Моделирование распределенной системы сбора, передачи и обработки данных для крупных научных проектов (мегапроект НИКА)[1]

В.В Кореньков, А.В. Нечаевский, В.В. Трофимов

Аннотация. В работе обоснована необходимость создания имитационной модели системы хранения и обработки данных ускорительного комплекса НИКА. В качестве платформы для создания модели выбрана система GridSim. В работе описан подход к моделированию системы хранения данных dCache и каналов передачи. На простом примере показаны возможности использования модели.

Ключевые слова: грид-технологии, грид-инфраструктуры, система хранения данных, оптимизация, моделирование, исследование, разработки, dCache, Tier1, НИКА, грид.

Введение

В Объединенном институте ядерных исследований создается ускорительный комплекс НИКА. Комплекс НИКА представляет собой ускоритель тяжелых ионов НИКА и установку МПД (Multi Purpose Detector), объединяющую детекторы для изучения ядерной материи в горячем и плотном состоянии, которое возникает при столкновении ускоренных тяжелых ионов. МПД является источником данных с интенсивностью потока десятки петабайт в год.

Ожидаемая интенсивность потока данных настолько велика, что массивы данных характеризуются как сверхбольшие. Для обработки таких потоков данных используются распределенные системы коллективного пользования, построенные на грид-технологиях.

Для оптимизации структуры будущего комплекса обработки данных необходимо определить его основные параметры, структуру и проверить предлагаемые технические решения с помощью моделирования. Для этих целей на базе пакета моделирования GridSim, создана имитационная модель грид-сайта. Модель позволяет оценить требуемые ресурсы на этапе офлайн обработки потока данных.

1. Схема обработки данных ускорительного комплекса НИКА

Хранение и использование экспериментальных данных в современных экспериментах физики высоких энергий является актуальной проблемой. Объем получаемых и обрабатываемых данных исключает возможность ее хранения и использования не только на одном кластере, но и в пределах одной организации, поэтому на первый план выходит создание распределенной системы хранения и обработки данных для эксперимента.

Для эксперимента МПД на НИКА предполагается, что поток данных будет иметь следующие параметры:

[1] Работа частично выполнена в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы» (Гос.контракт №07.524.12.4008).

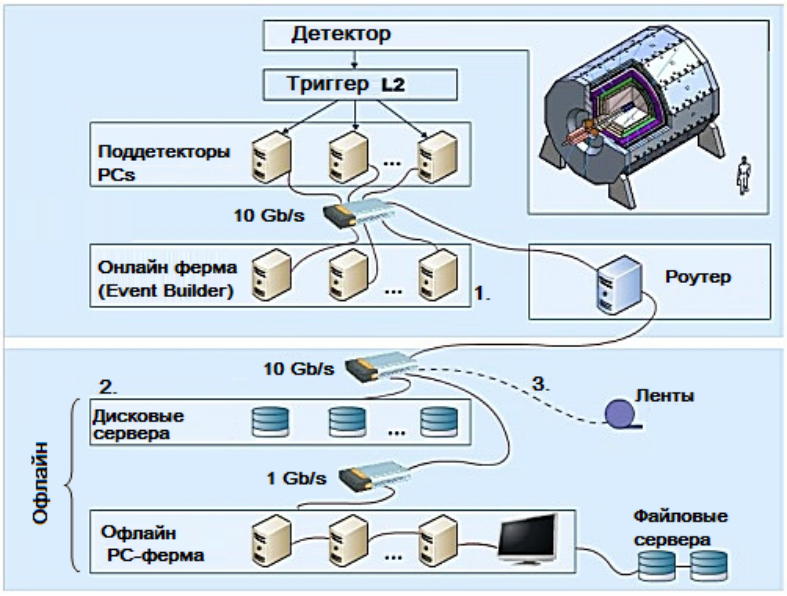

Рис. 1. Схема обработки физических данных ускорительного комплекса НИКА

- высокая скорость набора событий (до 6 КГц),
- в центральном столкновении Au-Au при энергиях НИКА образуется до 1000 заряженных частиц,
- размер файла с первоначальной моделируемой информацией с детекторов для одного события занимает около 0,45 МБ.

Схема получения и обработки данных представлена на Рис. 1. Данные, идущие от персональных компьютеров поддетекторов МПД, накапливаются специально предназначенными для сборки событий программами (Event Builder) компьютерной фермы в режиме онлайн. После формирования события в режиме офлайн через специально предназначенную для этой цели волоконно-оптическую линию связи со скоростью 10 Гб/с данные записываются на диск.

