Разработка методов построения рейтингов вычислительных систем, основанных на реализациях различных алгоритмов

А.А. Желтков

Московский государственный университет имени М.В. Ломоносова

В работе излагается подход к исследованию свойств реализаций различных алгоритмов при их запуске на разных вычислительных платформах. Рассматривается обобщение основных принципов построения известных суперкомпьютерных рейтингов и их применение для сравнения и анализа характеристик выполнения произвольных реализаций алгоритмов на разных вычислительных системах. Выделяется набор характеристик вычислительных систем, алгоритмов и их реализаций, которые представляются на данный момент наиболее существенными и востребованными для анализа. Предлагается концепция общедоступного инструмента для сравнения и анализа свойств реализаций алгоритмов и вычислительных систем, и описываются основные варианты его использования.

Ключевые слова: высокопроизводительные вычисления, суперкомпьютерные рейтинги, реализации алгоритмов, характеристики вычислительных систем, свойства алгоритмов.

1. Введение

Рейтинги суперкомпьютеров – одна из важнейших составляющих в области высокопроизводительных вычислений. Они способны показывать текущие тенденции в данной области и дают почву для анализа производительности и масштабируемости суперкомпьютерных вычислений.

На сегодняшний день широко известны такие суперкомпьютерные рейтинги, как TOP500, Graph500, Green500, GreenGraph500, HPCG, HPGMG. Они отличаются друг от друга тем, что основываются на различных алгоритмах и/или метриках, по которым сравниваются участвующие в них вычислительные системы. Так, например, рейтинг TOP500 [1] для сравнения систем использует бенчмарк LINPACK [2], реализующий алгоритм решения СЛАУ с плотной матрицей; рейтинг Graph500 [3], в соответствии с названием, сравнивает производительность систем на графовых алгоритмах. Рейтинги Graph500 и GreenGraph500 сравнивают удельную энергоэффективность систем на 1 Ватт потребляемой энергии при прогоне соответствующих тестам TOP500 и Graph500 бенчмарков. Каждый из этих рейтингов оценивает системы только в рамках конкретно заданных бенчмарков и метрик и дает ответ на вопрос «какая система лучше подходит для данной задачи?», а не «какая система наиболее производительная/эффективная вообще?».

Понимание того, что популярные рейтинги, такие как TOP500 и Graph500 далеко не всегда дают картину того, насколько производительны/эффективны будут те или иные вычислительные системы на реальных вычислительных приложениях привели к появлению рейтингов HPCG и HPGMG. Тем не менее, количество востребованных вычислительных задач и алгоритмов существенно превышает количество доступных рейтингов. Также, каждый из рейтингов имеет свою структуру хранения данных и их представление, поэтому получение сводной информации по нескольким рейтингам зачастую является затруднительным действием. Кроме того, большое количество прикладных вычислений выполняется не на суперкомпьютерах и сверхбольших системах, а на кластерах на порядки меньшей производительности, которые в силу этого не могут претендовать на попадание в такие рейтинги.

Это приводит к идее создания инструмента, который мог бы, с одной стороны, служить единым хранилищем описаний вычислительных систем и данных о производительности этих систем на различных бенчмарках (то есть различных классах задач и алгоритмов), а с другой стороны давал бы возможность оперировать данными не только по сверхбольшим системам, но и мог служить источником данных по системам любых масштабов. И, что является одним из наиболее

ценных свойств, он не ограничивается набором задач и алгоритмов, соответствующих популярным рейтингам, а позволяет добавлять результаты прогонов реализаций любых интересующих алгоритмов. Также, результаты прогонов не обязаны ограничиваться одной метрикой (как, например, FLOP/s в рейтинге TOP500), а предполагают наличие множества характеристик, по каждой из которых затем можно будет сравнивать эти прогоны между собой.

Данный инструмент на основании внесенных данных позволит строить рейтинги вычислительных систем как по конкретной реализации алгоритма (что позволит исследовать непосредственно его масштабируемость), так и по целому множеству реализаций алгоритмов, относящемуся к интересующему пользователя классу задач. Кроме того, имея такие данные, можно было бы сравнить производительность и эффективность различных алгоритмов и/или реализаций в рамках одной вычислительной системы.

Такой инструмент в перспективе может предоставить заинтересованным пользователям большие возможности как для исследования масштабируемости различных приложений на разных вычислительных платформах и архитектурах, так и для исследования самих вычислительных систем и их поведения при решении разным типов задач.

