504 Fraction=0.460 Mflops=10228269.78
Column=191520 Fraction=0.465 Mflops=10229191.66
Column=193536 Fraction=0.470 Mflops=10229136.43
Column=195720 Fraction=0.475 Mflops=10227329.25
Column=197736 Fraction=0.480 Mflops=10227293.10
Column=199752 Fraction=0.485 Mflops=10226867.92
Column=201768 Fraction=0.490 Mflops=10224786.23
Column=203952 Fraction=0.495 Mflops=10224530.38
Column=212184 Fraction=0.515 Mflops=10220082.95
Column=220416 Fraction=0.535 Mflops=10215375.41
Column=228648 Fraction=0.555 Mflops=10211360.50

Column=236880 Fraction=0.575 Mflops=10205202.48
Column=245112 Fraction=0.595 Mflops=10200166.01
Column=253344 Fraction=0.615 Mflops=10194726.96
Column=261576 Fraction=0.635 Mflops=10187898.70
Column=269808 Fraction=0.655 Mflops=10181891.52
Column=278040 Fraction=0.675 Mflops=10176059.10
Column=286272 Fraction=0.695 Mflops=10168871.70
Column=327432 Fraction=0.795 Mflops=10131694.53
Column=368592 Fraction=0.895 Mflops=10087746.84
Column=409752 Fraction=0.995 Mflops=10057537.15

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01R2R4      411768   168    30    36            4629.73              1.005e+04
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0010743 ..... 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.

4.2 Тест производительности кластера под Windows

Общие сведения о Linpack

Эталоном библиотеки для тестирования производительности суперкомпьютеров (не только кластеров) ещё с 80-х годов считается библиотека Linpack, сейчас расширенная до более функциональной LAPACK (Linear Algebra PACKage). Имеет интерфейсы для Fortran и C.

Аналоги LAPACK:

- Intel® MKL
- AMD™ ACML

- Sun Performance Library
- NAG’s LAPACK
- HP’s MLIB

Основная задача Linpack и его аналогов/модификаций состоит в решении системы линейных арифметических уравнений вида Ax=f методом LU-факторизации с выбором ведущего элемента столбца, где A — заполненная матрица размерности N. Исходная матрица разделяется на логические блоки размерностью NB × NB. Эти блоки в свою очередь разбиваются сеткой P × Q на более мелкие. Каждый из таких блоков «достанется» отдельному процессору системы.

Производительность в тесте Linpack измеряется в количестве производимых операций с плавающей запятой в секунду. Единицей измерения является 1 флопс (одна такая операция в секунду).

Основные параметры Linpack

N, ранг матрицы. Чем больше ранг, тем больше арифметических операций с плавающей запятой будет исполнено. N ограничен объёмом памяти, который система может выделить на процесс HPL. LIZARD может сам подбирать оптимальные, как он считает, параметры. Так, 26,000 подходит для четырёх узлов с 2 Гб ОЗУ на каждом. Но лучше-таки подбирать значение эмпирически, начиная с самого мелкого. Падение производительности обнаружится тогда, когда система начнёт писать в файл подкачки, и, соответственно, надо будет немного уменьшить значение ранга, чтобы получить оптимальное. N должно быть равно или больше P*Q.

P и Q – дополнительные коэффициенты, произведение которых необходимо подогнать к значению N. P*Q=КоличествоПроцессов. Можно приравнять P к количеству процессоров, а Q к количеству узлов – будет достаточно оптимально. Перед настройкой нужно учитывать Hyperthreading (а лучше вообще отключить).

NB – коэффициент, отражающий количество частей, на которые будет разбиваться задача. Показывает, какого объёма кусок данных будет получен каждым узлом. Таким образом, чем меньше значение данного коэффициента, тем оптимальнее загрузка процессора. Но можно настраивать так, как считается нужным, и смотреть производительность, которая получится в итоге (исходя из нужд архитектуры). При делении N на NB остаток должен быть равен нулю.

Для удобства можно воспользоваться Excel Linpack, при заполнении соответствующих ячеек самостоятельно высчитывающим значения коэффициентов.

HPL сохраняет результаты в файл hpl в своей рабочей папке с подробными комментариями. К сожалению, мне так и не удалось привести такой файл от нашей конфигурации в удобоваримый вид.

