



УДК 004.021, 004.415

© 2015 Г. **С.В. Мельман**, канд. техн. наук
(Дальневосточный федеральный университет, Владивосток),

В.А. Бобков, д-р техн. наук,

А.С. Черкашин

(Институт автоматизации и процессов управления ДВО РАН, Владивосток)

ТЕХНОЛОГИЯ РАСПРЕДЕЛЕННО-ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ И ВИЗУАЛИЗАЦИИ БОЛЬШИХ ОБЪЕМОВ СИНОПТИЧЕСКИХ ДАННЫХ*

Представлена технология создания системы визуализации больших объемов синоптических данных с использованием распределенных и параллельных гибридных вычислений.

Ключевые слова: научная визуализация, распределенные и параллельные вычисления, клиент-серверная технология, синоптические данные.

Введение

Исследования динамики атмосферных явлений и океана, выполняемые метеорологами, подразумевают работу с большими объемами пространственных данных. Поэтому при разработке программных систем, обеспечивающих автоматизацию научных исследований в этой области, возникают две основные задачи, от результативности решения которых зависит эффективность системы в целом: быстрая обработка больших объемов данных и визуализация 3D данных, облегчающая пространственное восприятие синоптических данных и анализ погодных явлений.

Ранее разработанная авторами система визуализации синоптических данных [1] предназначалась для работы на обычном персональном компьютере. Система представляла собой набор программных приложений и отдельную программу визуализации. При этом обеспечивалась приемлемая скорость визуализации

* Работа выполнена при финансовой поддержке программы РАН по стратегическим направлениям развития науки «Фундаментальные проблемы математического моделирования» (проект «Разработка и адаптация алгоритмов и математического обеспечения на супер-ЭВМ с экстремальным параллелизмом для решения задач моделирования и прогнозирования чрезвычайных ситуаций на море с применением данных спутникового мониторинга») и программы «Дальний Восток» (проект 15-И-4-025).

для результатов предобработки данных, сохраняемых на жестком диске. Однако последующее развитие системы по нескольким направлениям: расширение функциональности визуального анализа 3D объектов с более развитым интерфейсом и использованием динамически подключаемых модулей нескольких типов, многоплановая и темпоральная анимация, возросшие объемы спутниковых данных – потребовало принципиального повышения вычислительной производительности. Поэтому новая версия системы ориентирована на использование многопроцессорных систем с организацией параллельных вычислений.

С развитием технологий гетерогенного параллелизма появились работы с использованием гибридно-распределенных вычислений, как правило, учитывающих специфику конкретной тематики. Например, в работе [2] авторы предлагают использовать гибридно-распределенный подход с использованием MPI и многопроцессорных вычислительных архитектур для визуализации результатов математического моделирования векторных полей.

Вопросы одновременного использования многоуровневого параллелизма и распределенных вычислений и связанные с этим проблемы рассматривались авторами [3, 4]. В работе [5] исследовались вопросы визуализации результатов моделирования физических полей на примере моделирования распространения акустического сигнала.

В настоящей работе предлагается технология и система визуализации больших объемов синоптических данных с реализацией многоуровневого гибридного (GPU+CPU) параллелизма и распределенных вычислений. Новая архитектура программного комплекса построена на использовании клиент-серверной технологии с централизованной БД и возможностью подключения в качестве вычислительных узлов как персональных компьютеров, так и супер-ЭВМ. Используемый модульный подход обеспечил возможность расширения алгоритмической и функциональной базы за счет подключения дополнительных модулей по технологии plug-in.

Технология параллельно-распределенной обработки

Предлагаемая технология параллельно-распределенной обработки больших объемов данных на гетерогенных вычислительных архитектурах применительно к обработке и визуализации спутниковых синоптических данных основывается на следующих принципах:

- 1) распределенные вычисления – для обеспечения вычислений в реальном времени;
- 2) клиент-серверная модель – для централизованного управления;
- 3) распределенная база данных – здесь широко применяется принцип кэширования данных, каждый узел содержит только актуальные данные и только необходимые для работы модулей подключенных к этому узлу;
- 4) plug-in технология;
- 5) гибридный параллелизм (GPGPU+CPU) в подключаемых модулях.

