

ОРГАНИЗАЦИЯ ГРАФИЧЕСКОЙ КЛАСТЕРНОЙ СИСТЕМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ ВЫСОКОПРОИЗВОДИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

Запропоновано архітектуру графічного кластеру, що орієнтований на одночасне розв'язання різних класів задач з віртуалізацією апаратних ресурсів. Обґрунтовано вибір програмних і апаратних компонентів для побудови кластерної системи. На основі запропонованої архітектури розв'язана задача високопродуктивного злиття зображень з високою розрізняювальною здатністю.

A graphical cluster architecture based on the simultaneous solution of different classes of problems with the virtualization of hardware resources was proposed. Hardware and software components to build a cluster system are choice. On the basis of the proposed architecture the problem high-resolution image fusion is solved.

Введение

В современном мире увеличение объемов обрабатываемой информации приводит к необходимости увеличивать производительность вычислительных устройств при решении задач в различных отраслях. Примерами таких задач являются: моделирование сложных динамических процессов, обработка и хранение больших объемов данных. Например, моделирование распространения вредных примесей в атмосфере, что связано с решение больших систем дифференциальных уравнений [1], анализ и обработка спутниковых снимков, которые характеризуются высоким разрешением [2], анализ и обработка огромных массивов данных, накопленных о погоде и климате в течение длительного промежутка времени [3].

Важным фактором, влияющим на производительность вычислительной системы применительно к конкретной задаче или классу задач, является ее архитектура.

Достаточно давно увеличение скорости обработки информации достигается за счет распараллеливания. Поэтому разработка способов организации параллельных вычислительных систем для высокопроизводительной обработки данных является актуальной научной задачей.

Применение GPU как базиса для построения кластерных систем

Параллельные компьютерные системы можно условно классифицировать в соответствии с уровнем, на котором оборудование поддерживает параллелизм [4]:

- многоядерные системы;
- симметричные мультипроцессорные системы;
- распределенные компьютерные системы;

Распределенный компьютер представляет собой компьютерную систему с распределенной памятью, в которой вычислительные устройства объединены в единую систему по сети. Данный вид параллельных компьютерных систем проявляет высокую

способность к масштабированию. Основные подклассы распределенных компьютерных систем следующие:

- кластерные системы;
- массивно параллельные компьютеры;
- грид-системы;
- специализированные параллельные компьютеры;
- FPGA-системы;
- GPGPU-системы.

Особый интерес в данное время вызывают системы с архитектурой GPGPU. GPGPU (General Purpose GPU) - системами называют компьютеры, позволяющие производить вычисления общего назначения на видеокартах.

Вычисления общего назначения на видеокартах являются современным трендом исследований в области компьютерной инженерии. Они позволяют задействовать мощности современных графических карт для решения задач, которые не ограничиваются областью компьютерной графики.

Графические процессоры являются, по сути, сопроцессорами, значительно оптимизированными под задачи компьютерной графики - в частности под решение задач линейной алгебры.

На данный момент существует несколько технологий (например, CUDA и OpenCL) [5], позволяющие использовать достаточно большие мощности современных видеокарт без изучения особенностей низкоуровневого программирования данных устройств.

Одним из наибольших преимуществ данных систем является высокая распространенность (практически каждый современный персональный компьютер оснащен видеокартой, поддерживающей вычисления общего назначения) и низкая стоимость единицы вычислительной мощности по сравнению с, например, кластерными системами.

При построении современных вычислительных систем для высокопроизводительной обработки данных перспективным является совмещение возможностей кластерных систем и графических процессоров. Графические процессоры обеспечивают параллельную обработку данных по принципу SIMD с применением большого числа потоков инструкций (100, 1000 и более) [5].

В то же время, системы на основе GPU обладают относительно небольшим объемом памяти.

В свою очередь, кластер, состоящий из нескольких узлов, позволяет обеспечить независимую обработку данных с использованием практически неограниченного объема памяти, но с временными задержками при передаче данных с одного узла кластера на другой.