Lizard. Реализация Linpack для Windows-систем

Для Windows-систем специально была разработана оболочка для тестирования производительности кластера (Lizard, Linpack Wizard), имеющая в своей основе каноническую библиотеку, обёрнутую в удобный визуальный мастер (поставляется с HPC Tool Pack 2008). Позволяет данный мастер как экспресс-тест (со стандартными параметрами, автоматически

выбираемыми мастером), так и advanced для специфических настроек коэффициентов. Сопровождается всё комментариями.

Lizard. Оптимизация Linpack для Windows-систем

Microsoft для оптимизации рекомендует отключить все сервисы, от которых напрямую не зависит работа системы. Скрипт:

```
sc stop wuauserv
sc stop WinRM
sc stop WinHttpAutoProxySvc
sc stop W32Time
sc stop SstpSvc
sc stop Spooler
sc stop ShellHWDetection
sc stop RemoteRegistry
sc stop NlaSvc
sc stop NetTcpActivator
sc stop NetTcpPortSharing
sc stop netprofm
sc stop NetPipeActivator
sc stop MSDTC
sc stop KtmRm
sc stop KeyIso
```

```
rem sc stop gpsvc
sc stop bfe
sc stop CryptSvc
sc stop BITS
sc stop AudioSrv
sc stop SharedAccess
sc config WinRM start= disabled
sc config WinHttpAutoProxySvc start= disabled
sc config WAS start= disabled
sc config W32Time start= disabled
sc config TrkWks start= disabled
sc config Spooler start= disabled
sc config ShellHWDetection start= disabled
sc config RemoteRegistry start= disabled
sc config RasMan start= disabled
sc config NlaSvc start= disabled
sc config NetTcpActivator start= disabled
sc config NetTcpPortSharing start= disabled
sc config netprofm start= disabled
sc config NetPipeActivator start= disabled
sc config MSDTC start= disabled
sc config KtmRm start= disabled
```

sc config KeyIso start= disabled

sc config bfe start= disabled

sc config CryptSvc start= disabled

sc config BITS start= disabled

sc config AudioSrv start= disabled

sc config SharedAccess start= disabled

sc config SENS start= disabled

sc config EventSystem start= disabled

sc config PolicyAgent start= disabled

sc config AeLookupSvc start= disabled

sc config WerSvc start= disabled

sc config hkmsvc start= disabled

sc config UmRdpService start= disabled

sc config MpsSvc start= disabled

Native-средства для тестирования кластера

Кроме Linpack и Lizard, в Windows HPC Server 2008 (а именно, в HPC Pack 2008) есть стандартные средства тестирования производительности кластера, такие, как:

- MPI Ping-Pong Lightweight Throughput (переброс пакетами между узлами)
- MPI Ping-Pong Quick Check (проверка латентности сети, пропускной

способности etc)

5 Алгоритм-инструкция эффективного использования параллельных вычислений с применением имеющегося и вновь приобретаемого программного обеспечения

Разное программное обеспечение предназначенное для выполнения в параллельном режиме предъявляет свои требования к аппаратной конфигурации вычислительной системы для наиболее эффективного её использования. Одно ПО требует максимально быстрые процессоры, другое большое количество оперативной памяти на вычислительных узлах, третье максимально быструю подсистему ввода/вывода и т.д. Создать универсальную вычислительную систему, подходящую для всех параллельных программ довольно сложно, поэтому эффективнее всего создавать системы с разнородными компонентами, например вычислительными узлами, а затем, путем их комбинирования создавать оптимальные конфигурации для конкретного прикладного программного обеспечения.

Далее предлагается алгоритм наиболее эффективного использования аппаратного обеспечения для параллельных вычислений с применением прикладного программного обеспечения, закупленного или вновь закупаемого в СГАУ.

На рисунке 1 представлена блок-схема алгоритма. Рассмотрим подробнее каждый блок.

