«ФИЗИКА ЭЛЕМЕНТАРНЫХ ЧАСТИЦ И АТОМНОГО ЯДРА»
2001, ТОМ 32, ВЫП. 6

УДК 681.3.01

КОНЦЕПЦИЯ GRID И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ЭРУ LHC

В. В. Кореньков, Е. А. Тихоненко

Объединенный институт ядерных исследований, Дубна

«ФИЗИКА ЭЛЕМЕНТАРНЫХ ЧАСТИЦ И АТОМНОГО ЯДРА»
2001, ТОМ 32, ВЫП. 6

УДК 681.3.01

КОНЦЕПЦИЯ GRID И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ЭРУ LHC

В. В. Кореньков, Е. А. Тихоненко

Объединенный институт ядерных исследований, Дубна

Для полноценного участия в современных крупных физических экспериментах, куда вовлечены группы ученых из многих научных центров разных стран мира, одним из необходимых условий является надлежащая поддержка компьютинга в организациях-участницах этих экспериментов. Организация компьютинга включает в себя предоставление необходимых вычислительных ресурсов и ресурсов памяти, создание унифицированной программной среды, обеспечение надежной и быстрой сетевой связи с внешним миром, а также информационный сервис. Тенденции развития компьютинга современных крупных экспериментов привели к необходимости освоения и использования новых технологий, а именно технологий, относящихся к зарождающимся новым компьютерным инфраструктурам «grid». В обзоре прослеживаются особенности развития компьютинга для экспериментов, планируемых на LHC, в том числе в контексте использования и развития grid-технологий, а также освещаются проблемы поддержки компьютинга этих экспериментов в России и ОИЯИ: текущее состояние дел и перспективы развития.

The proper computing support in all scientific organizations which are members of the world-wide distributed next generation physical experiments is a necessary component of a full-range participation in these experiments. This includes CPU and storage facilities, fast external and local network communications and information support. The trends of computing development lead to a necessity of the usage of new modern technologies, in particular the technologies concerning a conception of «grid». The trends of computing support and development for the experiments which are planned at LHC are described. A current state and the perspectives of participation of Russian and JINR physicists in the LHC experiments at the running phase are considered.

ВВЕДЕНИЕ

К 2005 г. в CERN (Швейцария) планируется запуск крупнейшего в мире ускорительного комплекса частиц LHC (Large Hadron Collider) [1]. На этом ускорителе в течение 15–20 лет на 4 крупных физических установках (CMS, ATLAS, ALICE и LHCb) будут получены огромные объемы данных. Участниками этих экспериментов являются несколько тысяч ученых из более чем 30 стран мира. Предполагается, что после 2010 г. объем физических данных, полученных на LHC, будет составлять сотни Пбайт информации (1 Пбайт $= 10^{15}$ байт). Обработка данных такого масштаба является беспрецедентной и требует специальных усилий для создания средств хранения данных и последующего доступа к ним. Как ожидается, это приведет к кардинальным изменениям в информационно-сетевых технологиях. Стоит напомнить, что

исследования, ведущиеся в CERN, уже не единожды оказали решающее влияние на развитие информационно-сетевых технологий: с запуском коллайдера LEP в 1989 г., согласно потребностям сообщества физики высоких энергий, был дан мощный импульс к развитию сетевых коммуникаций в Европе и использованию протокола TCP/IP, а вскоре сотрудник CERN Тим Бернерс-Ли предложил, как осуществить доступ в Интернет простым нажатием на кнопку «мыши» — это было рождение Всемирной паутины (WWW), без которой теперь невозможно представить себе мир компьютеров. О стадии эксплуатации физических установок на LHC с точки зрения компьютинга* говорится исключительно как об «эре LHC», поскольку задачи организации этого компьютинга действительно беспрецедентны, ибо потребуется:

• отделять полезные физические сигналы в экспериментах при крайне тяжелых фоновых условиях,

• обеспечивать быстрый и прозрачный доступ к базам данных колоссального объема,

• обеспечивать прозрачный доступ к географически отдаленным вычислительным ресурсам,

• создать протяженную надежную сетевую инфрастуктуру в гетерогенной среде.

1. КОНЦЕПЦИЯ GRID И ПРОБЛЕМЫ ОРГАНИЗАЦИИ КОМПЬЮТИНГА ДЛЯ LHC

Десять последних лет исследований и достижений в области метакомпьютинга и сетевых технологий создали определенную базу для глобального объединения вычислений и высокоскоростных сетей связи. Концепция grid была сформулирована как некоторое обобщение современных тенденций развития сетевых и информационно-вычислительных технологий, с учетом все возрастающих потребностей в компьютерных ресурсах во всем мире [2–6].