После триггера высокого уровня отобранные события записываются в RAW файлы (скорость записи один файл в 1 минуту сбора данных) и затем полностью восстанавливаются.

Прогнозируемое количество обрабатываемых событий при этом приблизительно $19 \cdot 10^9$. Принимая скорость передачи данных от датчиков 4.7 ГБ/с, общий объем исходных данных может быть оценен в 30 PB ежегодно, или 8.4 PB после сжатия. Эти оценки основаны на особенностях DAQ и подобных оценках, выполненных для эксперимента ALICE [1].

В качестве системы обработки физической информации в эксперименте НИКА предполагается использование грид. Грид (название по аналогии с электрическими сетями – electric power grid) – это компьютерная инфраструктура нового типа, обеспечивающая глобальную интеграцию информационных и вычислительных ресурсов. Суть инициативы грид состоит в создании набора стандартизированных служб для обеспечения надежного, совместимого, дешевого и безопасного доступа к географически распределенным высокотехнологичным информационным и вычислительным ресурсам – отдельным компьютерам, кластерам и суперкомпьютерным центрам, хранилищам информации, сетям, научному инструментарию и т.д. [2].

Эксперименты, в которых для обработки данных используется грид-инфраструктура или облачные вычисления, имеют некоторые общие черты: большие потоки данных, длительный цикл проектирования и строительства, длительный период эксплуатации. Так компьютерная инфраструктура для эксперимента ALICE представляет собой иерархическую грид-структуру с компьютерными центрами класса Tier 0/1/2. Функциональные различия уровней иерархической модели представлены в Табл. 1 [3]. Для хранения и обработки данных с эксперимента PANDA [4] также предполагается использование грид.

Табл. 1. Уровни иерархической модели и их функции

Tier0	первичная реконструкция событий, калибровка, хранение копий полных баз данных
Tier1	полная реконструкция событий, хранение актуальных баз данных по событиям, создание и хранение наборов анализируемых событий, моделирование, анализ
Tier2	репликация и хранение наборов анализируемых событий, моделирование, анализ

Проектирование грид-структур больших масштабов, подразумевает не только привлечение специалистов, обладающих уникальными навыками, но и применение инструментов для моделирования. При создании распределенной системы требуется принять решения по архитектуре инфраструктуры, количеству ресурсных центров, объему требуемых ресурсов. Кроме того, необходимо обеспечить достаточную пропускную способность, решить проблемы сохранности данных, обеспечить распределение ресурсов между различными группами пользователей, выбрать алгоритмы обработки и запуска задач и многое другое. Для решения этих вопросов, а также обоснования решений требуется создание имитационной модели обработки данных эксперимента. Возникает необходимость создания имитационной модели, которая бы удовлетворяла всем условиям.

Актуальность темы обуславливается тем, что на основе модели в дальнейшем могут быть обоснованы рекомендации и техническое задание на разработку компьютерной инфраструктуры, рассмотрены различные варианты организации хранения данных эксперимента.

2. Исследование методов и средств имитационного моделирования грид

На сегодняшний день существуют различные инструменты моделирования грид-систем [5]. Проект GridSim разрабатывается группой исследователей в лаборатории по изучению облачных и распределенных вычислений отдела информатики и компьютерных вычислений в университете Мельбурна, Австралия. Пакет моделирования GridSim неоднократно применялся [6] для моделирования грид-структур и планировщиков.

GridSim – это библиотека классов, предназначенных для построения модели грид-системы. Она в свою очередь построена на стандартной библиотеке SimJava, с помощью которой можно моделировать поток дискретных событий во времени. Приложение создается расширением классов GridSim и объединением их в программу, которая моделирует обработку потока заданий грид-структурой, обладающей определенными ресурсами и с заданной дисциплиной их резервирования и использования. В сравнении с другими пакетами моделирования грид GridSim обладает рядом преимуществ. Основные преимущества представлены в Табл.2 [7]. С помощью GridSim можно проводить воспроизводимые эксперименты, которые сложно реализовать в настоящем окружении динамических грид-систем.

Проанализировав ряд систем, для разработки имитационной модели была выбрана платформа GridSim.

