**ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ТЕОРЕТИКО-ИНФОРМАЦИОННОГО ПОДХОДА К ОЦЕНКЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМПЬЮТЕРОВ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

А. А. Ракитский

Б. Я. Рябко

Ст. преп. каф. ПМиК ФГОБУ ВПО «СибГУТИ»
rakitsky.anton@gmail.com

Ректор ФГОБУ ВПО «СибГУТИ»
boris@ryabko.net

Аннотация:

В работе [1] был предложен метод теоретической оценки производительности компьютеров. В данной работе рассматривается приложение этого метода для оценки реальных компьютеров и суперкомьютеров, описываются проблемы, с которыми при этом столкнулись авторы и как они были решены. Кроме того приводятся результаты сравнения предлагаемого метода с существующими методами оценки производительности, а также описывается способ применения метода в сфере облачных вычислений.

**Введение**

Разработка компьютеров является весьма трудоёмким процессом, состоящим из множества этапов. Одним из важнейших этапов является тестирование компьютера для определения его характеристик, в том числе и производительности. Для этого на стороннем компьютере с помощью специализированного программного обеспечения строится виртуальная модель компьютера. На этой модели проводится тестирование, проверяется работа элементарных инструкций, производится запуск объемных задач, состоящих из множества команд. Также отрабатывается взаимодействие различных блоков вычислительного модуля, производится оптимизация конфигурации и компьютер проходит полное тестирование, направленное на выявление ошибок, которые практически неизбежно возникают в работах такого уровня. Построение модели, как и её тестирование, является трудоёмким процессом, который увеличивает время разработки и себестоимость новых компьютеров. Для оценки производительности на этом этапе используются наборы тестовых задач, которые запускаются на модели компьютера, так называемые бенчмарки. Главным недостатком бенчмарков как способа оценки производительности является их субъективность. Т.к. набор задач ограничен, то каждый бенчмарк осуществляет оценку исключительно из способности компьютера решать этот конкретный набор задач. По этой же причине результаты, получаемые при применении разных бенчмарков, могут существенно отличаться, так как разные компьютеры могут быть лучше приспособлены для решения тех или иных конкретных задач. Для достижения большей объективности, разработчики бенчмарков стремятся максимально расширить список задач, что делает процесс оценки ещё более трудоёмким. Тем не менее, практически нигде не применяются теоретические методы оценки производительности компьютеров, хотя это могло бы существенно упростить и ускорить процесс разработки.

В работе [1] Рябко Б.Я. предлагает способ теоретической оценки производительности компьютеров, базирующийся на основных принципах теории информации, и для применения которого достаточно описания архитектуры компьютера. Для оценки производительности компьютеров используется характеристика вычислительная способность. Вычислительная способность – это оценка размера множества различных задач, которые могут быть решены компьютером за некоторое время. Для определения вычислительной способности необходимо и достаточно обладать информацией об архитектуре компьютера: список инструкций комьютера и время их выполнения, объемы различных видов памяти, используемых компьютером.

**Основные идеи и определения**

Представим некоторую упрощённую модель компьютера. Пусть компьютер описывается набором инструкций и доступной памятью . При этом под инструкцией компьютера мы будем подразумевать имя инструкции в совокупности со значениями её операндов. Таким образом, инструкции с одним и тем же именем, которые обращаются к различным ячейкам памяти, или регистрам, будет считаться различными, и обе будут входить во множество как самостоятельные инструкции.

Вычислительная задача определяется как последовательность инструкций , где . Важно отметить, что если вычислительная задача содержит цикл, который выполнится раз, то последовательность будет содержать тело цикла, повторённое соответственно раз. Очевидно, что не все последовательности инструкций могут быть допустимы. Вполне возможно, что некоторая последовательность инструкций приводит к критической ошибке в работе процессора, либо определёнными условиями запрещено последовательное совместное использование каких-либо пар инструкций.