Grid — это зарождающаяся инфраструктура, которая может кардинально изменить наши привычные представления о компьютинге. Предполагается, что grid-структуры смогут объединить региональные и национальные вычислительные компьютерные инфраструктуры для создания всеобщего ресурса вычислительной мощности. Само название «grid» было выбрано по аналогии с электрическими сетями (power grid), в которых обеспечивается всеобщий доступ к электрической мощности и которые оказали огромное влияние на человеческие возможности и общество. Как и в электрических сетях, предполагается интегрировать большой объем географически удаленных ресурсов.

*Под «компьютингом» в данном контексте понимается применение средств вычислительной техники и средств связи для целей физических исследований.

Во время физических экспериментов на LHC экспериментальные данные будут записываться с очень высокой скоростью (20 Мбайт/с ÷ 1,5 Гбайт/с [7]) и затем храниться без изменений (так называемые «сырые» экспериментальные данные) непосредственно в CERN. Однако, вследствие географической отдаленности участников — тысяч физиков из университетов и институтов разных стран мира, возникнет необходимость хранить часть данных в распределенных, так называемых региональных центрах, а также в некоторых институтах и университетах, что позволит использовать вычислительные мощности и средства хранения данных этих региональных центров, а не только мощности CERN. Таким образом, будут созданы условия для анализа и обработки данных не только в CERN, но и во всех организациях, участвующих в данных работах, так что не будет необходимости обращаться к данным, расположенным в хранилищах CERN.

Данная модель организации исследований сочетает в себе два аспекта современных grid-технологий: вычислительный и информационный (Computational Grid и Data Grid). Подобные структуры могут найти свое применение не только в физике высоких энергий, но также, например, в биоинформатике, системе наблюдений за Землей, экологии, метеорологии.

Практическая реализация таких распределенных центров фактически приводит к созданию определенных grid-структур и включает в себя разработку промежуточного программного обеспечения (middleware), вычислительных структур (computing fabric), организации работы с данными, систем отладки (testbed) и приложений для конкретной научной сферы. Middleware будет обеспечивать эффективные, стандартные и прозрачные методы доступа к данным для осуществления кэширования данных*, репликации** и миграции (перемещения) файлов в гетерогенной среде. Таким образом, необходимо обеспечить управление универсальным пространством имен, эффективный перенос данных между сайтами, синхронизацию удаленных копий, доступ и кэширование данных на глобальном уровне, а также интерфейс к системам управления массовой памятью.

Математическое обеспечение для grid-структур характеризуется многоуровневостью. Рисунок 1 дает общее представление о «слоях» grid-структур с точки зрения программного обеспечения.

Для моделирования распределенных вычислительных центров для LHC был создан специальный международный проект MONARC (Models of Networked Analysis at Regional Centres for LHC Experiments) [8], краткое описание которого мы приводим далее в данном обзоре. Также мы считаем необходимым остановиться на описании нескольких крупных проектов, в которых в

*Кэш — быстродействующая буферная память.

**Репликация — тиражирование точных копий данных.

Рис. 1. Grid-архитектура с точки зрения программного обеспечения

настоящее время развивается и реализуется концепция grid. Эти проекты во многом взаимосвязаны между собой, а за основу математического обеспечения промежуточного слоя в большинстве проектов выбран набор инструментальных средств Globus [13], структуру и возможности которого мы также сочли целесообразным детализировать в заключение данного раздела.

1.1. Проект MONARC. Основной мотивацией для распределенной организации компьютерных ресурсов для LHC является задача максимизировать интеллектуальный вклад физиков всего мира: т. е. для участия в исследованиях присутствие ученых непосредственно в CERN не должно быть обязательным. Кроме того, по многим причинам сконцентрировать все ресурсы в одной организации весьма сложно.

Для LHC в рамках проекта MONARC [8] было осуществлено моделирование распределенных вычислительных центров. В проекте MONARC ставились следующие цели и задачи:

- моделирование компьютинга для LHC;
- разработка базовых моделей компьютинга, включая стратегию, приоритеты и политику, для эффективного анализа данных международными коллаборациями;

Таблица 1. Требования к региональным центрам для LHC нулевого и первого уровней*

Номер уровня	Производительность (SPECint95)	Дисковая память, Тбайт	Емкость роботосистем	Сетевая связь, Мбит/с
Tier 0	600K	560	2,5 Пбайт	от 622
Tier 1	200K	200	500 Тбайт	от 155

*SPECint95 — некоторая интегральная оценка производительности процессора; так, например, производительность процессора Intel PIII-500 МГц оценивается как 20,6 SPECint95 [10].

Таблица 2. Оценка требуемых в CERN ресурсов для экспериментов на LHC к 2006 г.