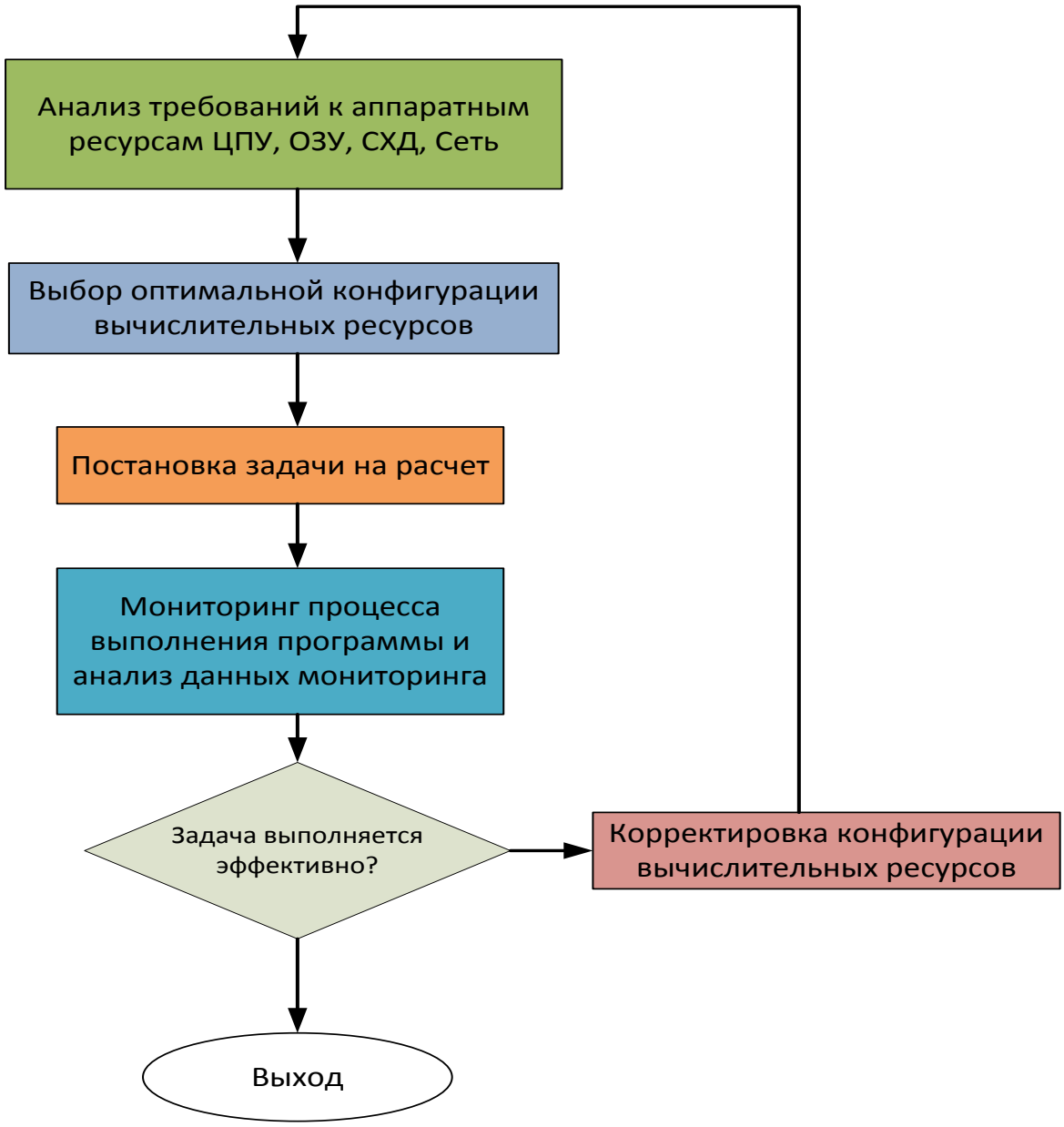


Рисунок 1 – Блок-схема алгоритма

Анализ требований к аппаратным ресурсам. На данном этапе необходимо проанализировать технические требования к аппаратному и системному программному обеспечению определенные разработчиком ПО. Эти требования могут быть получены из технической документации, которой сопровождается любое ПО. При недостатке информации можно запросить ее

конкретно у производителя с помощью систем обратной связи (вебсайт, форум, электронная почта и т.д.). Так же будет полезным анализ данных полученных от поисковых систем в сети Интернет на данное ПО.