Совмещение кластерной архитектуры и графических процессоров в каждом узле кластера позволяет воспользоваться преимуществами каждой архитектуры.

Применение графических кластерных систем упоминается в литературе [6-7] как эффективный инструмент для решения задач моделирования климата, обработки изображений и т.п.

Проблемы построения универсальной кластерной системы на основе GPU

При организации универсальной кластерной системы на основе графических процессоров возникают следующие проблемы:

- обеспечить независимое функционирование каждого узла кластера при решении разных задач одновременно;
- обеспечить гибкое независимое конфигурирование архитектуры кластера для одновременного решения разных задач;
- обеспечить динамическое выделение и миграцию ресурсов системы между узлами кластера.

Указанные выше проблемы возникают в связи с тем, что при построении кластерной системы присутствует естественное ограничение по количеству имеющихся в наличии физических узлов. Организация кластера усложняется тем, что указанные ресурсы в разнообразных сочетаниях необходимы в один момент времени (кластер универсальный и должен позволять решать разные задачи независимо друг от друга).

Фактически необходимо отобразить векторы ресурсов каждой задачи на реальные физические ресурсы кластерной системы. Для выполнения данного условия можно предложить два способа организации каждого узла кластера (рис. 1).

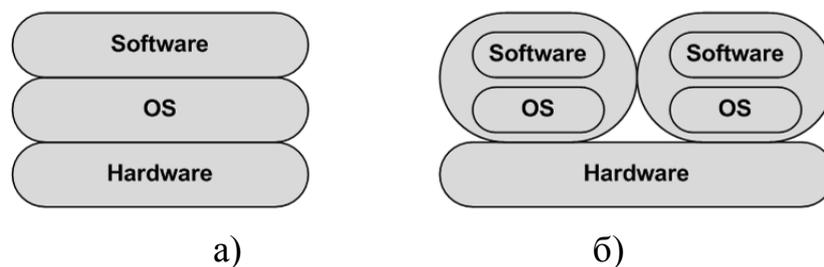


Рисунок 1 – Организация узла кластера

На рис. 1а показан вариант организации узла, при котором физические ресурсы узла напрямую предоставлены операционной системе и прикладному программному обеспечению. Это самый простой способ, но в этом случае выполнение условия независимости задач становится трудновыполнимым, т.к. в рамках одной операционной системы практически невозможно организовать изолированный доступ разных задач к одним и тем же физическим ресурсам. При этом прикладное программное обеспечение и задачи могут требовать разные виды операционных систем, что также невыполнимо при такой организации.

На рис. 1б показан альтернативный вариант организации узла кластера. Каждый узел представляет собой группу контейнеров, каждый из которых имеет доступ к одним и тем же физическим ресурсам, но в каждом контейнере организована изолированная среда для выполнения своей операционной системы и своего набора программного обеспечения. Такое решение называется виртуализацией, а контейнеры – виртуальными машинами.

Известно множество готовых решений для виртуализации [8]:

VMWare; Virtual Box; Parallels; Xen; KVM.

При выборе платформы для виртуализации необходимо учесть, что виртуальная машина – это прослойка между аппаратным обеспечением и гостевой операционной

системой. Поэтому при построении кластера нецелесообразно использовать платформу виртуализации, которая представлена в виде программного продукта, который запускается под управлением хост-операционной системой.

На основе приведенных критериев отбора была составлена сравнительная таблица известных платформ виртуализации, которые обеспечивают «чистую» виртуализацию, т.е. работают в режиме гипервизора на аппаратной платформе без необходимости в отдельной операционной системе (см. табл. 1).

Анализ табл. 1 показывает, что «чистую» виртуализацию обеспечивает незначительное число виртуальных машин (Hyper 5 и VMWare ESX). Учитывая тот факт, что VMWare ESX обеспечивает более широкий спектр поддерживаемых операционных систем и то, что имеющаяся в наличии аппаратная конфигурация узла кластера позволяет использовать данный виртуализатор по бесплатной лицензии, в качестве платформы для виртуализации был выбран продукт VMWare ESXi 5.0.