Эксперимент	ALICE	ATLAS	CMS	LHCb	Суммарный ресурс
Производительность (SPECint95)	600 000	420 000	520 000	220 000	1 760 000
Количество CPU	3 000	3 000	3 000	1 500	10 500
Дисковая память, Тбайт	800	750	650	450	2650
Емкость роботосистем, Пбайт	3,7	3,0	1,8	0,6	9,1
Скорость ввода/вывода, Гбайт/с					
с дисками	100	100	100	40	340
с лентами	1,2	0,8	0,8	0,2	3,0

• изучение и обобщение базовых требований по вычислительным и сетевым ресурсам и управлению данными;

• обеспечение максимальной производительности каждого отдельного набора ресурсов, входящего в grid-структуру.

В результате для организации компьютинга для LHC была предложена некоторая иерархическая структура вычислительных центров [9], включающая в себя вычислительные центры пяти уровней (Tiers), каждый с разным объемом ресурсов и различными возможностями сетевого доступа. Нулевой уровень (Tier 0) — это главный центр в CERN, первый уровень (Tier 1) — крупные национальные центры ведущих стран мира, второй уровень (Tier 2) — региональные центры внутри отдельной страны и значительно менее крупные центры третьего (Tier 3) и четвертого (Tier 4) уровней. Требования по ресурсам для центров нулевого и первого уровней (согласно отчету по проекту MONARC [9]) представлены в табл. 1.

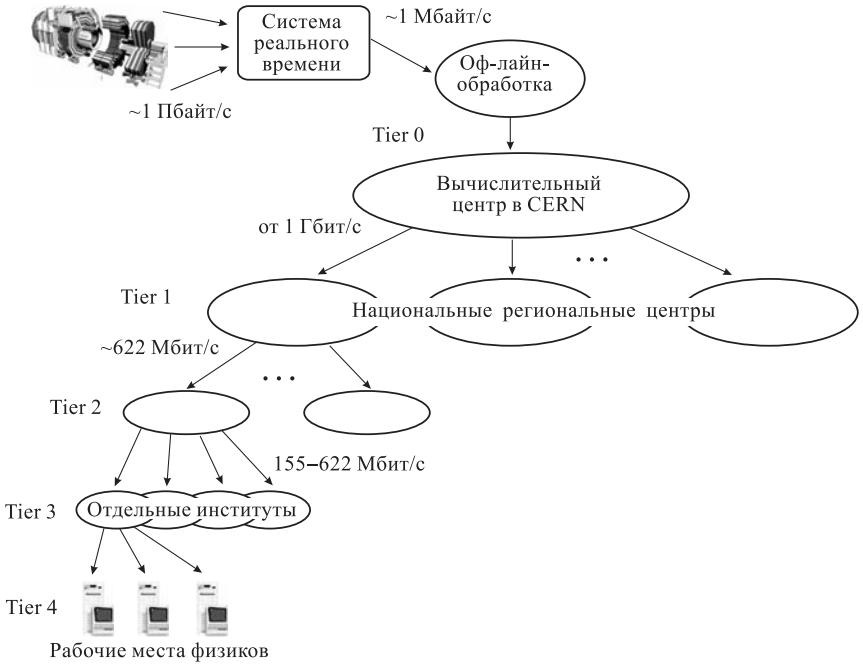

Рис. 2. Требуемые скорости коммуникаций в структуре региональных вычислительно-информационных центров для LHC

На основе результатов моделирования по проекту MONARC и с учетом потребностей всех четырех коллабораций на LHC были получены оценки ресурсов, которые потребуются в CERN к 2006 г. [11] (см. табл. 2).

Чтобы представить масштабы требуемого роста ресурсов, достаточно сказать, что на данный момент все четыре LHC-коллаборации располагают в CERN суммарным вычислительным ресурсом около 10 000 SPECint95 (примерно 1200 CPU) и дисковым пространством около 25 Тбайт (т. е. требуется увеличение производительности почти в 200 раз, а дискового пространства — в 100 раз).

Связь между отдельными компонентами структуры региональных центров должна быть очень высокоскоростной (см. рис. 2).

1.2. Проект EU Data Grid. В 2000 г. получил финансирование от Европейского сообщества проект EU Data Grid [12] для физики высоких энергий (организация компьютинга на LHC), биоинформатики и системы наблюдений за Землей. Общим во всех этих исследованиях является разделение данных с точки зрения информации и баз данных, распределенных по Европе и другим

континентам; основной целью является улучшение эффективности и скорости анализа данных посредством интеграции глобально-распределенных процессорной мощности и систем хранения данных, доступ к которым будет характеризоваться динамическим распределением данных по grid-инфрастуктуре, что предполагает управление репликацией и кэшированием.