Эти принципы реализованы в новой архитектуре системы визуализации. В архитектуре также учитывается специфика работы с синоптическими данными.

Под синоптическими данными подразумеваются скалярные и векторные поля распределения погодных характеристик. Данные могут быть одномерными (показания зондов, стационарных радиолокационных станций), двумерными (погодные карты, спутниковые снимки облачности, льдов и т.д.) и пространственными (поля ветра, давления, температуры и т.д.). Кроме того, система визуализации должна работать со стандартными, используемыми синоптиками форматами данных. При этом данные могут быть как статическими, так и изменяемыми во времени, т.е. динамическими.

Также предлагаемая архитектура дает пользователю совершенно прозрачную схему работы с данными, которая охватывает все стадии работы с данными – от запроса данных со специализированных БД до визуализации. Каждая из описанных стадий подготовки данных выполняется с помощью подключаемого модуля (plug-in), что позволяет значительно расширять функционал системы в целом.

Процесс визуализации в системе выполняется как простая циклическая последовательность: получение данных, обработка, подготовка к визуализации, визуализация.

Наибольшая вычислительная нагрузка связана с этапом обработки и подготовки к визуализации. Сюда включаются моделирование, восполнение, фильтрация, разряжение данных. Эта работа возлагается на распределенную вычислительную систему. Непосредственно визуализация осуществляется полностью на ПК пользователей, оснащаемых современными мощными графическими ускорителями.

В основу предыдущей версии системы был заложен модульный принцип, согласно которому все функциональные блоки были реализованы в качестве отдельных программных динамических модулей.

Необходимые динамические модули подключались автоматически. Такой подход позволил наращивать алгоритмическую базу без изменений основного приложения СВ.

Указанные решения позволили в значительной степени использовать предыдущие наработки, включая модули загрузки, фильтрации и визуализации.

Клиент-серверная часть распределенных вычислений выполняет следующие функции:

- серверная часть – обеспечивает централизованное управление процессом сбора и обработки данных;

- клиентское вычислительное приложение – обеспечивает подключение отдельных вычислительных модулей (plug-in) на отдельном узле (ПК);

- клиентское интерфейсное приложение – предоставляет пользователю интерфейс управления всей системой и необходимый инструментарий для визуального анализа.

Реализованные протоколы обмена данных позволяют запускать клиентские вычислительные приложения на кластере.

При этом необходимо портирование приложения и вычислительных модулей на операционную систему кластера.

Конвейер обработки данных

Процесс обработки и визуализации данных представлен на рис. 1 в виде разветвленного конвейера.

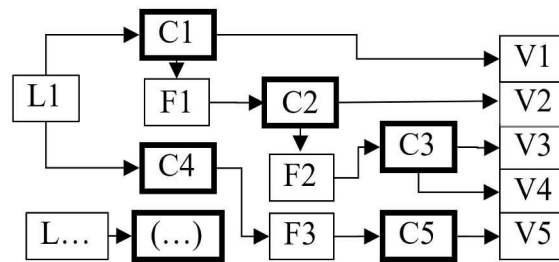


Рис. 1. Конвейер обработки данных.

В начале работы с системой пользователю предлагается список модулей для работы с данными (модуль загрузки – L). Пользователь может выбрать несколько модулей загрузки одновременно. Один модуль может работать с несколькими выборками данных, заданными пользователем, которые образуют контейнеры данных (контейнеры – C). Контейнер с данными – это набор данных с идентификатором, определяющим тип данных. Далее для каждого контейнера, при необходимости, определяется набор фильтров (фильтры – F), результатом работы которых также является контейнер с данными. Пользователю предлагается набор модулей визуализации (модуль визуализации – V) для контейнеров с данными согласно их типу. Для одного контейнера можно добавить несколько модулей визуализации, а на этапе выбора из всего набора модулей обработки пользователю доступны только те, которые предназначены для типа данных из контейнера.

