

## Проблемы оценки эффективности распределенных и многопроцессорных систем различных классов для задач обработки изображений

*А.И. Костюк, В.Н. Пуховский, А.В. Горбунов, Н.М. Коробейникова*

*Южный федеральный университет, Ростов-на-Дону*

**Аннотация:** В статье рассматривается сравнительный анализ использования распределенных и многопроцессорных систем различных классов для задач обработки изображений. Приводятся обозначения и классификации вычислений, характер часто применяемых операций для обработки изображений, классификация архитектур для обработки изображений. Исследуются вопросы соответствия общих классов вычислительных архитектур типам вычислительных операций, используемых в процессе обработки изображений. Показано, что каждая архитектура имеет преимущества и недостатки в некоторых специальных областях обработки, но ни одна из них не является оптимальной для всех случаев.

**Ключевые слова:** алгоритмы обработки изображений, распределенные системы, многопроцессорные вычислительные системы, анализ изображений, классификация архитектур.

### Введение

Во всех задачах, связанных с изображениями, за исключением, пожалуй, лишь самых простых, для обработки потока данных необходимо их параллельная обработка несколькими мощными процессорами. Для решения подобных задач требуется создание, как правило, мощных вычислительных ресурсов специального назначения, так называемых графических станций, обладающих повышенными значениями разрешающей способности, цветовой гаммы, скорости регенерации кадров изображений [1]. По мере роста требований к этим характеристикам возрастают требования к вычислительной мощности, пропускной способности и памяти. Однако создание графических станций из-за большой стоимости ограничивает возможность их широкого распространения, поэтому необходимы также и другие подходы, обеспечивающие близкие возможности, но при меньших затратах. Кроме того, для решения многих практически важных задач

---

требуются достаточно традиционные средства машинной графики, которыми привыкли пользоваться программисты традиционных компьютерных систем, и которые часто невозможно использовать в специализированных графических системах из-за особенностей их архитектуры и специфического программного обеспечения.

### **Материал и методы исследования**

Целью исследования в данной статье является сравнительный анализ использования распределенных и многопроцессорных систем различных классов для задач обработки изображений. Как правило, большинство специалистов в области анализа изображений предпочитают использовать универсальные ЭВМ, разрабатывая на них собственное программное обеспечение. Преимуществом такого подхода является то, что в этих условиях несложно быстро вносить изменения в программы и отлаживать любой алгоритм с минимальными затратами времени. Однако эта легкость в создании программ анализа изображений приводит, как правило, к большому времени счета на ЭВМ – так совсем простые преобразования, требующие относительно небольших вычислений, занимают от нескольких минут до нескольких часов процессорного времени на лучших универсальных ЭВМ, что неприемлемо почти во всех серьезных приложениях. Поэтому значительно большее внимание следует уделять эффективности и скорости вычислений.

В связи с необходимостью передачи больших объемов информации, развитие получили системы со специализированными скоростными шинами или процессоры, имеющие встроенные высокопроизводительные каналы передачи информации, многопроцессорные и распределенные вычислительные системы. Недостатком использования как распределенных, так и многопроцессорных систем является их относительно большая стоимость, а также проблема коммутации, возникающая с увеличением

количества вычислительных узлов, используемых в такой системе. Для распределенных систем, размещаемых на облачных платформах, можно привести следующие характеристики приложений [2]:

- Работу можно разделить на дискретные задачи, которые будут одновременно выполняться в множестве ядер.
- Время выполнения каждой задачи ограничено. Принимаются входные данные, выполняется обработка, и выводятся результаты. Приложение в целом выполняется в течение конечного периода (от нескольких минут до нескольких дней). Такие задачи обычно решаются единовременным выделением большого количества ядер, число которых постепенно сводится к нулю по мере выполнения приложения.
- Такому приложению не обязательно работать круглосуточно и ежедневно. Но система должна быть готова к сбоям узлов или самого приложения.
- Задачи приложения могут быть независимыми и выполняться параллельно. Или же они могут быть тесно взаимосвязаны, то есть должны взаимодействовать в процессе выполнения или обмениваться промежуточными результатами.
- В зависимости от характера рабочей нагрузки могут потребоваться значительные размеры виртуальных машин для больших объемов вычислений.