Что касается будущих экспериментов на LHC, то, как и в текущих экспериментах по физике высоких энергий, для них можно выделить две основные категории в работе с физическими данными, а именно производство данных и анализ данных конечным пользователем. Производство данных включает в себя получение экспериментальных данных в CERN (эти данные будут поступать непосредственно с онлайновых систем сбора информации детекторов), распределенное моделирование физических событий*, реконструкцию событий и частичную переобработку данных. Анализ данных конечным пользователем включает в себя интерактивный анализ и удаленный анализ. Наиболее часто используемые данные потребуется хранить в памяти с наиболее быстрым доступом (например, на дисковых кэшах). В процессе анализа будут создаваться новые сложные объекты событий, которые сохранятся для дальнейшего анализа. Значительное количество времени будет затрачиваться на чтение объектов (т. е. на их поиск и чтение из кэша или с ленты). В силу независимости событий их обработка предполагает крупномодульный параллелизм, основанный на высокой степени свободы в управлении вводом/выводом, что позволит обрабатывать события параллельно на различных процессорах. Задачи управления данными будут состоять в организации стандартного и быстрого переноса файлов из одной системы хранения данных в другую. Немаловажными задачами также являются управление распределенным иерархическим кэшем, обеспечение проблем безопасности и прав доступа для пользователей.

Европейский проект EU Data Grid является комплексным проектом, в который вовлечено множество организаций, специалистов по программному обеспечению и ученых. Архитектура создаваемой grid-инфраструктуры должна быть достаточно простой, гибкой, масштабируемой, предполагающей быстрое создание прототипов и, конечно, отвечающей требованиям распределенного функционирования. Проект включает в себя несколько рабочих пакетов:

• создание приложений для всех отраслей (физики высоких энергий, биологии и наблюдения за Землей) по осуществлению прозрачного доступа к распределенным данным и предоставлению высокопроизводительных вычислительных ресурсов;

* «Событием» в физических установках на коллайдерах называются данные, регистрируемые установкой в результате неупругого столкновения частиц встречных пучков.

• управление рабочей загрузкой (распределенное планирование и управление ресурсами);

• управление данными (создание интегрированного инструментария и инфраструктуры промежуточного слоя для согласованного управления и разделения объемов информации петабайтного порядка в grid-среде с эффективным использованием ресурсов);

• мониторирование (как доступ к статусной информации и информации об ошибках в grid-среде);

• управление вычислительными структурами на кластерах, состоящих из тысяч вычислительных узлов;

• создание виртуальной частной сети, объединяющей вычислительные ресурсы и ресурсы данных, участвующие в отладке grid-инфрастуктуры;

• управление массовой памятью (создание глобального grid-интерфейса к существующим системам управления массовой памятью).

В качестве основы промежуточного программного обеспечения выбран набор инструментальных средств Globus [13].

1.3. Проект GriPhyN как междисциплинарная программа исследований. Проект GriPhyN [14] (Grid Physics Network) организован в США с целью создания так называемых виртуальных grid-структур (data-intensive virtual grid) для ряда отраслей науки, в которых будут накапливаться и обрабатываться колоссальные объемы научных данных (до нескольких десятков Пбайт).

В настоящее время идет подготовка нескольких научных экспериментов нового поколения. В этих экспериментах на новом уровне сложности будут исследоваться фундаментальные силы природы и структура Вселенной. К этим экспериментам относятся не только упомянутые выше эксперименты по физике высоких энергий, но и эксперименты с использованием интерферометров для регистрации гравитационных волн бинарных пульсаров, новых сверхзвезд и иных экзотических объектов (эксперимент LIGO), а также автоматизированная цифровая космическая съемка с очень высоким разрешением (более 1012 пикселей), которая позволит значительно развить систематическое изучение звезд, галактик и крупномасштабных космических структур (эксперимент SDSS). Все эти эксперименты рассчитаны на длительный период и предполагают накопление и последующую обработку огромных массивов данных.

Исследование взаимодействий частиц с целью поиска новых физических явлений и частиц будет проводиться на ускорителе LHC в течение двух десятилетий. Примерно столько же лет на LIGO будут регистрироваться и анализироваться космические гравитационные волны наиболее энергетических природных объектов. Проект SDSS предполагает крупномасштабную космическую съемку для создания наиподробнейшего каталога астрономических данных. Все перечисленные эксперименты характеризуются широчайшей

географической разобщенностью нескольких тысяч участников этих проектов. Задачи, поставленные в этих экспериментах, являются беспрецедентными в истории науки и общества с точки зрения их практической реализации, поскольку потребуется, как и в задачах, стоящих перед экспериментаторами на LHC, выполнить следующее:

• отделять очень малые полезные сигналы от колоссальных фоновых сигналов,

• обеспечивать быстрый и прозрачный доступ к экспериментальным данным, находящимся в огромных хранилищах (от 100 Тбайт на начальном этапе до 100 Пбайт в последующие десять лет),

• обеспечивать прозрачный доступ также и к распределенным процессорным ресурсам (при увеличении потребностей в вычислительных ресурсах в 1000 раз к 2010 г.)