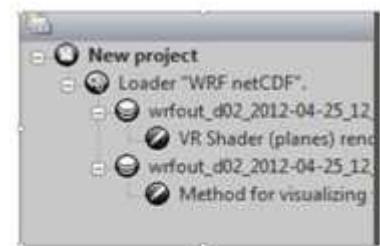


Рис. 2. Древоподобное представление конвейера обработки данных в пользовательском приложении.

В клиентском интерфейсном приложении конвейер обработки данных удобно представлять в виде древоподобной структуры (рис. 2).

Система визуализации синоптических объектов с параллельно-распределенной обработкой

Обработка входных данных и функциональность системы. Данные перед визуализацией проходят несколько стадий:

- 1) сбор данных от источников (искусственные спутники Земли, зонды, метеостанции);
- 2) пересылка к удаленному хранилищу (УБД) – удаленная база данных, не являющаяся частью описываемой системы визуализации;
- 3) запрос пользователя к УБД для выбора интересующих данных и получение данных с удаленного сервера: профилей спутникового мониторинга по протоколу HTTP/FTP/Samba и др., считывание данных из файловых контейнеров датацентра (NETCDF погодной модели WRF) или хранящихся локально на жестких

либо сетевых дисках;

4) конвертация, распаковка (данные могут храниться в текстовом или компактном бинарном виде). Спутниковые данные, данные зондов и метеостанций имеют узкоспециализированные форматы данных;

5) предобработка согласно заданным пользователем параметрам (применение моделей преобразования);

6) пространственное и темпоральное восполнение (применение моделей восполнения), регуляризация, фильтрация;

7) подготовка к визуализации – загрузка актуальных для текущего ракурса и момента времени данных в оперативную память ПК;

8) визуализация.

Синоптиками-исследователями система предоставляет следующие возможности:

развитые возможности интерактивного визуального анализа пространственных физических полей синоптических объектов;

возможность простого распространения системы между обычными персональными компьютерами; высокая производительность, необходимая для быстрой обработки и визуализации синоптических данных;

расширяемость алгоритмической базы, необходимая для добавления поддержки новых форматов данных и методов их обработки/визуализации;

интерактивный удобный интерфейс с возможностью управления данными, параметрами и методами визуализации, хронологией времени для динамических данных;

одновременная работа с разными наборами данных в одном окне;

одновременная работа нескольких алгоритмов визуализации на одном наборе данных.

Сервер. Серверная часть системы реализована в программном приложении, объединяющем в себе базу данных и супервизор. Супервизор координирует работу распределенной системы, формируя пакеты с заданиями для вычислительных узлов и их модулей. База данных используется для централизованного сбора и хранения результатов обработки данных с вычислительных узлов. В БД хранятся только актуальные данные для текущих настроек конвейера. Если пользователь поменял параметры какого-либо вычислительного модуля, то все контейнеры данных, которые от него зависят, очищаются и пересчитываются заново, после чего актуализируются на стороне клиентского интерфейсного приложения для визуализации.

Протокол обмена данными в базе данных реализован на TSP, который обеспечивает доставку данных без потерь.