Задача обработки (анализа) изображений в общем случае может быть разбита на ряд подзадач, причем каждой из них соответствует свой тип вычислений, представленный в таблице №1. В этой таблице каждому типу вычислительных операций поставлено в соответствие его условное обозначение.

Таблица № 1.

Обозначения классификации вычислений

Локальные / глобальные вычисления	a/b
Линейные / нелинейные вычисления	c/d
Интенсивное использование памяти / интенсивный счет	e/f
Объектно / координатно-ориентированные вычисления	g/h
Контекстно-свободные / зависимые вычисления	i/j
Символические вычисления / вычисления на элементах	k/l

Для обработки изображений особенно важно разграничить символьные вычисления и обработку на уровне элементов изображения. Принципиальное различие этих двух видов преобразований обнаруживается при рассмотрении способа доступа в память. В настоящее время большинство процедур высокого уровня выполняются в символьной форме, когда данные хранятся и обрабатываются в виде списков, а не в форме изображения. При выполнении операций нижнего уровня существует прямое физическое соответствие между позицией элемента на изображении и его расположением в памяти ЭВМ. Такое прямое соответствие «данные – память» уже отсутствует на более высоких уровнях обработки, где данные, как правило, вызываются из памяти с помощью указателей на связные списки.

Для символьных вычислений выделяют следующие операции, имеющие критическое время выполнения:

- операторы установления соответствия для получения логических выводов;
- сортировка по выбранным параметрам.

Воспользуемся классификацией вычислений, приведенной в работе [3]. Схема классификации, приведенная в таблице №2, может быть использована для определения оптимальной архитектуры в каждом конкретном случае

путем выбора алгоритмических блоков, как описано ниже. Так, например, если выделены вычислительные операции, то можно определить структуру требуемых вычислений.

В таблице №2 представлен набор программных операций, которые обычно используются при анализе изображений вместе с соответствующей классификацией.

Таблица № 2.

Характер часто применяемых операций для обработки изображений

Операция	a/b	c/d	e/f	g/h	i/j	k/l
Сегментация	a	d	f	h	i/j	l
Свертка	a	c	f	h	i/j	l
Сортировка	a	d	e	h	j	l
Построение гистограммы	b	c	f	h	j	l
Нахождение линии	b	d	e/f	g	j	l
Описание формы	b	c	f	g	j	k
Корреляция	a	c	f	h	j	l
Выбор точки	b	d	f	g	j	l

Отметим характерные особенности вышеприведенных операций. Большинство задач, связанных с предварительной обработкой изображения и выделением признаков характеризуется обработкой большого объема данных с проведением над ним массовых операций. Алгоритмы предобработки легко поддаются распараллеливанию, так как на этапах предварительного анализа вычисления носят локальный характер. Благодаря этому становится возможным производить фазу анализа в реальном времени, оставив заключительный анализ результирующих данных малого объема универсальной вычислительной машине, не обладающей сверхвысокой производительностью. Такие операции, как локальная фильтрация и свертка,

использующие данные только из локальной и заранее известной области обработки, требуют значительно меньшего объема памяти, чем те операции, исходные данные для которых разбросаны по всему изображению или расположение которых существенно зависит от результата предыдущих вычислений (например, при обработке некоторых отношений или при последовательном точечном анализе изображения). Эти особенности обработки являются важными и для разработчиков, – многие последние достижения, позволившие значительно увеличить быстродействие параллельных архитектур, основаны на использовании предсказуемости потока данных в плане его дальнейшего распараллеливания. За счет этого в них (при оптимальных условиях) производительность увеличивается пропорционально числу параллельных обрабатываемых элементов.

Исходя из сказанного, можно построить таблицу соответствия общих классов вычислительных архитектур типам вычислительных операций, используемых в процессе обработки изображений.

Таблица № 3.