Таким образом, аппаратная конфигурация одного узла кластера следующая:

CPU: Intel Core 2 Quad Q8400; RAM: 4 GiB; HDD: 4TiB; video card NVIDIA GeForce GTX 460; PCIe network card (1Gbit/s).

Таблица 1 – Сравнение платформ виртуализации

Виртуальная машина	Хост-процессор	Хост -ОС	Поддерживаемые ОС	Лицензия
VMWare ESX	Intel x86, AMD64	Нет	Linux, Windows и т.д.	Проприетарная
Virtual Box	Intel x86, AMD64	Linux, Windows и т.д.	Linux, Windows и т.д.	GPL
Parallels	Intel x86, Intel VT-x	Linux, Windows и т.д.	Linux, Windows и т.д.	Проприетарная
Xen	Intel x86, AMD64	Linux	Linux, Windows и т.д.	GPL
KVM	Intel AMD	Linux	Linux, Windows и т.д.	GPL2
Hyper V	Intel x86-64, AMD64	Нет	Linux, Windows и т.д.	Проприетарная

Основные характеристики гипервизора, установленного на каждом узле:
 не более 200 Мб занимает сам гипервизор;
 обеспечивает запуск следующих гостевых ОС: Windows, Red Hat, Solaris, Debian, Mac OS X, Ubuntu и т.д.;
 технология Direct Driver для доступа к устройствам ввода-вывода;
 организация ресурсов в виде динамически распределяемых пулов.

Архитектура кластерной системы на основе GPU

В результате применения технологии виртуализации структура графического кластера будет иметь следующий вид: рис. 2.

На рис. 2 показано M узлов кластера (в реальной системе $M = 8$). Каждый узел обладает набором разделяемых ресурсов (процессор, память, графический процессор и т.п.).

На аппаратной платформе каждого узла развернут гипервизор VMWare ESXi. Внутри гипервизора организован блок из N виртуальных машин, где N - количество задач. В составе кластера присутствуют виртуальные машины с гостевыми операционными системами Windows и Linux семейств.

Для формирования общей вычислительной инфраструктуры в состав каждой виртуальной машины введен виртуальный сетевой интерфейс, который обеспечивает подключение виртуального узла к сети кластера. Сеть кластера построена на основе сети Ethernet со скоростью передачи данных 1Gbit/s.

В результате на базе одного физического кластера в любой момент можно получить произвольное количество виртуальных кластеров, как объединение группы виртуальных узлов в единую вычислительную систему. Гибкое конфигурирование структуры кластера обеспечивается системой управления. В качестве системы управления используется VMWare vSphere Client. Для запуска данного программного обеспечения в составе кластерной системы выделен управляющий терминал. На рис. 3 показан интерфейс управляющего терминала при подключении к хосту (физическому узлу) в режиме доступа к консоли виртуального узла. Для обеспечения связи с внешним миром кластерная система имеет доступ к сети Интернет по двум независимым каналам.

Решение задачи слияния изображений на кластере графических процессоров

Как было сказано ранее, предложенная архитектура графической кластерной системы предназначена для решения задач высокопроизводительной обработки данных, в частности, изображений с высокой разрешающей способностью.

На данный момент практически построенная система используется для решения двух независимых задач: анализ климатических данных [3] и обработка спутниковых снимков с равноправным разделением ресурсов (50%).

В качестве примера реализации рассмотрим результаты, полученные на графическом кластере для задачи слияния спутниковых снимков, которая описана в [9].