2. Описание сервиса построения рейтингов вычислительных систем, основанных на реализациях различных алгоритмов

Предполагается создание сервиса, являющегося единым источником информации о производительности/эффективности реализаций различных алгоритмов на различных вычислительных платформах. В качестве источника информации о вычислительных задачах, алгоритмах, реализациях и взаимосвязях между ними предполагается использовать открытую энциклопедию алгоритмов AlgoWiki [4].

Отдаленным частным аналогом сервиса может служить известный список Top500, являющийся рейтингом суперкомпьютеров, отражающим скорость решения больших систем линейных уравнений с плотной матрицей на основе теста High Performance Linpack [2]. Получаемые с помощью данного сервиса рейтинги вычислительных систем предоставляют данные о запусках произвольной реализации любого алгоритма из AlgoWiki, позволяя работать не только с максимальными значениями, характерными для рекордных рейтингов, но и со значениями произвольного диапазона.

Цель – накопление и последующий анализ данных, изучение влияния характеристик вычислительных систем, особенностей алгоритма, реализации и входных данных задачи на производительность вычислений (а также их прочие характеристики, например, энергоэффективность и т.д.).

2.1 Функциональные требования к описываемому сервису построения рейтингов

2.1.1 Тривиальные запросы к сервису

Ниже представлен список наиболее простых типовых запросов, которые ожидаются от пользователей сервиса, и которые требуется эффективно выполнять.

- Показать данные прогонов на фиксированной вычислительной платформе. Это значит, что фиксируется конкретная вычислительная система (например, суперкомпьютер Ломоносов-2), а все остальные параметры произвольны, т.е. интересуют данные по всем задачам/методам/алгоритмам/реализациям, выполненных именно на Ломоносове-2. При этом также фиксируется метрика (FLOPS/TEPS/Power/etc.), чтобы показывать данные по сравнимым прогонам. Такой запрос может быть нужен, например, с целью определения того приложения, которое максимально эффективно на Ломоносове-2.
- Показать данные прогонов заданного алгоритма на нескольких фиксированных платформах. Это значит, что зафиксирован конкретный алгоритм, а параметр, отвечающий вычислительной системе, может принимать только значения из задан-

ного набора (например, зафиксированы 2 допустимых значения – суперкомпьютеры Ломоносов-2 и Ломоносов-1). Остальные параметры при этом произвольны, но, опять же, нужно зафиксировать метрику, чтобы показывать данные по сравнимым прогонам. В данном случае решается задача демонстрации производительности/эффективности самых разных реализаций данного алгоритма на этих двух платформах.

- Показать данные прогонов по всем алгоритмам, решающим данную задачу. Это значит, что фиксируется только лишь задача, и пользователю выводятся данные прогонов по всем реализациям алгоритмов, решающих (участвующих в решении) заданной задачи. При этом также необходимо учитывать, что прогоны должны быть сравнимыми (с одной и той же метрикой).
- Показать данные прогонов по всем вычислительным системам с заданными компонентами / характеристиками. Это значит, что будут зафиксированы некоторые технологические детали описания систем (они могут быть самыми разными – тип процессора, частота процессора, размер кэша, число ядер в процессоре, топология интерконнекта, тип параллельной файловой системы, тип СХД (системы хранения данных) и т.п.). Исходя из них выбираются все системы, подходящие под эти условия и выводятся данные по прогонам именно на них. Такие запросы могут быть нужны для анализа влияния конкретных компонент вычислительных систем на производительность вычислений (или иные характеристики выполнения реализаций алгоритмов на вычислительных платформах).

2.1.2 Базовые выборки, предоставляемые сервисом

Помимо тривиальных запросов, которые тем не менее могут быть нужны пользователям, для обеспечения удобного взаимодействия пользователя с сервисом полезно иметь набор предварительно отобранных выборок данных, которые с большой вероятностью могут быть востребованы большим количеством пользователей.

Это будет способствовать тому, что:

1) пользователь в ряде случаев сэкономит время на навигации по сервису и быстрее получит желаемые данные.

2) пользователь на примере предоставленных выборок сможет лучше ознакомиться с функциональностью сервиса и его доступными возможностями.

Под базовыми выборками понимаются два типа сущностей:

1) Готовые срезы данных по прогонам по наиболее востребованным запросам. При этом на основании этих срезов пользователю можно будет строить более специфичные, уточненные выборки, добавляя необходимые характеристики в итоговый вывод, а также дополнительно фиксируя еще некоторые параметры.