Классификация архитектур для обработки изображений

Архитектура	a/b	c/d	e/f	g/h	i/j	k/l
Клеточно-числовая	++/+	++/0	-/+	--/+	+/0	+/--
Конвейерная	++/0	+/0	-/+	-/0	+/0	+/--
Теоретико-числовая	++/++	++/-	0/++	--/+	++/-	+/--
Систолическая	++/++	++/-	0/++	--/+	++/-	+/--
Потоковая	0/0	0/0	-/++	-/0	0/-	+/+
Матрично-потоковая	++/++	++/0	0/++	0/+	++/0	+/+
Ассоциативная	0/0	0/0	0/-	-/0	0/0	+/++

Обозначения:

отличное соответствие: ++

хорошее соответствие: +

удовлетворительное соответствие:	0
не соответствует:	-
совершенно не соответствует:	--

Из приведенной таблицы видно, что каждая архитектура имеет преимущества и недостатки в некоторых специальных областях обработки, но ни одна из них не является оптимальной для всех случаев. Так первая фаза анализа изображения – фильтрация, наиболее эффективна на клеточной архитектуре, однако все остальные операции, а тем более, рекурсивные алгоритмы сегментации, эффективно реализовать на такой структуре практически невозможно. С другой стороны, использование матрицы независимых процессоров (матрично-потокковая ВС) позволяет производить операции предобработки параллельно в каждой из зон изображения. Алгоритм фильтрации изображения на базе такой системы не столь эффективен, как при использовании клеточной архитектуры, однако применение одного и того же набора процессоров для выполнения сложных фаз обработки позволяет избежать дополнительной передачи данных в системе.

Одним из ключевых вопросов при создании такой многопроцессорной графической системы является оценка ее производительности (особенно на стадии проектирования). Сделано, однако, очень мало для количественной оценки возможностей предлагаемых архитектурных решений. Часто на практике задачу оценки производительности упрощают, сводя ее к оценке пиковой производительности системы в Mflops или Mips. Однако такую оценку нельзя считать достоверной, поскольку реальная производительность системы всегда оказывается ниже пиковой и зачастую на несколько порядков.

В качестве варианта решения можно предложить иерархические рекурсивные методы оценки производительности, например, SLALOM [4]

---

или HINT [5], критерием производительности которых служит «качество решения, полученное за фиксированное время». В этом случае «качество решения» – это точность вычисления некоторой функции. Так в тесте HINT используется пошаговое вычисление гиперболической функции на фиксированном интервале. При каждой новой итерации происходит уменьшение в два раза шага дискретизации, что приводит к лавинообразному нарастанию количества обрабатываемых данных, и соответственно, объема вычислений. Достоинствами данного подхода является возможность получения достоверной оценки производительности всей системы в целом, а также простота его реализации для различных архитектур. Адаптация данного метода для распределенных и многопроцессорных вычислительных систем [6] позволяет протестировать как уже существующие системы, так и оценить проектируемые посредством эмулятора на модели.

В качестве примера (рис. 1), приведем оценку тестирования нескольких вычислительных систем, произведенную как традиционными методами, так и методом HINT [4], в котором производительность системы определяется через QUIPS (Quality Improvement Per Second).

Параллельный метод оценки NAS показывает общую производительность (Cray Y-MP = 1.0) равную 0.94 для 128-процессорного nCUBE 2S, 1.61 для 256-процессорного nCUBE 2S, и 2.19 для 128-процессорного Paragon'a. Соотношение от 2 до 3 на процессор является постоянным для всех компьютеров, протестированных методом NAS. Однако эти цифры обманчивы, так как, например, 16-процессорный Paragon стоит больше, чем 64-процессорный nCUBE 2S при аналогичном уровне производительности.

Если сравнить при этом затраты на организацию тестирования, то параллельный метод оценки NAS требует (по экспертной оценке NAS) нескольких человеко-месяцев, чтобы настроить программу тестирования на

---



конкретную архитектуру. В то же время HINT обеспечивает ту же самую информацию менее, чем за два часа преобразований и прогонок, что говорит о более дешевом, эффективном и гибком способе тестирования архитектур.