Порядок работы:

1) система запускается на одном или нескольких ПК, объединенных ЛВС. На каждом ПК с клиентским вычислительным приложением должны быть одинаковые наборы вычислительных модулей. Для функционирования системы должны быть запущен один сервер, как минимум, одно вычислительное приложение и одно интерфейсное приложение;

- 2) пользователь задает конфигурацию конвейера в клиентском приложении;
- 3) супервизор запускает конвейер расчета актуальных данных – рассылает пакет с заданиями для старта расчетов вычислительными приложениями;
- 4) супервизор сообщает интерфейсному приложению об окончании расчетов;
- 5) интерфейсное приложение забирает те контейнеры с данными, которые согласно заданной конфигурации участвуют в визуализации;
- 6) интерфейсное приложение запускает процесс рендеринга;
- 7) если пользователь меняет конфигурацию конвейера или настройки вычислительных модулей, то супервизор начинает процесс с пункта 3, иначе вычислительный процесс останавливается, работает только интерфейсное приложение.

Схема распределения вычислений исключает возможность запроса на запись одного и того же набора данных из разных вычислительных приложений, так как между вычислительными узлами распределяется работа по расчету данных на разные моменты «времени» (имеется в виду дата/время съемки данных). Параллелизм в рамках расчета одного момента «времени» идет на уровне вычислительного модуля. Таким образом, удалось избежать схемы с блокировкой базы данных, и, соответственно, сбор данных от разных вычислительных приложений может происходить независимо в параллельном режиме.

Клиентское интерфейсное приложение. Актуальными считаются данные только для выбранной временной точки динамических данных. Таким образом, для темпоральной анимации на каждом шаге временной шкалы необходима работа вычислительного конвейера. В этом случае узким местом будет пересылка данных по ЛВС. Чтобы избежать замедления темпоральной визуализации в клиентском интерфейсном приложении, предусмотрен кэш данных. При первом проходе по темпоральной шкале все полученные данные в ключевых моментах сохраняются на жестком диске. После первого прохода, когда все данные собраны локально в бинарном виде, визуализация выполняется на большой скорости. Ключевые моменты – это моменты времени, в которые были сделаны замеры набора данных. Промежуточные сетки данных, между ключевыми моментами, вычисляются на лету на клиентской стороне. При завершении работы приложения все временные файлы удаляются. Эксперименты показали, что скорости работы современных ЦП и жестких дисков вполне достаточно для плавной темпоральной анимации данных из кэша, так как основная вычислительная нагрузка идет именно в конвейере обработки данных.

Если пользователь изменил параметры фильтров или других вычислительных модулей, то, как было сказано выше, происходит пересчет данных, но порядок расчета ключевых моментов происходит не от самого раннего к самому позднему, а в обе стороны от текущего положения темпорального ползунка. Это сделано для того, чтобы пользователь как можно быстрее мог продолжать визуальные исследования данных в окрестности текущего «времени».

Более подробно функциональные возможности и инструментарий визуального анализа клиентского интерфейсного приложения можно найти в работах [1, 6]. На рис. 3 и 4 показаны примеры визуализации данных.