• организовывать весь процесс доступа к данным и их анализа из различных точек земного шара по сетевым каналам с высокой пропускной способностью.

Требования по вычислительным и архивным ресурсам для этих трех экспериментов различны. Наибольшие процессорные затраты необходимы для LIGO (уровня петафлопов). Объемы данных на LHC будут значительно больше, чем в LIGO, а в LIGO — значительно больше, чем в SDSS.

Эксперимент SDSS уже находится в рабочей стадии, а в LIGO съем данных начнется с 2002 г. Ускоритель LHC и физические установки ускорителя будут запущены после 2005 г. Поэтому по проекту GryPhiN намечено постепенное создание 19–20 так называемых региональных центров второго уровня (согласно классификации, предложенной проектом MONARC) для этих трех экспериментов: 2 или 3 для SDSS, 5 центров для LIGO и 12 центров для LHC (по 6 для установок CMS и ATLAS соответственно).

Авторы GriPhyN хорошо отдают себе отчет в том, что реализация их проекта возможна потому, что уже разработано и апробировано значительное число приложений, ориентированных на использование в распределенных системах. В рамках проекта PPDG [15] успешно тестировались приложения для использования распределенных баз данных в физике высоких энергий. В проекте China Clipper [16] создаются модели анализа данных в распределенных системах. Проектом Globus разработан целый комплекс математического обеспечения промежуточного уровня (middleware). В проекте MONARC, как уже было упомянуто выше, выполнено моделирование компонентов grid и их взаимодействие. В проектах Nile [17] и Condor [18] созданы системы удаленной обработки заданий. В проекте GIOD [19, 20] исследовались массовые перемещения объектных данных между удаленными сайтами.

Одновременно с GriPhyN в США будут реализовываться сходные grid-структуры и для других отраслей научных исследований. Планируются работы по созданию глобальной информационной системы, содержащей дан-

ные наблюдений за Землей (3 Пбайт данных уже к 2001 г.). Создания grid требуют и проект по исследованию головного мозга человека, и изучение генома человека, и проекты по объединению уже накопленных и дальнейшему сбору астрофизических и географических данных, и метеорологический анализ спутниковых данных. Это далеко не полный перечень запланированного (и, в основном, подкрепленного финансированием) намерения реализации концепции grid в США. Во многих странах Европы сейчас также идет создание национальных grid-сегментов. Принимая во внимание тот факт, что в России ведутся научные исследования практически во всех перечисленных выше областях, а также не забывая об общей мировой тенденции создания глобального информационного общества XXI века, необходимо отметить, что в России также назрела необходимость создания локального grid-сегмента [21, 22].

1.4. Проект Globus. Как уже было упомянуто выше, при разработке grid-структур важнейшим моментом является создание математического обеспечения промежуточного слоя (middleware), которое будет обеспечивать безопасный доступ к данным большого объема в универсальном пространстве имен, перемещать и тиражировать (реплицировать) данные с высокой скоростью с одного географически удаленного узла (сайта) на другой и организовывать синхронизацию удаленных копий.

Наиболее развитым на настоящий момент математическим обеспечением промежуточного слоя является набор инструментальных средств Globus, разрабатываемый в рамках одноименного проекта [13]. Следует отметить, что Globus не является замкнутым комплексом утилит, а представляет собой инфрастуктуру сервисов (services) и набор инструментов для разработки распределенных приложений [3].

К основным видам сервисов, включенных в Globus, относятся: связь, информационное обслуживание, безопасность, управление ресурсами, запуск и управление заданием, доступ к удаленным данным.

В рамках данного обзора не представляется целесообразным детализировать структуру этих сервисов с точки зрения их программной реализации, использования различных сетевых протоколов и стандартов. Однако мы считаем необходимым в нескольких словах охарактеризовать степень сложности проблем, возникающих при создании подобных наборов инструментальных средств.

Применямые в Интернете протоколы по многим причинам не вполне удовлетворительны, поскольку велики накладные расходы, потоковая модель TCP для ряда режимов непригодна, а интерфейсы не позволяют контролировать все параметры. Поэтому неизбежно возникает идея альтернативных интерфейсов связи. Базовый коммуникационный слой Globus, называемый Nexus, вводит понятие **коммуникационной связи** как совокупости начальной и конечной точек соединения. Под методом коммуникации понимается

не только используемый протокол связи, но и аспекты безопасности, надежности, качества обслуживания и компрессии. Приложение может управлять методом коммуникации на каждой отдельной связи путем задания атрибутов для начальной и конечной точек. Реализация связи с помощью Nexus требует наличия базы данных с динамически собираемой информацией о сети, включая топологию сети, протоколы, пропускную способность и задержки.