2) Предзаполненные «конструкторы» запросов данных о прогонах, где будут предложены наиболее востребованные (популярные) для анализа параметры, при заполнении которых конкретными значениями (пошаговом) пользователь на выходе получит срез данных по прогонам с интересующими его параметрами. Который при этом далее можно будет еще дополнить и уточнить.

В качестве готовых срезов данных предлагаются следующие выборки:

1) Выборка-аналог TOP500. То есть, зафиксирована классическая реализация алгоритма, использующаяся в тесте Linpack / все реализации, удовлетворяющие требованиям подачи в TOP500 (LU-разложение матрицы, вычисления с двойной точностью, сложность – $\frac{2}{3}n^3 + O(n^2)$ [1]). Результаты выдаются по всем прогонам этих реализаций, при этом зафиксирована метрика = FLOPS.

2) Linpack Benchmark SP (Single Precision) – то же самое, что и предыдущий вариант, но для вычислений с одинарной точностью, а не с двойной.

3) Выборка-аналог Green500. То же самое, что и в п.1, но в качестве метрики будем фиксировать энергоэффективность прогонов (соотношение FLOPS/W).

4) Выборка-аналог Graph500 (или GreenGraph500). Алгоритм – BFS (Breadth-first search), метрика – TEPS (соответственно, TEPS/W).

Для «конструкторов» имеет смысл очертить список значимых параметров, которые с большой вероятностью были бы востребованы для анализа и поэтому нужны в них. При этом стоит учитывать, что некоторые параметры допускают и требуют фиксацию только одного конкретного значения (например, метрика результата прогона – она может быть задана только одна: время выполнения, производительность с заданной единицей измерения, эффективность, энергопотребление системы), а другие параметры допускают выбор нескольких значений (что подразумевает объединение результатов по каждому из них). В качестве значимых параметров рассматриваются следующие показатели, относящиеся к самим системам, исполняемым реализациям и непосредственно прогонам (запускам программных реализаций):

I. Характеристики систем
   1. Параметры CPU: тактовая частота (диапазон), количество ядер (диапазон), размеры кэша (диапазон), техпроцесс (диапазон), производитель (возможен множественный выбор) ИЛИ конкретная модель CPU (множественный выбор)
   2. Параметры узла: наличие/количество CPU (диапазон), наличие/количество ускорителей (диапазон), объем ОЗУ на узле (диапазон), наличие особого интерконнекта между CPU/ускорителями
   3. Параметры уровня системы: количество узлов (диапазон), интерконнект (семейство или пропускная способность), операционная система (множественный выбор), объем СХД (диапазон), тип СХД (множественный выбор)
   4. Параметры ускорителей: тип (множественный выбор), тактовая частота (диапазон), количество ядер (диапазон), размеры кэша (диапазон), техпроцесс (диапазон), производитель (возможен множественный выбор) ИЛИ конкретная модель ускорителя (множественный выбор)

II. Характеристики реализаций
   1. Тип алгоритма, которому должны соответствовать выбираемые реализации (графовые алгоритмы, алгоритмы линейной алгебры, …) – возможен множественный выбор

III. Характеристики прогонов
   1. «Весовая категория» входных данных, то есть градация по диапазонам размеров задачи (сверхбольшие, большие, средние, малые, также можно взять за основу именование из Graph500)
   2. «Весовая категория» результатов, то есть градация по диапазонам значений метрик результата прогона (главным образом, производительности). Так, к примеру, можно будет отделить суперкомпьютеры, просто большие кластеры и, например, обычные ПК, а также мобильные устройства.
   3. Дата проведения прогона (диапазон)

При фиксации нескольких из параметров, предлагаемых в конструкторе, пользователь получает возможность получить соответствующую им выборку данных (которая представляет собой рейтинг прогонов программных реализаций алгоритмов, основанный на выбранной пользователем метрике). Полученную выборку пользователь в дальнейшем может скорректировать, добавляя и задавая значения нужным дополнительным параметрам и/или уточняя набор значений ранее заданных параметров.

Также надо отметить, что параметры могут быть несовместимы между собой, или быть формально совместимыми, но совмещение которых при этом в данный момент не представляется полезным и информативным (что, однако, не мешает вернуться к ним и пересмотреть полезность их совмещения в будущем). В качестве примера заведомо несовместимых параметров можно привести название модели центрального процессора и количество CPU-ядер (поскольку оно однозначно определяется моделью) или название модели интерконнекта и его пропускная способность.