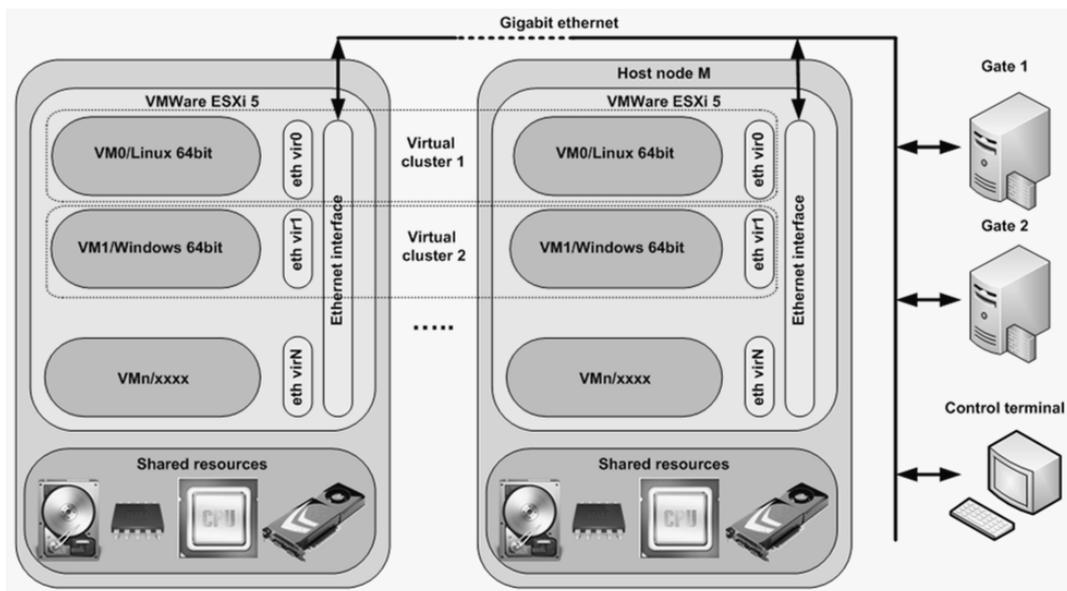


Рисунок 2 – Архитектура графической кластерной системы

При реализации данной задачи для обеспечения сравнения эффективности параллельных реализаций методов были выбраны два метода: взвешенное усреднение [10] и метод, основанный на преобразовании цветового пространства (IHS) [11]. Кроме того, для всех реализованных методов метрики были также получены для кластера процессоров (без GPU).

Для проведения экспериментов на реальных данных были использованы данные, предоставляемые проектом USGS Global Visualization Viewer [12] от Геологической службы США. В экспериментах были использованы снимки со спутника Landsat 7. Снимки - это фотографии районов Донецкой области (Украина). При реализации методов были задействованы технологии CUDA и MPI.

В таблицах 2 и 3 используются следующие обозначения:

WA – метод взвешенного усреднения; IHS – метод, основанный на преобразовании цветового пространства; HDWT – метод дискретного вейвлет-преобразования Хаара; DDWT – метод дискретного вейвлет-преобразования Добеши.

Как следует из таблицы 2, реализация метода IHS имеет лучшее время на кластере с GPU. Это объясняется тем, что набор команд GPU является более эффективным для реализации IHS. При этом реализация метода HDWT имеет лучшее время на кластере с процессором (см. табл. 3). Это объясняется тем, что дискретное вейвлет-преобразование эффективно реализуется с помощью арифметических операций процессора.

Для оценки производительности в работе оценивалось количество пикселей изображения, обрабатываемых в единицу времени в предположении, что все операции, в том числе, для работы с памятью, в среднем равны по сложности. На рис. 4 показано сравнение увеличения производительности для каждого метода при реализации на графическом кластере предложенной архитектуры. Анализируя рис. 4, можно сделать вывод, что прирост производительности параллельной реализации на GPU лежит в диапазоне 2-18 раз в зависимости от выбранного метода.