Управление информацией в Globus осуществляется посредством сервиса MDS (Metacomputing Directory Service). При этом формируется иерархическое древовидное пространство имен DIT (Directory Information Tree) распределенной структуры (в том смысле, что отдельные поддеревья могут располагаться на различных серверах). Информационное дерево каталогов DIT содержит в себе информацию о всех доступных ресурсах (компьютерах, сетях, протоколах, алгоритмах) и их состоянии.

Обеспечение **безопасности** в распределенной среде, помимо решения обычных проблем аутентификации, авторизации и разграничения прав, связано также с поддержкой локальной гетерогенности и глобального контекста безопасности. Под локальной гетерогенностью понимается участие объектов, определенных в разных административных доменах с собственной политикой. Поддержка глобального контекста безопасности требует установления доверительных отношений потенциально между любыми двумя процессами в распределенной среде, поскольку сетевые приложения, в отличие от традиционных приложений клиент–сервер, могут порождать процессы на множестве компьютеров, и при этом возможно последующее взаимодействие этих процессов.

Под **управлением ресурсами** в распределенной среде понимается обнаружение и выделение ресурсов с установлением аутентификации, авторизации и подготовкой ресурсов к их использованию в сетевом приложении. При этом должны учитываться аспекты автономии сайтов (как структурных единиц со своей административной политикой), гетерогенности сайтов (как структурных единиц с различными локальными системами управления ресурсами) и единой политики управления (в смысле определения правил взаимодействия локальных сайтов с глобальной средой). Для организации **заказа ресурсов** в Globus разработан специальный язык спецификации RSL (Resource Specification Language). В качестве интерфейса между глобальной распределенной средой и ее автономными частями — структурными единицами (отдельными компьютерами или кластерами компьютеров) в Globus используется локальный менеджер GRAM, который обрабатывает RSL-запросы, **запускает** задания и контролирует их выполнение, а также периодически обновляет статусную информацию в базе данных MDS.

Задачи **управления данными** в Globus обеспечиваются с помощью интерфейса GASS. Управление данными в Globus ограничивается удаленными операциями ввода/вывода для файлов, управлением локальных кэшей фай-

лов и переносами файлов в модели клиент–сервер с поддержкой различных протоколов. В настоящее время в рамках проекта Globus ведутся работы по организации управления данными, включая управление репликами и оптимизацию переноса файлов через глобальные сети.

2. ФЕРМЫ И КЛАСТЕРЫ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

Как мы можем наблюдать, в последние годы физика высоких энергий (ФВЭ) и ядерная физика испытывают все возрастающие потребности в вычислительных ресурсах, и все более острой становится необходимость найти оптимальную аппаратную базу для реализации этих потребностей. С появлением в 1995 г. процессоров Intel Pentium Pro, обеспечивавших высокую производительность при невысокой стоимости по сравнению с RISC-процессорами и Unix-рабочими станциями, стало очевидно, что с точки зрения соотношения цена/производительность использование персональных компьютеров становится наиболее эффективным [23]. В начале 1997 г. более десяти групп из NASA, DESY, Fermilab, Sandia, NIH [24] сообщили о своих оценках производительности для различных приложений на персональных компьютерах, а также объявили о своих планах создания кластеров*, состоящих из тысяч процессоров. К 1999 г. Linux PC-фермы** стали доминировать в компьютинге физики высоких энергий и ядерной физики [26, 27], что оказало решающее влияние на широкое использование операционной системы Linux во многих других сферах науки.

В физике высоких энергий, по самой природе независимых взаимодействий, задачи реконструкции событий из экспериментальных или модельных данных являются типичными задачами вычислений со слабосвязанным параллелизмом. Такого рода параллелизм хорошо реализуется на кластерных структурах со слабосвязанными процессорами при использовании определенных программных средств, таких, например, как PVM [28], MPI [29] или разработанный в Fermilab CPS (Cooperative Process Software).

Linux позволяет достаточно просто совершать переход от работы на Unix-станциях к работе на персональных компьютерах [30]. На настоящий момент Linux является открытой, хорошо поддерживаемой и широко используемой

*Вычислительный кластер — это совокупность компьютеров, объединенных в рамках некоторой сети для решения определенной задачи [25].

**Ферму можно определить как некоторую узкоспециализированную разновидность кластерной структуры, в которой осуществляется не только выполнение вычислительной задачи, но и связь с системами массовой памяти или организация ввода/вывода в больших объемах, как, например, на фермах с записью данных в режиме реального времени. Можно также добавить, что термин «ферма» используется преимущественно в ФВЭ.