Время выполнения инструкции обозначим как . Тогда время выполнения последовательности инструкций будет равно:

(1)

Число разных вычислительных задач, у которых время выполнения равно , эквивалентно размеру множества всех допустимых последовательностей инструкций, чьё время выполнения равно так же , т.е.

 (2)

где – количество различных вычислительных задач, у которых время выполнения равно , и

 (3)

Поэтому,

(4)

(Здесь и ниже – это целое число, и - это количество элементов, если множество, и длина, если - это слово).

**Определение 1**. Пусть есть компьютер с набором инструкций и пусть - время выполнения инструкции . Вычислительная способность определяется как

 (5)

**Утверждение 1.** Предел (5) существует, если – конечное множество, а времена выполнения инструкций - целые числа и их наибольший общий делитель равен 1. Доказательство утверждения 1 приводится в [1].

Простейший способ оценить вычислительную способность заключается в том, что мы предполагаем все последовательности инструкций допустимыми. Другими словами, рассмотрим набор инструкций как алфавит и предположим, что все слова (последовательности букв-инструкций) могут быть выполнены. Тогда возможно использовать метод вычисления пропускной способности канала без потерь, предложенный Шенноном[2]. Важно отметить, что этот метод является также верхней границей оценки вычислительной способности и для всех остальных моделей, потому что для любого компьютера множество допустимых последовательностей инструкций является подмножеством всех слов над «алфавитом» .

Рассмотрим компьютер с некоторым набором инструкций , время выполнения которых определено как , и все последовательности инструкций являются допустимыми. В этом случае оценить вычислительную способность (5) можно при помощи метода, предложенного Шенноном, который показал, что равна логарифму наибольшего целого решения следующего уравнения:

 (6)

где . Другими словами, .

Особенности оценки вычислительной способности для многопроцессорных систем и для многоядерных процессоров приводится в [1].

**Метод оценки вычислительной способности современных компьютеров**

При оценке вычислительной способности реальных современных компьютеров возникают сложности, которые не позволяют в чистом виде применить описанный выше метод. Архитектура современных компьютеров далеко не так проста, как предложенное описание. Современные процессоры являются сложными многоконвейерными системами с несколькими уровнями кэш-памяти. Рассмотрим отдельно конвейеры и кэш-память. Смысл использования кэш-памяти заключается в том, что она располагается непосредственно в процессоре и соответственно время обращения к этой памяти существенно меньше, чем время обращения к оперативной памяти. Опишем компьютер с кэш-памятью.

Допустим, имеется компьютер с набором инструкций , оперативной памятью , кэш-памятью 1го уровня и кэш-памятью 2го уровня , при этом время обращения к оперативной памяти и к кэш-памяти 1го и 2го уровней определяется соответственно как и . Рассмотрим особенности составления уравнения (6) для такого компьютера. При выполнении инструкции, которая обращается к каким-либо данным, компьютер в первую очередь ищет их в кэш-памяти 1го уровня. Если они там не были обнаружены, то поиск осуществляется уже в кэш-памяти 2го уровня, а если и там их не оказалось, то в оперативной памяти. Соответственно время выполнения инструкции в зависимости от расположения данных будет равно , если данные находятся в кэш-памяти 1го уровня, , если данные находятся в кэш-памяти 2го уровня и , если данные хранятся только в оперативной памяти.

Рассмотрим инструкцию , у которой только один операнд и он обращается к данным из памяти. Временем выполнения этой инструкции мы будем считать время обращения к данным плюс время обработки данных, которое мы обозначим как . Тогда, при построении уравнения, эта инструкция будет рассмотрена следующим образом:

1. Рассмотрим ситуацию, когда данные находятся в кэш-памяти 1го уровня. В этом случае мы добавим слагаемое , где – размер кэш-памяти 1го уровня. Инструкции, которые обращаются к различным ячейкам памяти, считаются различными, поэтому мы добавим инструкций с одинаковым временем исполнения, которые можно объединить в одно слагаемое.
2. Рассмотрим ситуацию, когда данные находятся в кэш-памяти 2го уровня. Тогда мы добавляем слагаемое , где – размер кэш-памяти 2го уровня.
3. Последняя ситуация, когда данные не были обнаружены в кэш-памяти и находятся в оперативной памяти. В таком случае мы добавляем слагаемое
 , где - размер оперативной памяти компьютера.