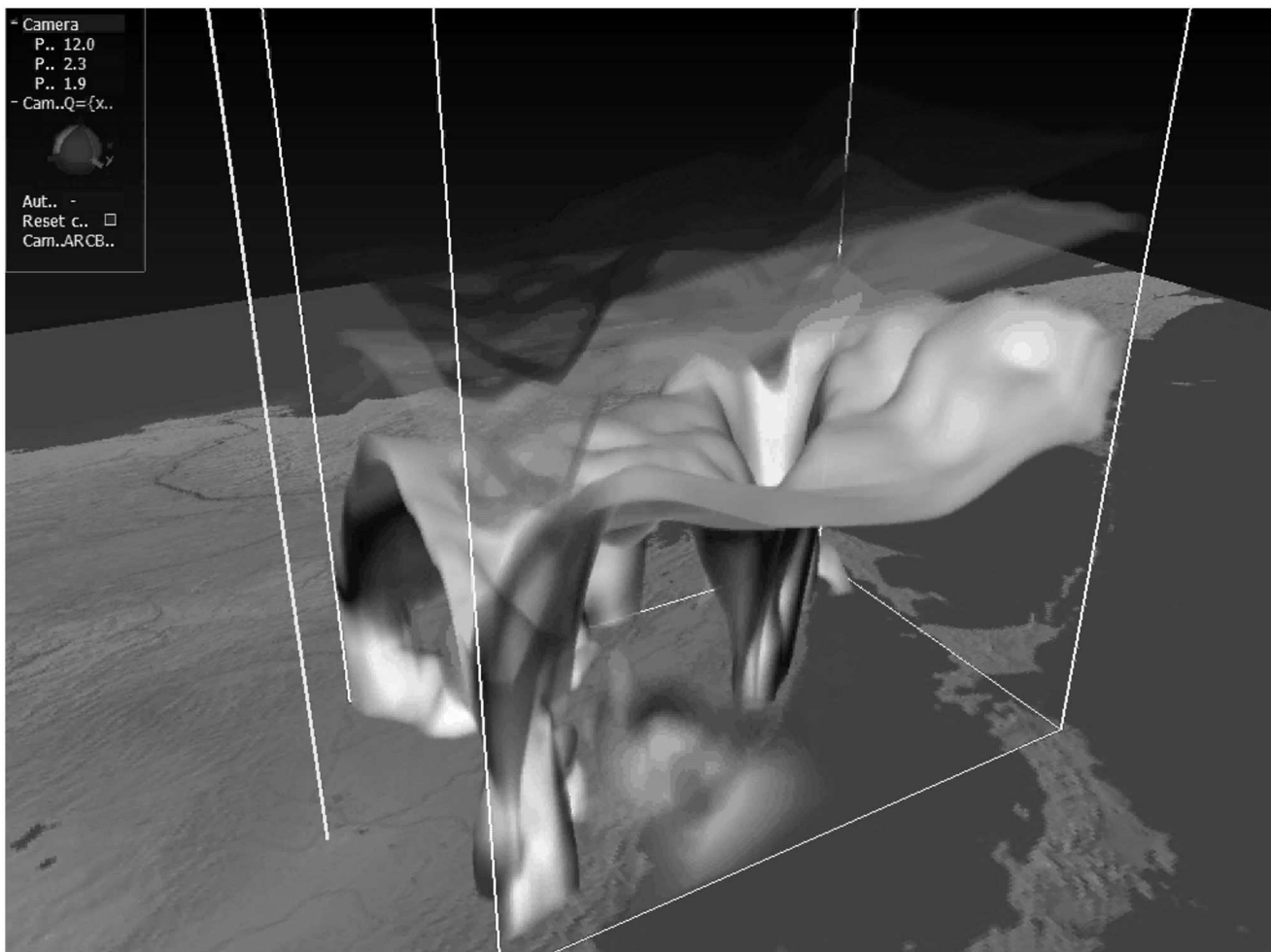


Рис. 3. Визуализация температурных слоев, полученных математической мезомасштабной прогнозной моделью WRF-ARW.

Наряду с функциональными возможностями, пользователю предлагается простая настройка распределенной системы на нескольких ПК, объединенных сетью. Также разработан простой в освоении SDK для создания новых модулей (plug-in) визуализации, загрузки и фильтрации данных на языке C++.

Клиентское вычислительное приложение. Клиентское вычислительное приложение является интерфейсом, связывающим вычислительные модули с распределенной БД. Оно обеспечивает кэш (копии только необходимых данных из БД), занимается пересылкой данных и поддержанием их в актуальном состоянии. Для обеспечения темпоральной анимации вычислительный конвейер работает только для ключевых моментов времени.

Пакет с заданиями, рассылаемый диспетчером для вычислительных узлов, содержит описание конвейера со всеми параметрами вычислительных модулей, заданными пользователем в интерфейсном приложении, количество темпоральных срезов (измерений) и порядковый номер, присвоенный узлу. Каждый вычислительный узел, получив пакет, выполняет полный проход по конвейеру для своей части данных и отправляет результат в базу данных. Зная свой порядковый номер и количество темпоральных срезов, узел обрабатывает только те, для которых остаток от деления номера среза на количество узлов дает порядковый номер этого узла.