операционной системой, включающей средства мультизадачности. Основная часть системной инфраструктуры Linux вне ядра — библиотеки, компиляторы, утилиты также являются свободно распространяемыми продуктами от GNU (Free Software Foundation). Свободно доступен ряд компиляторов с языков C, C++ и Фортран: gcc, g77, Microway, Absoft, g++, egcs и KAI, а также f2c-конвертор. Linux пригоден для многих платформ, в том числе Intel 386, 486, Pentium, P-Pro, P-II, P-III, DEC Alpha, MIPS R4x00 и SPARC Fujitsu(AP+). Большинство широко используемых физиками пакетов общего назначения (редакторы, текстовые процессоры, сетевые средства связи, математические библиотеки и т. п.), а также физические приложения легко адаптируются в Linux-среде. Для Linux-платформы также поддерживается большинство развитых систем пакетной обработки заданий для многомашинной среды (их называют еще системами диспетчеризации заданий или системами балансировки загрузки) — коммерческие системы LSF [31] и CODINE [32], свободно распространяемые NQS [33], NQE, PBS [34], Condor [18], DQS и некоторые другие. Linux официально поддерживается не только компаниями — мировыми лидерами в разработке баз данных: Oracle, Informix, Computer Associates, Sybase, IBM, Objectivity, но и компаниями Netscape, Corel, Interbase, Adaptec, Cygnus, Sun, SGI, DELL, Compaq. Это внушает оптимизм в плане дальнейшего успешного развития операционной системы Linux*.

Как уже было упомянуто выше, главной причиной все более широкого использования персональных компьютеров является высокая производительность при низкой цене. Персональные компьютеры дешевле Unix-рабочих станций не менее чем в 10 раз. За период с 1997 по 1998 г. соотношение цена/производительность для персональных компьютеров снизилось ровно вдвое, и сохраняется тенденция уменьшения этого показателя не менее чем 1 % за неделю [23]. Быстрой является также тенденция увеличения скорости процессора: происходит удвоение скорости каждые 12–18 месяцев. Процессоры AMD и Intel уже вышли на гигагерцевый рубеж частоты. Фирмами Hewlett Packard и Intel объявлено о совместном создании нового мощного процессора Itanium [36].

Сочетание большого количества персональных компьютеров приводит к достижению производительности суперкомпьютера. К настоящему моменту во всем мире для применения в исследованиях по физике высоких энергий и ядерной физике создано несколько сотен ферм и кластеров персональных ЭВМ, состоящих каждый из десятков или сотен персональных компьютеров.

*Мы считаем необходимым подчеркнуть, что операционную систему Linux начали активно использовать в компьютинге для ФВЭ во многом благодаря пионерским работам по адаптации специализированного математического обеспечения CERN в Linux-среде, выполненным безвременно ушедшим из жизни сотрудником ОИЯИ Виктором Балашовым [35].

(Так, например, в Fermilab в 1999 г. общая производительность Linux PC-ферм (в основном на базе 2xCPU Pentium III 500 МГц — всего более 300 процессорных единиц) составляла более 6000 SPECint95 [37].) Эти фермы и кластеры успешно используются для решения различных задач: это и моделирование по методу Монте-Карло, и оф-лайн-обработка данных, и триггерный отбор событий в режиме реального времени. Следует добавить, что произошел также переход от использования Unix-рабочих станций как настольных компьютеров (desktop computing) к повсеместному использованию PC.

Неудивительно, что во всех планах и прогнозах по созданию региональных вычислительно-информационных центров для LHC основной упор делается на использование ферм и кластеров персональных компьютеров как главного источника требующихся колоссальных процессорных мощностей.

2.1. Проект Grid Data Farm. В рамках участия в эксперименте ATLAS в Японии создан проект Grid Data Farm for Petascale Data Intensive Computing [38]. Проект инициирован KEK и поддержан рядом других японских научно-исследовательских центров. В процессе реализации этого проекта будет создан масштабируемый кластер из нескольких тысяч процессорных единиц, каждая из которых будет иметь примерно 1 Тбайт дисковой памяти; данные из CERN будут поступать по каналу пропускной способности 600 Мбит/с. В проекте Grid Data Farm будет реализовано:

- создание распределенной файловой системы объемом несколько Пбайт;
- параллельный ввод/вывод и параллельная обработка для быстрого анализа данных;
- глобальная аутентификация и контроль за доступом;
- глобальное управление ресурсами и планирование для всех тысяч узлов, входящих в кластер;
- разделение данных между отдельными региональными центрами и эффективный доступ;
- совместное использование программ;
- мониторинг системы и администрирование;
- аварийная устойчивость.

К 2005 г. планируется завершить создание данной фермы, которая далее будет использоваться в действующем эксперименте ATLAS в режиме реального времени. Предполагается распространить опыт создания фермы на другие отрасли науки: биоинформатику, астрономию, систему наблюдений за Землей.