 При построении уравнения необходимо рассмотреть все три ситуации и поэтому инструкция будет представлена в уравнении в следующем виде:

Теперь необходимо рассмотреть такую неотъемлемую часть любого современного процессора, как конвейер. Принцип работы конвейера заключается в том, что он позволяет выполнить инструкцию в несколько этапов. Конвейер состоит из нескольких блоков, каждый из которых выполняет какую-то несложную задачу. Например, условные пять блоков: выборка инструкции, декодирование инструкции, декодирование адресов операндов инструкции, непосредственно выполнение инструкции и запись результата выполнения. На каждом из этих блоков инструкция выполняется довольно быстро. И такая структура позволяет нам существенно ускорить выполнение всех инструкций, т.к. когда инструкция освободила какой-то блок – на нём уже может начать выполняться следующая. Таким образом, если мы рассмотрим последовательность инструкций, то время выполнения каждой инструкции будет не суммой времён выполнения на каждом блоке, а максимальной задержкой, которая может возникнуть в конвейере из-за этой инструкции. Однако такая простая структура конвейеров актуальна лишь для ранних моделей процессоров. В более современных процессорах структура конвейера несколько сложнее.

Рассмотрим процессоры Intel Pentium Pro, построенные на основе архитектуры P6. В них каждая инструкция разбивается на микроинструкции, каждая из которых может быть выполнена по отдельности и время выполнения которых занимает единицу времени. Это позволяет распараллеливать выполнение этих микроопераций внутри блоков конвейера. Так, в рассматриваемой нами архитектуре каждый блок конвейера в единицу времени может выполнять 3 микрооперации, причём микрооперации могут на всех блоках выполняться полностью независимо, их порядок важен только в исполняющем блоке, которые кроме всего прочего состоит из 5 полностью параллельных исполняющих блоков. Каждый из этих блоков может выполнять микрооперации только определённого типа, например, выполнять арифметические операции или работать с данными в памяти. Тем не менее, благодаря переупорядочиванию микроопераций и грамотному их распределению, мы считаем, что пропускная способность блока в 3 микрооперации сохраняется. Если обобщить, то это означает, что процессор одновременно выполняет 3 потока микроопераций. При этом временем инструкции мы будем считать количество микроопераций, на которые она разбивается. Это следует из того факта, что для выполнения инструкции необходимо выполнение всех её микроопераций. Отдельно следует учесть задержку на работу с данными из памяти компьютера. Время, необходимое на доступ к данным, мы будем прибавлять как необходимую задержку на конвейере к каждой инструкции, у которой в качестве операнда есть ячейка памяти. Более подробно каждый процессор в отдельности был рассмотрен в работе [3].

**Применение вычислительной способности для оценки производительности реальных компьютеров**

Для применения метода был написан программный комплекс [4,5,6], который позволяет из условного описания списка инструкций и описания характеристик компьютера (тактовой частоты процессора, объемов всех видов памяти, времени доступа до каждого вида памяти) получить уравнение (6). Для проверки метода были рассмотрены основные современные процессоры Intel и некоторые процессоры AMD. Выбраны были те процессоры, для которых информация об архитектуре, списке инструкций и времени выполнения инструкций является открытой и легкодоступной. Для проверки метода, необходимо сравнить его с существующими методами оценки производительности компьютеров. В настоящее время известных методов теоретической оценки производительности не существует, поэтому для сравнения использовались результаты бенчмарков. Бенчмарк – это набор контрольных задач, необходимых для определения сравнительных характеристик производительности вычислительной системы. В контексте представляемого метода мы рассматриваем только те бенчмарки, которые оценивают производительность компьютеров.