Наибольшее внимание следует уделить требованиям к таким характеристикам вычислительной системы как:

- тип и производительность процессора (ЦПУ). Наличие поддержки тех или иных инструкций (SSE, AVX, SSE2, SSE4), оптимизаций и т.д.;
- тип и объем оперативной памяти (ОЗУ). Здесь как правило одно требование – чем больше ОЗУ тем лучше. Но опять же существуют определенные пределы при которых параллельная программа не может использовать больше памяти чем определено размерностью задачи, тогда слишком большой объем ОЗУ будет лишним;
- тип и производительность подсистемы хранения данных (СХД). Существуют параллельные программы которые используют дисковую подсистему вычислительной системы для хранения временных данных во время расчета, создавая при этом большую нагрузку на подсистему ввода/вывода;
- тип и характеристики коммуникационной сети для межпроцессного обмена данными. Здесь опять же необходимо знать насколько интенсивный обмен данными происходит во время выполнения параллельной программы и насколько велик объем данных пересылаемых за одну транзакцию. При этом предъявляются повышенные требования либо к величине задержек либо к пропускной способности сети.

После проведенного анализа требований ПО к аппаратным ресурсам производится выбор оптимальной конфигурации вычислительной системы.

Далее рассмотрим пример выбора определенных вычислительных узлов на суперкомпьютере «Сергей Королёв».

Суперкомпьютер состоит из двух управляющих и 136 вычислительных серверов. Управляющие сервера имеют локальные имена mgt1, mgt2, вычислительные узлы именуются: n1-n42, n57-n70, n113-n168, n225-n243, n246-n250

На каждом узле кластера, установлен жесткий диск для хранения временных данных, смонтированный в директорию /tmp. При завершении задачи на узле, эта директория очищается. На узлах n1-n34 доступно 56Гб., на n35-n42, n57-n70, n113-n168 - 120Гб, n239-n242, n246-n250 - 244Гб.

На узлах n239-n242, n246-n249 установлено по 96 Гб оперативной памяти. Эти узлы требуются для определенных задач, например ANSYS Mechanical. Для использования этих узлов, необходимо в файле задания при указании количества требуемых узлов, указывать параметр bigmem, например:

#PBS -l nodes=1:ppn=8:bigmem+15:ppn=8

при этом для задачи выделяются один узел с большим объемом памяти и 15 обычных узлов.

На узлах n225-n238, установлено по 24 Гб оперативной памяти. Для

использования в своих задачах, указывайте параметр ppn=12:mem в файле задания, т.к. эти ноды имеют по 12 ядер на узел, например:

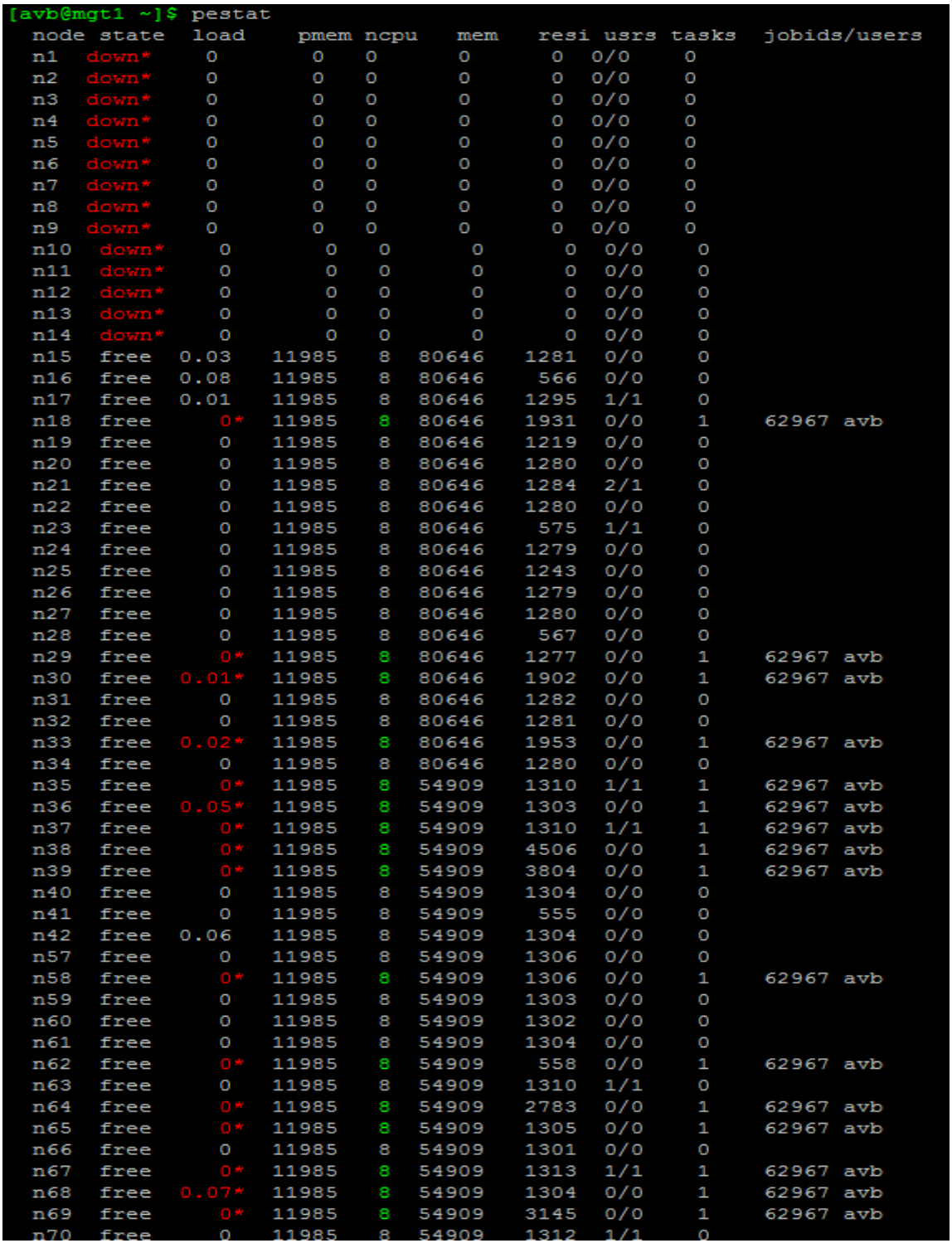
#PBS -l nodes=8:ppn=12:mem

На узлах n243 и n250 установлено по две графические карты Nvidia Tesla 2070. Для использования графических карт в своих заданиях, необходимо указывать в файле задания параметр узла gpu, например:

#PBS -l nodes=1:ppn=2:gpu

После выбора конфигурации производится постановка задачи на расчет. На суперкомпьютере «Сергей Королёв» это делается командой *qsub*.