2.2 Предоставление данных в описываемый сервис построения рейтингов

Источником данных для построения рейтингов по ним, как и в «больших» рейтингах, упомянутых ранее (TOP500, Green500 и т.д.), должны быть пользователи. Предполагается, что сервис будет являться открытым для подачи заявок по результатам прогонов любых известных реализаций алгоритмов на произвольных вычислительных платформах. Такой подход влечет за собой необходимость верификации предоставляемых данных, что является отдельной задачей.

При представлении результатов запуска, пользователи сопровождают их информацией о запуске (часть ее обязательна, часть опциональна), которую можно разбить на следующие группы:

- достигнутое значение производительности (в общем случае по нескольким метрикам) – обязательно (хотя бы по одной метрике). Примеры:
  - основная метрика (FLOPS, TEPS, etc.)
  - время выполнения
  - энергоэффективность (FLOPS/W) или просто потребленная мощность
- конфигурация компьютера, на которой это значение производительности достигнуто (выбор уже существующего описания или добавление нового) – обязательно. Соответствует описанию вычислительной системы в обновленном рейтинге Top50 [5] (описывается состав компьютера по узлам, узлы по компонентам (CPU, RAM, ускоритель). Также сюда входят:
  - количество узлов ВС, непосредственно на которых производился запуск – обязательно
  - тип (конфигурация) узлов, непосредственно на которых производился запуск – обязательно
- характеристика входных данных, на котором это значение производительности достигнуто (обобщенный размер в своей единице измерения), а также дополнительные атрибуты:
  - размер матрицы (масштаб графа и т.п.) – обязательно
  - тип графа (матрицы, и т.п.)
- атрибуты запуска, не связанные с входными данными (например, процессорная матрица в HPL, параметры алгоритма: размер блока и т.д.)
- исходный код, на котором это значение производительности достигнуто – обязательно
- описание компилятора и опций, используемых библиотек, на которых это значение производительности достигнуто (Makefile). При этом отдельно запрашиваются такие параметры, как:
  - версия компилятора
  - флаги компилятора
- команда запуска, соответствующая полученному результату прогона

Предъявленные функциональные требования к сервису вкупе с требованиями к предоставляемым данным по мере наполнения сервиса данными прогонов позволят проводить качественный анализ производительности и масштабируемости различных приложений на различных вычислительных архитектурах, и получать представление об эффективности использования систем (или типов систем) для решения интересующих классов задач.

2.3 Описание архитектуры вычислительных систем в рамках сервиса

Критически важной составляющей, необходимой для информативности и полноценного функционирования предлагаемого в данной работе сервиса, является детальность и точность описания архитектуры вычислительных систем. Общие сведения о системах, которые можно наблюдать в больших рейтингах (большинство из них ограничиваются предоставлением данных о общем количестве узлов, количестве оперативной памяти и текстовым описанием процессора, ускорителей и конфигурации узла) хороши в «первом приближении», для ознакомления человека с отдельными примерами того, какие системы существуют и доминируют в области высокопроизводительных вычислений с точки зрения разных их приложений. Но когда встает задача предметного анализа влияния различных характеристик и компонент систем на поведение про-

изводительности и эффективности тех или иных реализаций алгоритмов, и количество исследуемых систем не ограничено десятками, а много больше (и, следовательно, изучение требует некоторой автоматизированной обработки) – для этих целей описания систем должны быть значительно более детализированы и, что важно, структурированы с тем, чтобы разные характеристики были отделимы друг от друга и выборку систем можно было строить в разрезе каждого отдельно взятого параметра (здесь имеется в виду, что неструктурированное текстовое описание центрального процессора, где перечислены все его спецификации, такие как частота, размеры кэшей, техпроцесс, технология векторизации и т.п., пусть и обладающее максимальной детальностью, в данном случае не годится, поскольку не дает возможности реализовать автоматизированное построение выборки, где была бы зафиксирована, например, конкретная тактовая частота процессора).

Поэтому одним из основных требований к программной реализации такого сервиса является возможность хранить и обрабатывать максимально полные и гранулярные описания систем (но в разумных пределах, с учетом специфики задачи анализа влияния характеристик и компонент вычислительных систем на показатели исполнения реализаций различных алгоритмов) и с учетом того, что со временем могут появляться новые важные характеристики систем и их компонент, которые нужно будет также учитывать и обрабатывать.