Приведём график сравнения результатов оценки производительности компьютеров на базе процессоров Intel и AMD. На рис. 1 представлен график, где изображены 2 характеристики: вычислительная способность и PassMark. PassMark – это один из самых популярных в данный момент бенчмарков, при помощи которого составлена огромная база данных о производительности различных процессоров. Вторая характеристика – вычислительная способность. Из-за различий в величинах измерения характеристик график построен относительно первого компьютера (Pentium III), значение характеристик которого было взято за единицу. Все последующие значения сравниваются относительно него. Как можно заметить – характеристики ведут себя схожим образом и показывают одинаковые тенденции к изменению. Единственное более серьезное отклонение наблюдается только на компьютере Pentium IV, однако показатели бенчмарков для этого процессора с принципиально отличной от остальных архитектурой в то время вызывали много споров и теоретически этот процессор должен был существенно повышать производительность, что не показали некоторые из тестов (некоторые наоборот отразили существенный рост производительности). Это объясняется в первую очередь тем, что любой бенчмарк является все-таки конечным набором задач, который не может в полной мере оценить все возможности компьютера. На самом деле, результаты сравнения говорят о высокой точности предлагаемого метода и его отличной применимости на практике.



Рисунок 1. Сравнительная диаграмма процессоров семейств P6,P7,PM, Core 2 и AMD K10.

**Применение вычислительной способности для оценки производительности суперкомпьютеров**

Метод был также применён для оценки производительности суперкомпьютеров. Для получения оценки было произведено обобщение формулы для многопроцессорных компьютеров, с учётом межузлового взаимодействия. Пусть у нас есть суперкомпьютер с вычислительными узлами, при этом для каждого узла известно из каких процессоров он состоит, и какие объемы памяти на нём установлены. Кроме того нам известен протокол обмена данными между узлами. Тогда определим вычислительную способность узла как , где - суммарная вычислительная способностью всех процессоров узла с расчётом, что память узла является общей для всех процессоров, - вычислительная способность сетевого процессора. Сетевой процессор – это виртуальный процессор, который мы определяем для описания межузлового взаимодействия. В качестве инструкций этого процессора мы используем команды передачи/получения пакета данных с учётом скорости передачи и минимального и максимального размеров пакета. Таким образом, рассчитав вычислительную способность каждого узла, мы определим вычислительную способность суперкомпьютера как

 (7)

Для проверки описанного метода мы использовали данные проектов, которые занимаются исследованием и оценкой производительности существующих суперкомпьютеров (TOP500 и HPCC).



Рисунок 2. Сравнение вычислительной способности с результатами проекта TOP500.

На рис. 2 Rmax – это значение бенчмарка LINPACK (который оценивает способность суперкомпьютера решать СЛАУ), Rpeak – пиковое значение производительности суперкомпьютера, представляемое производителем, Computer Capacity –вычислительная способность. Как видно, предлагаемый метод в некоторых позициях расходится с обоими графиками, причем на последнем суперкомпьютере показывает результат ниже чем у обеих характеристик проекта. Это объясняется тем, что вычислительная способность оценивает возможность суперкомпьютера решать все возможные задачи, а предлагаемые характеристики рассматривают только конкретный тип задач. Ещё лучше это видно на следующем графике (рис. 3), где вычислительная способность сравнивается с бенчмарком LINPACK(G-HPL), бенчмарком G-FFT, реализующим алгоритм быстрого преобразования Фурье, и теоретическим максимумом (предоставляемым производителем). На этом графике хорошо видно, что характеристика показывает усреднённое значение относительно показателей бенчмарков, что ещё раз подтверждает универсальность и применимость метода. Более подробно результаты и методы оценки производительности суперкомпьютеров рассмотрены в [7].



Рисунок 3. Сравнение вычислительной способности с результатами проекта HPCC.