Когда задача запустится необходимо провести мониторинг выполнения задания и анализ полученных с целью выявления проблем. Для этого можно воспользоваться либо системой мониторинга, доступной через веб-сервер по адресу http://sk.ssau.ru/ganglia, либо консольной командой *pestat.* На рисунке 2 приведен вывод команды *pestat*.



```
[avb@mgt1 ~]$ pestat
  node state  load    pmem ncpu   mem   resi usrs tasks  jobids/users
  n1   down*    0       0    0      0      0  0/0     0
  n2   down*    0       0    0      0      0  0/0     0
  n3   down*    0       0    0      0      0  0/0     0
  n4   down*    0       0    0      0      0  0/0     0
  n5   down*    0       0    0      0      0  0/0     0
  n6   down*    0       0    0      0      0  0/0     0
  n7   down*    0       0    0      0      0  0/0     0
  n8   down*    0       0    0      0      0  0/0     0
  n9   down*    0       0    0      0      0  0/0     0
  n10  down*    0       0    0      0      0  0/0     0
  n11  down*    0       0    0      0      0  0/0     0
  n12  down*    0       0    0      0      0  0/0     0
  n13  down*    0       0    0      0      0  0/0     0
  n14  down*    0       0    0      0      0  0/0     0
  n15  free  0.03   11985    8  80646   1281  0/0     0
  n16  free  0.08   11985    8  80646    566  0/0     0
  n17  free  0.01   11985    8  80646   1295  1/1     0
  n18  free     0*  11985    8  80646   1931  0/0     1     62967 avb
  n19  free     0   11985    8  80646   1219  0/0     0
  n20  free     0   11985    8  80646   1280  0/0     0
  n21  free     0   11985    8  80646   1284  2/1     0
  n22  free     0   11985    8  80646   1280  0/0     0
  n23  free     0   11985    8  80646    575  1/1     0
  n24  free     0   11985    8  80646   1279  0/0     0
  n25  free     0   11985    8  80646   1243  0/0     0
  n26  free     0   11985    8  80646   1279  0/0     0
  n27  free     0   11985    8  80646   1280  0/0     0
  n28  free     0   11985    8  80646    567  0/0     0
  n29  free     0*  11985    8  80646   1277  0/0     1     62967 avb
  n30  free  0.01*  11985    8  80646   1902  0/0     1     62967 avb
  n31  free     0   11985    8  80646   1282  0/0     0
  n32  free     0   11985    8  80646   1281  0/0     0
  n33  free  0.02*  11985    8  80646   1953  0/0     1     62967 avb
  n34  free     0   11985    8  80646   1280  0/0     0
  n35  free     0*  11985    8  54909   1310  1/1     1     62967 avb
  n36  free  0.05*  11985    8  54909   1303  0/0     1     62967 avb
  n37  free     0*  11985    8  54909   1310  1/1     1     62967 avb
  n38  free     0*  11985    8  54909   4506  0/0     1     62967 avb
  n39  free     0*  11985    8  54909   3804  0/0     1     62967 avb
  n40  free     0   11985    8  54909   1304  0/0     0
  n41  free     0   11985    8  54909    555  0/0     0
  n42  free  0.06   11985    8  54909   1304  0/0     0
  n57  free     0   11985    8  54909   1306  0/0     0
  n58  free     0*  11985    8  54909   1306  0/0     1     62967 avb
  n59  free     0   11985    8  54909   1303  0/0     0
  n60  free     0   11985    8  54909   1302  0/0     0
  n61  free     0   11985    8  54909   1304  0/0     0
  n62  free     0*  11985    8  54909    558  0/0     1     62967 avb
  n63  free     0   11985    8  54909   1310  1/1     0
  n64  free     0*  11985    8  54909   2783  0/0     1     62967 avb
  n65  free     0*  11985    8  54909   1305  0/0     1     62967 avb
  n66  free     0   11985    8  54909   1301  0/0     0
  n67  free     0*  11985    8  54909   1313  1/1     1     62967 avb
  n68  free  0.07*  11985    8  54909   1304  0/0     1     62967 avb
  n69  free     0*  11985    8  54909   3145  0/0     1     62967 avb
  n70  free     0   11985    8  54909   1312  1/1     0
```

Рисунок 2 – Вывод команды *pestat*.

В выводе команды *pestat* следует обратить внимание на следующие

столбцы:

- **load** – количество выполняющихся процессов прикладной программы на вычислительном узле;
- **ncpu** – количество физических ядер на вычислительном узле;
- **pmem** – объем физической оперативной памяти;
- **resi** – объем оперативной памяти используемой пользовательским приложением на узле.

Для достижения оптимальной эффективности использования суперкомпьютера необходимо соблюдение следующих условий.

1. Количество выполняющихся процессов не должно превышать количество физических ядер на узле. При нарушении этого условия происходит падение производительности вычислений, т.к. возникают накладные расходы на разделение процессорного времени между всеми выполняющимися процессами.
2. Объем используемой оперативной памяти прикладным приложением не должен превосходить объем физической оперативной памяти на узле. Несоблюдение этого условия ведет к тому что недостающий объем оперативной памяти компенсируется из так называемой виртуальной памяти расположенной на жестком диске вычислительного узла. Так скорость доступа к жесткому диску намного ниже скорости доступа к оперативной памяти, это ведет к резкому падению производительности и увеличению времени выполнения приложения.

По завершению этапа оценки эффективности выполнения параллельной программы необходимо скорректировать параметры вычислительной системы и произвести расчет снова.

Литература

1. Воеводин Вл. В. Суперкомпьютеры и парадоксы неэффективности. Открытые системы. 2009. № 10.-С. 37-45.
2. SPEC CPU2006 Documentation. [Электронный ресурс]. URL: http://www.spec.org/cpu2006/Docs (дата обращения 2.11.2013).
3. HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. [Электронный ресурс]. URL: http://www.netlib.org/benchmark/hpl (дата обращения 5.11.2013).
4. Суперкомпьютерный центр СГАУ | Суперкомпьютер "Сергей Королев". [Электронный ресурс]. URL: http://hpc.ssau.ru/node/6 (дата обращения 11.11.2013).