Табл. 2. Функции и свойства симуляторов грид

Функция	GridSim	OptorSim	Monarc	ChicSim	SimGrid	MicroGrid
Репликация данных	Да	Да	Да	Да	Нет	Нет
Издержки записи/чтения диска	Да	Нет	Да	Нет	Нет	Да
Комплексное фильтрование или запросы данных	Да	Нет	Нет	Нет	Нет	Нет
Планировка пользовательских задач	Да	Нет	Да	Да	Да	Да
Резервирование ЦПУ	Да	Нет	Нет	Нет	Нет	Нет
Симуляция нагрузки	Да	Нет	Нет	Да	Нет	Нет
Дифференцированное QoS сети	Да	Нет	Нет	Нет	Нет	Нет
Генерация фонового сетевого трафика	Да	Да	Нет	Нет	Да	Да

3. Моделирование офлайн-уровня обработки физических данных эксперимента НИКА

В качестве примера грид-структуры уровня T1 мы рассмотрим офлайн уровень обработки физических данных ускорительного комплекса НИКА. Для эффективной работы грид-сайта, проведения исследований по оптимизации нагрузки, разработки и тестирования новых алгоритмов с точки зрения скорости достижения результата необходимо использовать средства моделирования грид-систем. При создании модели предполагается, что основой для построения системы хранения данных будет dCache [8]. Модель сайта T1 строится на алгоритме обработки данных (Рис. 1).

1. Данные появляются с заданной частотой и записываются на локальные диски компьютеров. После перемещения данных на второй уровень диск очищается.

2. Данные перемещаются автоматически на второй уровень по каналам. В качестве носителей второго уровня используются пулы системы dCache, рассматриваемые в модели как единая память. При обработке данных предполагается, что вначале данные попадают в дисковый пул системы хранения, а затем по локальному протоколу передается на узлы обработки. Непосредственное монтирование директории на рабочих узлах не используется.

3. Для долгосрочного хранения данных используется ленточный робот. Копии файлов автоматически создаются на лентах, после чего файлы удаляются с дисковых пулов.

Отличительная особенность конфигурации dCache – наличие не менее двух уровней хранения: жесткие диски и ленточный накопитель. Под ленточным накопителем подразумеваются автоматизированные библиотеки, оснащенные роботизированным загрузочным механизмом и стойкой на несколько картриджей (ленты). Объем такой библиотеки (Q) можно определить простейшими вычислениями, исходными данными для которых будут производительность установки (p), время ее работы (T) и емкость накопителей (c).

$$Q=p\cdot T/c$$

Другие вопросы создания грид-сайта требуют более тщательного анализа и выбора приемлемого варианта. Таким образом, перед разработчиками системы встают следующие вопросы:

– определение необходимого количества драйвов,

– способы группировки файлов на лентах,

– политика записи файлов.

Стоит отметить ряд ключевых особенностей GridSim, которые потребовали доработки из-за несоответствия требованиям модели:

– создавать файлы может только пользователь;

– все объекты моделирования объединены в сеть при помощи каналов передачи данных;

– пользователь может копировать (создавать) только один файл единовременно.

Для решения этих вопросов потребовалось расширение существующих классов и добавление новых объектов. Так в систему добавлены следующие объекты (Рис. 2):

– Drive - драйв магнитофона;

– Arm - рука робота;

– ReelArchive - архив картриджей;

– Reel - картридж;

Набор этих классов позволяет моделировать все процессы, происходящие с копией файла на лентах: загрузку и выгрузку ленты манипулятором, монтирование на драйве, поиск файла на ленте и его чтение/запись.

Задача моделирования сетевой инфраструктуры в библиотеке GridSim решена с помощью классов Router, Link, NetPacket и некоторых других классов. Этот набор средств позволяет моделировать прохождение пакетов по сети. Пользователю предоставляется возможность встраивать свои планировщики пакетов в исходную модель. Такой подход обеспечивает высокую точность моделирования. Его недостатком применительно к задаче моделирования T1 является избыточность – вопросы маршрутизации, столкновений пакетов, влияние фоновой загрузки каналов в данной модели не рассматриваются и, следовательно, уровень детализации до пакета представляется избыточным. В нашем случае, интерес представляет только изменение нагрузки на отдельные компоненты сети.