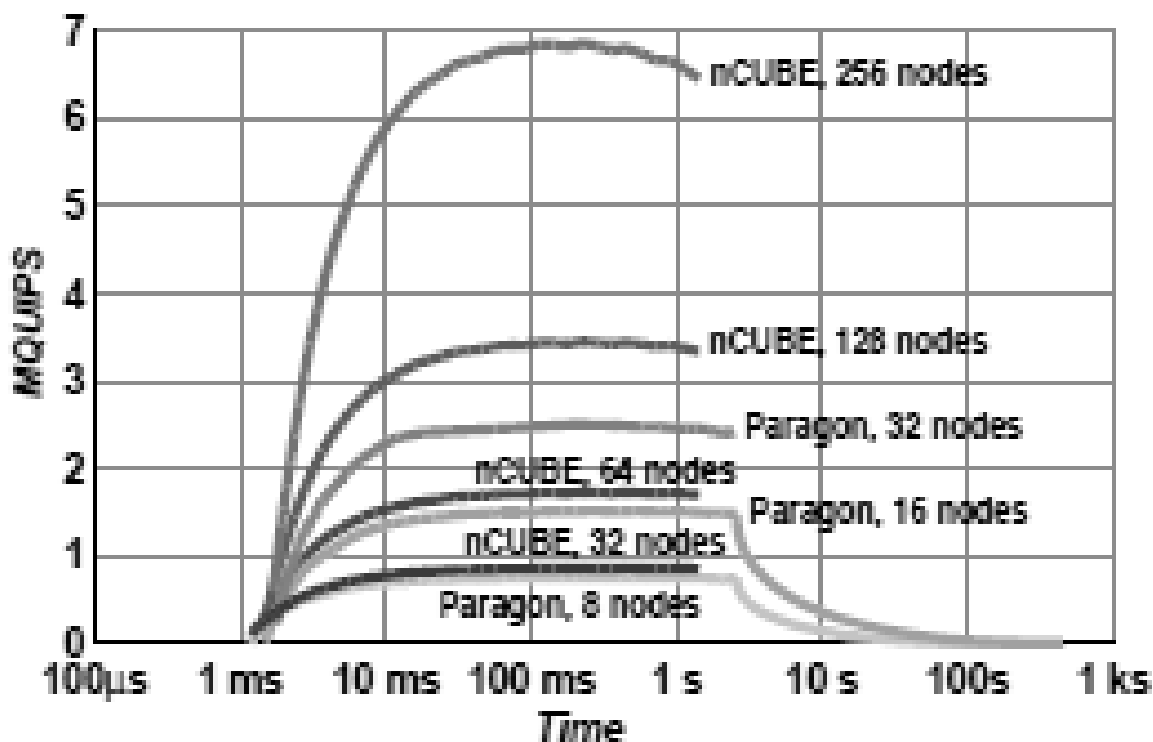


Рис. 1. – Результаты HINT-тестирования параллельных компьютеров.

Рисунок взят из [5]

### Заключение

Таким образом, в связи со сложностью и неоднородностью всего комплекса алгоритмов цифровой обработки и идентификации изображений ни одна из известных архитектур в чистом виде не может претендовать на роль оптимальной для решения всех этапов задач обработки изображений. С другой стороны, существует явная диспропорция в объемах данных и вычислениях на отдельных этапах обработки. При этом, если такие фазы как фильтрация, сегментация, выделение контуров, повышение контрастности изображения легко поддаются распараллеливанию практически без

накладных расходов, то последующие фазы анализа уже не имеют явно выделяемых параллельных участков вычислений. Это диктует применение комплексной архитектуры вычислителя, состоящего из нескольких блоков, ориентированных на специальную обработку, причем ту или иную архитектуры следует выбирать исходя из относительной важности используемых блоков обработки. Такой подход требует ясного понимания сложности и повторяемости каждой операции для выбора архитектуры системы [7-10]. Оценить назначение каждой операции внутри системы можно, вводя коэффициенты относительного веса или фактора ее стоимости. Если такое назначение можно сделать для большинства прикладных задач, оптимальная вычислительная архитектура выбирается на основе сопоставления комбинации вычислительных блоков и соответствующих им архитектур. При этом каждый блок может функционировать как подпрограмма, зашитая в аппаратуру и управляемая универсальной ЭВМ.

Наиболее целесообразным подходом к определению производительности распределенных и многопроцессорных систем представляется процесс их архитектурного моделирования, с последующей эмуляцией на полученной модели теста типа HINT.

*Работа выполнена в рамках проектной части госзадания Минобрнауки России № 2.3928.2017/4.6 в Южном федеральном университете*

### Литература

1. Айдинян А.Р. Аппаратные средства вычислительной техники. М.: Директ-Медиа, 2016. 125 с.
2. Руководство по архитектуре приложений Azure. URL: [docs.microsoft.com/ru-ru/azure/architecture/guide/](https://docs.microsoft.com/ru-ru/azure/architecture/guide/).
3. СБИС для распознавания образов и обработки изображений: Пер. с англ. /Под. ред. К. Фу. М.: Мир, 1988. 248 с.