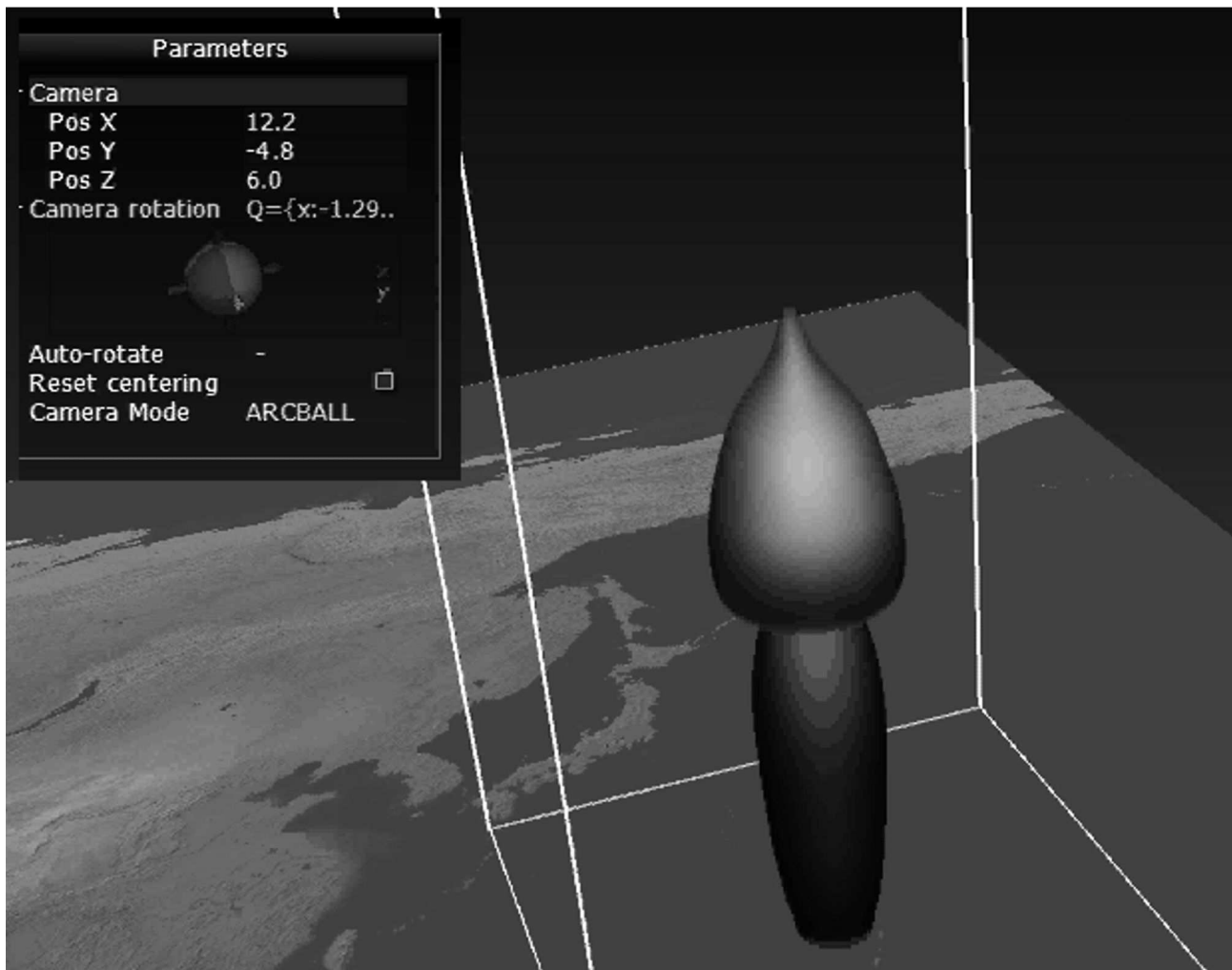


Рис. 4. Визуализация температурных аномалий в ядре тропического циклона (грибовидная структура).