Исходя из вышеизложенного потребовалось дополнить GridSim следующим механизмом. Вводится понятие *операция передачи данных.* Под этим подразумевается запись/чтение части или целого файла экспериментальных данных. В

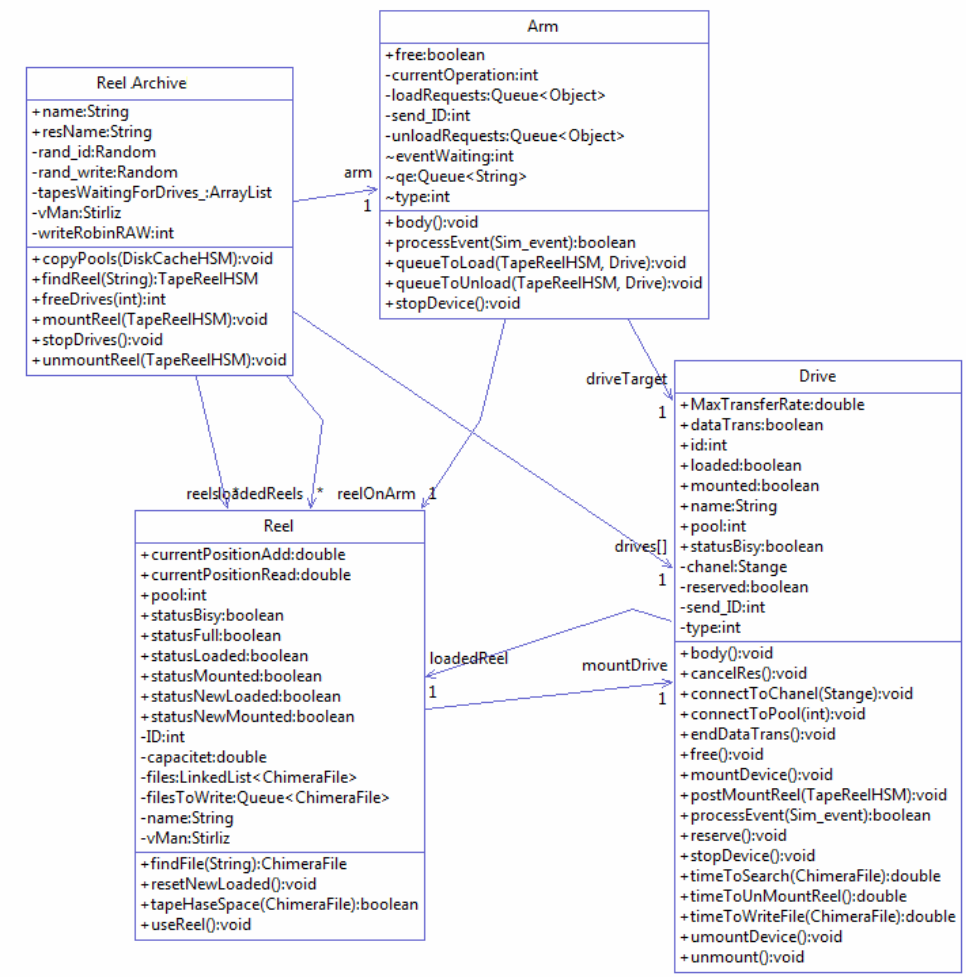

Рис. 2. Описание новых классов в модели

этом случае ввод и вывод служебной и диагностической информации считается пренебрежимо малым. Операция рассматривается как атомарная, т.е. начинается методом «начать операцию». Параметрами метода являются: устройство 1 – источник данных, устройство 2 – получатель и список всех промежуточных устройств, которые необходимо пройти от источника до получателя. Элемент сети в системе описывается классом Stange. Взаимодействие классов, описывающих сеть, изображено на Рис. 3.

Результаты моделирования доступны пользователю в виде таблиц и графиков. Для этой цели используются классы генератора лога и визуального отображения результатов:

– Info - описание вычислительной структуры и потока заданий;

– Reporter - генератор лога;

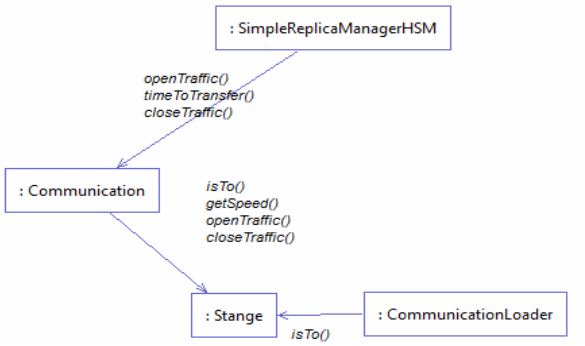

Рис. 3. Взаимодействие классов, описывающих сеть

– парсер лога;

– объект визуального отображения результатов и другие.

Ниже приведем пример задачи, которая возникает при проектировании системы сбора и хранения данных.