Метод по аналогии с суперкомпьютерами может быть применён и в сфере облачных вычислений. Облачные вычисления – это модель, которая обеспечивает повсеместный и удобный сетевой доступ к конфигурируемым вычислительным ресурсам (серверы, устройства хранения данных, сети передачи данных, виртуальные распределённые вычислительные системы). Основными потребителями облачных ресурсов являются компании, которым по тем или иным причинам невыгодно приобретать собственный комплекс серверов, создавать инфраструктуру для хранения больших массивов данных или приобретать суперкомпьютер для выполнения трудоемких вычислительных задач.

В контексте данной работы особенно интересна последняя область. Многим компаниям (научным центрам, небольшим университетам) большие вычислительные мощности требуются крайне редко, для решения частных и редких трудоёмких задач, на подготовку которых уходит больше времени, чем на их непосредственное решение с помощью высокопроизводительных систем. Вполне естественно, что более выгодным вариантом для таких компаний будет временное приобретение в аренду вычислительных мощностей, которые позволят решить поставленную задачу, чем покупка, установка и обслуживание собственного суперкомпьютера со всей инфраструктурой.

Наиболее распространённым открытым программным обеспечением для управления облачными вычислительными центрами обработки данных (ЦОД) является OpenStack. Это открытое программное обеспесчение, полностью документированное, с открытым исходным кодом, а, следовательно, с возможностью его модификации в зависимости от особенностей архитектуры ЦОД. Именно эти факторы обуславливают несомненную популярность OpenStack.

Это программное обеспечение позволяет создавать многоарендные архитектуры IaaS (Infrastructure as a Service) облачных вычислительных центров, когда вычислительные и сетевые ресурсы необходимо распределить между несколькими пользователями-арендаторами. Каждому арендатору предоставляются виртуализированные ресурсы в виде множества виртуальных машин, связанных виртуальной сетью. Пользователь может сам настраивать конфигурацию виртуальных машин, в том числе параметры процессора, памяти и всего программного наполнения.

Именно такой подход к предоставлению ресурсов пользователю и открывает возможность для применения метода. Как мы уже показали выше, метод аналитической оценки вычислительной способности может быть применён при проектировании высокопроизводительных распределённых вычислительных систем. Нетрудно заметить схожие моменты в настройке конфигурации облачной распределённой вычислительной системы и обычном высокопроизводительном кластере, например в подборе и настройке параметров процессора. Благодаря описанному в работе методу пользователь может выбрать оптимальную конфигурацию системы, которая будет решать требуемые задачи и при этом иметь минимальную стоимость.

Происходить это может следующим способом. Пользователь определяет, какая производительность системы будет достаточна для решения его задачи, после чего, при помощи метода определяет, при каких конфигурациях будет достигаться требуемая производительность и выбирает из них минимальную по стоимости. Аналогичным образом возможно и обратное, пользователь подбирает конфигурацию вычислительной системы с максимальной производительностью исходя из суммы, которую готов потратить.

**ЛИТЕРАТУРА**

1. **Ryabko B.** An information-theoretic approach to estimate the capacity of processing units // Performance Evaluation. 2012. V. 69, P. 267-273.
2. **Shannon C. E.** A mathematical theory of communication // Bell Sys. Tech. J. 1948. V. 27, P. 379-423, P. 623-656.
3. **Ракитский А.А.** Теоретическая оценка вычислительной способности процессоров Intel // Вестник СибГУТИ. 2013.3 с. 29-45.
4. **Ракитский А.А., Рябко Б.Я., Фионов А.Н.** Программа нахождения вычислительной способности по характеристическому уравнению. № гос. регистрации 2013619017.
5. **Ракитский А.А.** Программа для преобразования списка инструкций компьютера в характеристическое уравнение. № гос. регистрации 2013619015.
6. **Ракитский А.А.** Программа для построения списка инструкций компьютера в требуемом формате представления. № гос. регистрации 2013619014.
7. **Ракитский А.А.** Использование вычислительной способности как характеристики для оценки и сравнения суперкомпьютеров // Вестник СибГУТИ. 2013.4 с. 67-84.