При этом нумерация узлов и срезов начинается с нуля. Пример: есть температурные профили, получаемые раз в день на протяжении 5 дней, т.е. 5 температурных срезов, также известно, что вычислительных узлов 3, тогда нулевой узел обрабатывает срезы 0 и 3, первый узел – 1 и 4, второй узел обрабатывает 2 срез. Так как в типичной задаче количество срезов исчисляется десятками, а количество узлов значительно меньше, то эффективность схемы не страдает.

Заключение

Вычислительные эксперименты проводились с данными тропических циклонов и спутниковых профилей с применением разных алгоритмов визуализации: объемный рендеринг на шейдерах и 3D текстурах, визуализация изоповерхностей методом «марширующих кубиков», визуализация движения центра ТЦ. Использовались вычислительные модули конвертации профилей спутникового мониторинга в регулярную решетку, восполнение недостающих данных и расчет аномалий в центрах тропического циклона.

Автоматическая загрузка данных о спутниковых профилях осуществлялась по протоколу FTP из центра спутникового мониторинга ИАПУ ДВО РАН.

Также были проведены эксперименты с данными прогнозной WRF-модели,

соответствующими территории Приморского края и временному периоду с 25.04.2012 по 29.04.2012.

Эксперименты подтвердили эффективность системы построенной по предложенной технологии. Разработанная система визуализации позволила полностью в автоматическом режиме делать запросы к удаленному хранилищу данных и визуализировать полученные данные в рамках одного пользовательского интерфейса.

В перспективе планируется расширение множества модулей загрузки синоптических данных из мест общего пользования (через Интернет), создание модулей визуализации узкоспециализированных данных (наложение погодных карт, карт ледяного покрова и т.д.), разработка механизма расширения измерительного инструментария с помощью модулей по технологии plug-in.

ЛИТЕРАТУРА

1. Бобков В.А., Мельман С.В. и др. Программный комплекс визуализации синоптических данных // Информатика и системы управления. – 2014. – № 1(39). – С.141-150.
2. David Camp, Christoph Garth, Hank Childs, et. Streamline Integration Using MPI-Hybrid Parallelism on a Large Multicore Architecture // IEEE Transactions on visualization and computer graphics. – 2011. – Vol. 17, NO. 11. – P.1702-1713.
3. Коньшин И.Н., Сушко Г.Б., Харченко С.А. Трехуровневая MPI+TBV+CUDA параллельная реализация блочного итерационного алгоритма решения СЛАУ для мелкоблочных неструктурированных разреженных матриц // Научный сервис в сети Интернет: поиск новых решений: Труды Междунар. суперкомпьютерной конф. – М.: Изд-во МГУ. – 2012. – С.522-528.
4. Кривов М.А., Притула М.Н., и др. Оптимизация приложений для гетерогенных архитектур. Проблемы и варианты решения // Информационные технологии и вычислительные системы. – 2012. – № 3.
5. Чусов А.А., Стаценко Л.Г. Разработка адаптируемых распределенных систем параллельного моделирования, анализа и визуализации физических полей // Монография [Электронный ресурс] / Инженерная школа ДФУ. – Владивосток, 2014.
6. Бобков В. А., Мельман С. В., Май В. П. Визуализация динамики циклонов // Информационные технологии. – 2012. – №2. – С.49-54.

Статья представлена к публикации членом редколлегии А.И. Абакумовым.

E-mail:

Мельман Сергей Владимирович – 8(423)2754424, melman.sv@dvfu.ru;

Бобков Валерий Александрович – 8(423)2313776, bobkov@iacp.dvo.ru;

Черкашин Александр Сергеевич – alexandr@cherkashin.ws.