4. Gustafson J. et al., The Design of a Scalable, Fixed-Time Computer Benchmark. URL: [johngustafson.net/pubs/pub30/pub30-alt.htm](http://johngustafson.net/pubs/pub30/pub30-alt.htm).

5. Gustafson John L. and Snell Quinn O. HINT: A New Way To Measure Computer Performance. URL: [johngustafson.net/pubs/pub47/Hint.htm](http://johngustafson.net/pubs/pub47/Hint.htm).

6. Каляев А.В., Гузик В.Ф., Каляев В.А., Костюк А.И., Поленов М.Ю. Оценка производительности многопроцессорных вычислительных систем с массовым параллелизмом. М.: Радио и связь, 2003. 136 с.

7. Мунтян Е.Р., Поленов М.Ю., Костюк А.И. О подходе к модернизации программной системы поддержки управленческих решений // Известия ЮФУ. Технические науки, 2015, № 3 (164). С. 54-62.

8. Мунтян Е.Р., Поленов М.Ю., Костюк А.И. Автоматизированная программная среда поддержки управленческих решений // Известия ЮФУ. Технические науки, 2014, № 6 (155). С. 101-108.

9. Земцов А.Н., Болгов Н.В., Божко С.Н. Многокритериальный выбор оптимальной системы управления базы данных с помощью метода анализа иерархий // Инженерный вестник Дона, 2014, №2. URL: [ivdon.ru/magazine/archive/n2y2014/2360/](http://ivdon.ru/magazine/archive/n2y2014/2360/).

10. Воробьев С.П., Горобец В.В. Исследование модели транзакционной системы с репликацией фрагментов базы данных, построенной по принципам облачной среды // Инженерный вестник Дона, 2012, №4 (часть 1). URL: [ivdon.ru/magazine/archive/n4p1y2012/1149/](http://ivdon.ru/magazine/archive/n4p1y2012/1149/).

### References

1. Aydinian A.R. Apparathnye sredstva vychislitel'noy tekhniki [Hardware Computing]. М.: Direkt-Media, 2016. 125 p.

2. Rukovodstvo po arkhitekture prilozheniy Azure [Azure Application Architecture Guide]. URL: [docs.microsoft.com/ru-ru/azure/architecture/guide/](https://docs.microsoft.com/ru-ru/azure/architecture/guide/).



3. SBIS dlya raspoznavaniya obrazov i obrabotki izobrazheniy [VLSI for pattern recognition and image processing]: Per. s angl. Pod. red. K. Fu. M.: Mir, 1988. 248 p.

4. Gustafson J. et al., The Design of a Scalable, Fixed-Time Computer Benchmark. URL: [johngustafson.net/pubs/pub30/pub30-alt.htm](http://johngustafson.net/pubs/pub30/pub30-alt.htm).

5. Gustafson John L. and Snell Quinn O. HINT: A New Way To Measure Computer Performance. URL: [johngustafson.net/pubs/pub47/Hint.htm](http://johngustafson.net/pubs/pub47/Hint.htm).

6. Kalyaev A.V., Guzik V.F., Kalyaev V.A., Kostyuk A.I., Polenov M.Yu. Otsenka proizvoditel'nosti mnogoprotsessornykh vychislitel'nykh sistem s massovym parallelizmom [Performance evaluation of multiprocessor computing systems with mass parallelism]. M.: Radio i svyaz', 2003. 136 p.

7. Muntyan E.R., Polenov M.Yu., Kostyuk A.I. Izvestiya YuFU. Tekhnicheskie nauki, 2015, № 3 (164). pp. 54-62.

8. Muntyan E.R., Polenov M.Yu., Kostyuk A.I. Izvestiya YuFU. Tekhnicheskie nauki, 2014, № 6 (155). pp. 101-108.

9. Zemtsov A.N., Bolgov N.V., Bozhko S.N. Inzhenernyj vestnik Dona (Rus), 2014, №2. URL: [ivdon.ru/magazine/archive/n2y2014/2360/](http://ivdon.ru/magazine/archive/n2y2014/2360/).

10. Vorob'ev S.P., Gorobets V.V. Inzhenernyj vestnik Dona (Rus), 2012, №4 (part 1). URL: [ivdon.ru/magazine/archive/n4p1y2012/1149/](http://ivdon.ru/magazine/archive/n4p1y2012/1149/).