4. Практическое применение модели

С помощью системы моделирования можно исследовать прохождение набора заданий и передачу файлов через грид-структуру с заданной пользователем топологией и параметрами центров обработки. Модель позволяет получить оценку временных параметров обработки потока заданий при заданной пользователем дисциплине распределения ресурсов между заданиями и структурой очередей к центрам обработки.

Моделирование дает ответы на вопросы:

а) какие вычислительные ресурсы требуются для обработки данных;

б) как должны быть связаны между собой центры обработки;

в) какой должен быть уровень сжатия данных;

г) какая должны быть конфигурация роботизированной библиотеки;

д) хватит ли ресурсов на обработку потока данных и предоставление данных пользователям.

Проиллюстрировать применение упомянутых выше классов можно на примере моделирования процесса обработки данных с одновременной записью ее на ленты. Задача проектировщика – определить необходимое количество драйвов библиотеки. При этом исследуются два вопроса: какое количество драйвов библиотеки необходимо для того, чтобы записать весь поток “сырых” (RAW) данных с детекторов эксперимента, насколько при этом процесс обработки данных (поток заданий от пользователей) будет мешать записи, если обработка потребует загрузки файлов с лент на диски.

Допустим, что в нашем распоряжении имеется библиотека, количество драйвов в библиотеке фиксировано и равно пяти. Это существенно меньше необходимого, но достаточно для иллюстрации возможностей модели. Когда для обслуживания поступающих на сайт заданий и записи RAW данных используются одни и те же пулы (драйвы), процесс начинает вести себя хаотично, многократно монтируя и размонтируя ленты для записи даже при незначительных загрузках. Для того чтобы избежать этой ситуации, в рассматриваемой модели пулы лент разделены на принимающие данные (RAW) и обслуживающие поток заданий (DLT). Возникает вопрос, каким образом распределить драйвы между двумя пулами при фиксированных параметрах потока заданий. Мы предполагаем, что файлы запрашиваются случайным образом.

Моделируемая система – двухуровневая. На первом уровне находится дисковый массив, на втором уровне ленточный накопитель. В существующей модели скорость записи и чтения с дискового массива не зависит от загрузки. Параметры драйвов и робота соответствуют параметрам планируемым к установке устройств (Табл. 3). Количество драйвов в роботе фиксировано и есть только одна «рука» загружающая файлы в драйв.

Результаты моделирования приведены в Табл. 4. С помощью модели исследовались следующие характеристики:

– время выполнения – астрономическое время выполнения потока заданий, которое из общих соображений должно уменьшаться с увеличением количества драйвов;

– длина очереди – максимальная длина очереди на запись RAW данных на ленту.

Моделирование показывает, что при заданном темпе сбора данных для записи должно быть выделено не менее двух драйвов. С другой стороны, для обработки потока заданий должно быть выделено не менее двух драйвов для чтения накопленной информации. Если за критерий оптимальности принять минимальное астрономическое время выполнения потока за-

Табл. 3. Параметры для моделирования ленточной библиотеки

Параметр	Значение
Время монтирования/размонтирования (с)	22
Скорость поиска (с)	300
Скорость чтения/записи (с)	120
Скорость перемотки (с)	1000
Время загрузки/разгрузки картриджа в драйв (с)	100
Размер файла (МБ)	6000

Табл. 4. Результаты моделирования

Эксперимент	Драйвов RAW	Драйвов DLT	Время выполнения	Длина очереди
1	1	4	28959	13
2	2	3	28703	1
3	3	2	28814	1
4	4	1	59275	1

даний, то оптимальным можно считать распределение драйвов по варианту номер 2.

Этот пример иллюстрирует один из вариантов использования программы. Такие исследования могут быть проведены с использованием аналитических моделей теории массового обслуживания, однако добавление простейших условий группировки заданий и файлов, значительно усложняет аналитические модели, тогда как для имитационной модели изменения сводятся к нескольким строчками программного кода.

Заключение

Созданная система моделирования позволяет проводить разнообразные эксперименты с исследуемым объектом, не прибегая к физической реализации, что позволяет предсказать и предотвратить большое число неожиданных ситуаций в процессе эксплуатации, которые могли бы привести к неоправданным затратам, потере данных, а, возможно, и к повреждению дорогостоящего оборудования. В процессе моделирования можно определить минимально необходимое оборудование, обеспечивающее потребности передачи, обработки и хранения данных, оценить необходимый запас производительности оборудования, обеспечивающего возможное увеличение производственных потребностей, выбрать несколько вариантов оборудования с учетом текущих потребностей и перспективы развития в будущем, провести проверку работы системы, выявить ее «узкие» места и т.д.