Для этих целей предлагается модель описания вычислительных систем, основанная на описании систем в обновленном рейтинге Top50 суперкомпьютеров России и СНГ [5], но со значительно более расширенным набором характеристик (атрибутов) компонент (таких, как CPU и ускорители, а также конфигурация узла вообще), а также уточненной иерархией компонент систем для более детального отражения особенностей архитектуры.

В соответствии с данной моделью, различные компоненты систем (такие, как узлы, процессоры, ускорители) и сами системы, как и раньше, представляются отдельными сущностями (объектами), которые могут иметь связи между собой и которым может быть приписан произвольный набор значений других сущностей (атрибутов). Но в уточненной модели для описания характеристик процессора теперь предлагается следующий набор атрибутов:

- Модель CPU
- Семейство CPU
- Количество ядер процессора
- Наличие технологии hyperthreading
- Базовая тактовая частота, МГц
- Максимальная тактовая частота (Turbo Boost), МГц
- Размер кэша L1, КБ
- Размер кэша L2, КБ
- Размер кэша L3, КБ
- Размер кэша L4, МБ
- Количество уровней кэш-памяти
- Используемая технология векторизации
- Ширина векторных регистров, бит
- Наличие DMA (Direct memory access)
- Техпроцесс, нм
- Средняя мощность, Вт
- Частота системной шины, GT/s
- Макс. число каналов памяти
- Макс. пропускная способность памяти, ГБ/с

Набор характеристик ускорителя расширился по аналогии, с учетом специфики графических и иных ускорителей:

- Модель ускорителя
- Архитектура GPU
- Количество SM/SMX или векторных ядер
- Количество GPU-ядер

- Количество тензорных ядер
- Тактовая частота, МГц
- Средняя мощность, Вт
- Техпроцесс, нм
- Частота памяти, МГц
- Тип памяти GPU (GDDR5 / HBM / …)
- Размер кэша L1, КБ
- Размер кэша L2, КБ
- Количество уровней памяти ускорителя
- Объем памяти ускорителя, GB
- Макс. пропускная способность памяти, GB/s

При этом в новой модели отдельно описывается интерконнект системы как между узлами, так и внутри узла (между компонентами, расположенными на одном узле). Интерконнект также обладает своим базовым набором атрибутов:

- Модель
- Семейство (InfiniBand / Ethernet / …)
- Количество каналов
- Топология
- Пропускная способность
- Задержка сигнала (latency)

Кроме того, в новой модели описания отдельно может быть описана система хранения данных вместе со своими характеристиками и предусмотрена возможность наличия нескольких типов памяти на узле (включая энергонезависимую память и локальный диск).

Приведенные уточнения описаний систем дают возможность производить существенно более глубокий анализ влияния особенностей систем на производительность вычислений и позволяют более точно и наглядно отразить специфику архитектуры различных систем.

3. Заключение

Построение рейтингов производительности вычислительных систем по фиксированным техническим параметрам в рамках заданных реализаций алгоритмов на сегодняшний видится полезной задачей, которая могла бы помочь оценить вклад различных характеристик систем и успешность их совмещения для решения тех или иных прикладных задач.

Возможность проведения всестороннего анализа результатов прогонов реализаций типовых алгоритмов на различных вычислительных системах при условии наличия их детального описания позволяет выделить ключевые параметры и компоненты систем с точки зрения того или иного алгоритма, а также дает почву для построения прогнозов производительности и эффективности существующих или планируемых вычислительных систем исходя из их спецификаций.

Все это дает основания полагать, что предлагаемый в работе сервис построения выборок (рейтингов) различных систем может быть крайне востребован в области высокопроизводительных вычислений, как и в области производства вычислительного оборудования.

Литература

1. TOP500 Supercomputer Sites. http://www.top500.org.
2. Dongarra, J.J., Luszczek, P., Petitet, A.: The LINPACK Benchmark: past, present and future. Concurrency Computat.: Pract. Exper. 15 (2003) 803–820.
3. Graph 500. http://www.graph500.org.

4. Antonov A. et al. Hierarchical Domain Representation in the AlgoWiki Encyclopedia: From Problems to Implementations //International Conference on Parallel Computational Technologies. – Springer, Cham, 2018. – С. 3-15.
5. Nikitenko D., Zheltkov A. The Top50 list vivification in the evolution of HPC rankings //International Conference on Parallel Computational Technologies. – Springer, Cham, 2017. – С. 14-26.