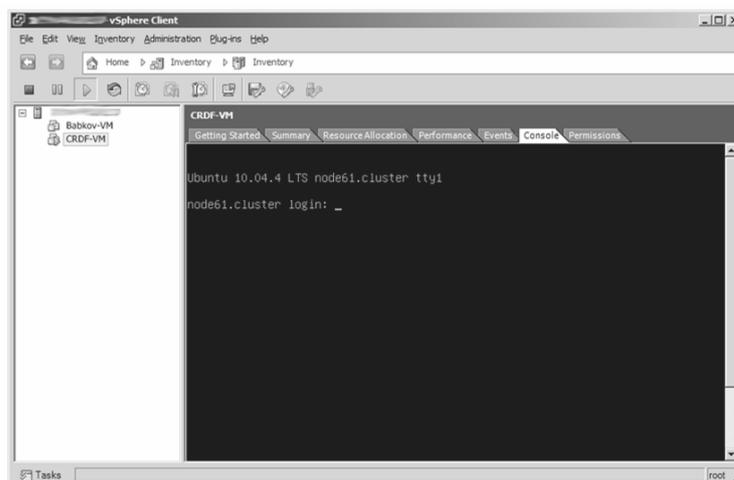


Рисунок 3 – Подключение к хосту в режиме доступа к консоли виртуального узла

Как следует из таблицы 2, реализация метода IHS имеет лучшее время на кластере с GPU. Это объясняется тем, что набор команд GPU является более эффективным для реализации IHS. При этом реализация метода HDWT имеет лучшее время на кластере с процессором (см. табл. 3). Это объясняется тем, что дискретное вейвлет-преобразование эффективно реализуется с помощью арифметических операций процессора.

Для оценки производительности в работе оценивалось количество пикселей изображения, обрабатываемых в единицу времени в предположении, что все операции, в том числе, для работы с памятью, в среднем равны по сложности. На рис. 4 показано сравнение увеличения производительности для каждого метода при реализации на графическом кластере предложенной архитектуры. Анализируя рис. 4, можно сделать вывод, что прирост производительности параллельной реализации на GPU лежит в диапазоне 2-18 раз в зависимости от выбранного метода.

Таблица 2 – Экспериментальные результаты для графического кластера

<i>Ширина, px</i>	<i>Высота, px</i>	<i>T_{WA}, с</i>	<i>T_{IHS}, с</i>	<i>T_{HDWT}, с</i>	<i>T_{DDWT}, с</i>
16280	14960	0,73	3,01	3,61	3,70
8140	7480	0,90	0,85	0,98	1,02
4070	3736	0,32	0,30	0,34	0,35

Таблица 3 – Экспериментальные результаты для кластера процессоров

Ширина, px	Высота, px	T _{WA} , с	T _{IHS} , с	T _{HDWT} , с	T _{DDWT} , с
------------	------------	---------------------	----------------------	-----------------------	-----------------------

Ширина, px	Высота, px	T_{WA} , с	T_{IHS} , с	T_{HDWT} , с	T_{DDWT} , с
16280	14960	49,0	67,1	11,6	13,5
8140	7480	2,08	6,24	2,89	3,23
4070	3736	0,51	1,57	0,66	0,73