Применение системы моделирования позволит определить параметры системы обработки данных ускорительного комплекса НИКА на этапе технического проектирования.

Дальнейшее развитие системы предполагает внесение дополнений с целью создания модели грид-сайта уровня T1 с использованием 2-х и 3-х уровней dCache. Для моделирования предполагается использовать оригинальный алгоритм назначения пулов dCache и оригинальные данные по потокам. Также необходимо провести полномасштабные испытания модели с целью выявления ошибок и создания базы сценариев моделирования.

Важное значение разработанной системы моделирования связано с созданием в Объединенном институте ядерных исследований автоматизированной системы обработки и хранения данных (АСОД) уровня T1 для эксперимента CMS на Большом адроном коллайдере и предназначенной для работы в составе глобальной грид-системы для обработки данных (WLCG). АСОД нацелена на проведение полного цикла обработки физической информации, получаемой в ходе проведения эксперимента, обеспечения работ по моделированию физических процессов, защищенного хранения и приема/передачи данных в другие центры WLCG. Основной системой хранения данных в АСОД является dCache. Очевидно, что в процессе длительного (10 лет и более) функционирования центра будет необходимо оперативно масштабировать систему хранения и повышать эффективность использования ленточного робота в системе dCache, без остановки работы всего комплекса. В этом процессе предварительное моделирование работы системы хранения будет являться необходимым инструментом.

Результаты работы могут быть рекомендованы для использования при проектировании грид-системы для сбора, передачи, обработки и хранения данных с мегаустановок или других аналогичных установок, генерирующих большие объемы данных.

Литература

1. P. Cortese et al «ALICE Technical Design Reort of the Computing.» // CERN/LHCC 2005-018, ALICE TDR 12, 2005.
2. В.В. Кореньков «Грид технологии: статус и перспективы» // «Вестник Международной академии наук. Русская секция», ISSN 2221-7479, 1, 2010, С. 41-44.
3. В.А. Ильин, В.В. Кореньков, А.А. Солдатов «Российский сегмент глобальной инфраструктуры LCG» // Открытые системы, ISSN:1028-7493, Изд: Открытые системы, 1, 2003, С. 56-60.
4. Веб-портал проекта PANDA: http://www-panda.gsi.de/
5. А.В. Нечаевский, В.В. Кореньков «Пакеты моделирования DataGrid» // Системный анализ в науке и образовании, ISSN: 2071-9612, Изд: Международный университет природы, общества и человека «Дубна», 1, 2009.
6. Веб-портал проекта GridSim: http://www.gridbus.org/gridsim/
7. Sulistio A., Cibej U., Venugopal S., Robic B., Buyya R. «A Toolkit for Modelling and Simulating Data Grids: An Extension to GridSim» // Concurrency and Computation: Practice and Experience (CCPE), Online ISSN: 1532-0634, Printed ISSN: 1532-0626, 20(13): 1591-1609, Wiley Press, New York, USA, Sep. 2008.
8. Веб-портал проекта dCache: http://www.dcache.org/

Кореньков Владимир Васильевич. Директор Лаборатории информационных технологий Объединенного института ядерных исследований (ОИЯИ), заведующий кафедрой «Распределенных информационно-вычислительных систем» Международного университета природы, общества и человека «Дубна». Окончил Московский государственный университет в 1976 году. Доктор технических наук, старший научный сотрудник. Автор более 250 печатных работ. Область научных интересов: распределенные и параллельные вычисления, грид-технологии, сети, базы данных и распределенные системы хранения сверхбольших объемов информации, корпоративные информационные системы. E-mail: korenkov@cv.jinr.ru

Нечаевский Андрей Васильевич. Инженер-программист Лаборатории информационных технологий Объединенного института ядерных исследований (ОИЯИ). Окончил Международный университет природы, общества и человека «Дубна» в 2006 году. Автор 10 печатных работ. Область научных интересов: грид-технологии, хранение и обработка больших массивов данных, имитационное моделирование грид-систем. E-mail: Andrey.Nechaevskiy@gmail.com

Трофимов Владимир Валентинович. Ведущий программист Лаборатории информационных технологий Объединенного института ядерных исследований (ОИЯИ). Окончил Уральский политехнический институт им. С.М.Кирова в 1978 году. Автор 15 печатных работ. Область научных интересов: хранение и обработка больших массивов данных, автоматизация измерений, автоматизированные системы управления технологическим процессом.
E-mail: trofimov@jinr.ru