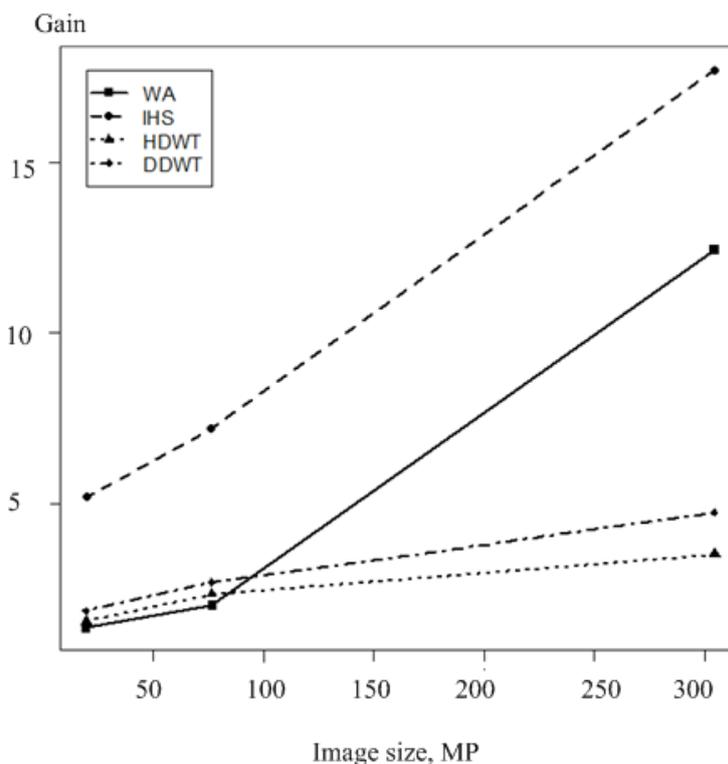


Рис. 4 - Прирост производительности для параллельных реализаций на графическом кластере

Заключение

Предложена архитектура и набор программных и аппаратных компонентов для организации графического кластера с возможностью одновременного изолированного выполнения разных классов задач с обеспечением гибкого распределения ресурсов. На предложенной архитектуре решена задача слияния спутниковых снимков и получено практическое ускорение в 2-18 раз по сравнению с кластером обычных процессоров.

Научная новизна заключается в решении задачи слияния мультиспектральных изображений на кластере графических процессоров.

Практическая значимость заключается в обеспечении с помощью предложенной архитектуры кластерной системы универсальной платформы для независимого решения задач высокопроизводительной обработки данных на основе графического кластера.

Дальнейшее направление исследований связано с накоплением данных об использовании ресурсов кластерной системы при решении разных классов задач и формировании рекомендаций и методов оптимального распределения ресурсов.

Литература

1. F. Molnár Jr., T. Szakály, R. Mészáros, I. Lagzi, Air pollution modelling using a Graphics Processing Unit with CUDA. *Computer Physics Communications* V. 181 - 2010. - P. 105-112.
2. S. Yoo, S. Jo, K. Choi, C. Jeong, "A Framework for Multisensor Image Fusion using Graphics Hardware", *Proc. of the 11-th International Conference on Information Fusion*, June, 2008, P. 1-5.
3. Rodrigues Zalipynis R.A., Zapletin E.A., Averin G.V. *The Wikience: Community Data Science. Concept and Implementation.*, Proceedings of the 7th International Scientific-Technical Conference "Informatics and Computer Technologies" (ICT-2011), Vol. 1, pp. 113-117, Donetsk, November 22-23, 2011.
4. Tanenbaum E. *Computer architecture* / E. Tanenbaum: 4th ed. - St. Univ, Peter, 2003. – P.697.
5. J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley, 2011
6. С.Д. Погорелый, Ю.В. Бойко, М.И. Трибрат, Д.Б. Грязнов, Анализ методов повышения производительности компьютеров с использованием графических процессоров и программно-аппаратной платформы CUDA, *Математичні машини і системи*, 2010, № 1, С. 40-54.
7. Z. Fan, F. Qiu, A. Kaufman, S. Yoakum-Stover, GPU Cluster for High Performance Computing, *ACM / IEEE Supercomputing Conference 2004*, November 06-12, Pittsburgh, PA, P. 20-32
8. Comparison of platform virtual machines [Электронный ресурс]. Режим доступа:
http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines
9. V. Babkov, K. Titarenko, "Fusion of multispectral images in the problem of satellite images processing", *Proc. of the conference Simulation and computer graphics*, Donetsk, Ukraine, October 2011, pp. 25-29.
10. S. Rahmani, M. Strait, D. Merkurjev, M. Moeller, and Todd Wittman. "An Adaptive IHS Pan-sharpening Method", *IEEE Letters on Geoscience and Remote Sensing*, vol. 7(11), 2010, pp. 1545-1550.
11. G. Pajares, J.M. de la Cruz, "Wavelet-based image fusion tutorial", *Pattern Recognition*, vol. 37, no. 9, 2004, pp. 1855-1872.
12. USGS Global Visualization Viewer [Электронный ресурс]. Режим доступа: <http://glovis.usgs.gov/>.