2.2. Создание ферм персональных компьютеров в ОИЯИ. В соответствии с тенденцией все более активного использования PC-Linux ферм и кластеров для целей физических экспериментов за последние два года в

ОИЯИ было создано несколько небольших ферм* персональных компьютеров [39]. Еще в 1998 г. был разработан и реализован проект по созданию интегрированного вычислительного комплекса на базе персональных ЭВМ для массовой обработки однородной информации. В результате была создана первая в ОИЯИ ферма персональных компьютеров (ферма ЛФЧ–ЛВЭ), которая с 1999 г. успешно используется для расчетов нескольких физических экспериментов (STAR, EXCHARM, NA48), и планируется ее использование для экспериментов, ориентированных на LHC. К лету 2000 г. в ОИЯИ были созданы еще две фермы: в ЛЯП и ЛИТ, которые будут использоваться для ряда физических экспериментов, в том числе для ALICE, ATLAS и CMS. Общая производительность трех PC-ферм ОИЯИ — более 1000 SpecInt95. Конечно, ресурсы ферм ОИЯИ весьма скромные, но тем не менее ОИЯИ уже на данный момент располагает определенной базой для проведения различных расчетов и исследований по тематике ALICE, ATLAS и CMS.

3. ПЕРЕХОД НА ОБЪЕКТНО-ОРИЕНТИРОВАННУЮ ПЛАТФОРМУ В ПОСТРОЕНИИ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ И БАЗ ДАННЫХ ДЛЯ ЭКСПЕРИМЕНТОВ ПО ФИЗИКЕ ВЫСОКИХ ЭНЕРГИЙ

Вторая половина 90-х годов знаменательна тем, что произошел переход на объектно-ориентированный подход (ООП) в создании математического обеспечения для приложений ФВЭ. Объектно-ориентированное программирование — это сравнительно новый подход к программированию, нашедший свое наиболее гармоничное выражение в языке програмирования C++, созданном Б. Страуструпом еще в 1979 г. и претерпевшем с того времени существенные модернизации [40].

Как хорошо известно, по мере развития вычислительной техники возникали различные методики программирования: вначале это было программирование в машинных кодах, затем появился язык ассемблер, далее — первый язык высокого уровня Фортран. Однако по мере усложнения программ и увеличения размера их кодов программы становились нечитабельными и плохо управляемыми. Тогда возникла идея создания структурированных языков, таких как Алгол, Паскаль и C. Сутью структурного программирования является возможность разбиения программы на составляющие ее элементы. Дальнейшее усложнение задач программирования привело к необходимости создания нового подхода, в результате чего и был выработан объектно-

*Если следовать приведенному нами выше определению понятий «кластер» и «ферма», то «фермы» в ОИЯИ на данном этапе следует называть «вычислительными кластерами»; однако стихийно сложилась практика наименования их «фермами».

ориентированный подход, аккумулирующий идеи структурного программирования и сочетающий их с новыми мощными концепциями. Объектно-ориентированный подход дает возможность разложить сложную проблему на составные части, причем каждая составляющая становится самостоятельным объектом, содержащим свои коды и данные, которые к этому объекту относятся. Разбиение сложной задачи на группы более простых увеличивает надежность программного обеспечения, создает основу для гибкого и четкого процесса управления разработкой кода, открывает новые возможности для повторного использования кода. Принципиальным моментом с точки зрения создания программы становится необходимость разработки прежде всего структуры данных и только затем — способов работы с этими данными. Этот аспект крайне важен для приложений ФВЭ, поскольку данные сохраняют свою актуальность достаточно долго, в то время как для интерпретации этих данных могут применяться различные методы.

Сама концепция **объектов** очень естественна для организации логического процесса анализа физических данных. Кроме того, как справедливо отмечает создатель C++ Б. Страуструп, наиболее сильной стороной этого языка является возможность активного его использования в программах, предназначенных для широкого диапазона прикладных областей. В этом смысле можно считать типичным приложение, которое включает в себя доступ к глобальной и локальной сетям, численные расчеты, графику, интерактивное взаимодействие с пользователем и обращение к базам данных. Ранее такие области считались раздельными и реализация математического обеспечения производилась группами специалистов различного профиля и с использованием разных языков программирования. Создание подобных приложений как раз особенно актуально для современных крупных экспериментов по ФВЭ.

Для того чтобы прояснить мотивацию перехода на ООП в создании математического обеспечения для ФВЭ, следует кратко остановиться на основополагающих концепциях ООП в данном контексте.

Инкапсуляция предполагает рассмотрение частей системы как *объектов*, включающих в себя переменные, описывающие *состояние* объектов, и программы, описывающие *поведение* этих объектов. Внутренние механизмы этого скрыты от пользователя. Например, при реконструкции трека электрона в электромагнитном калориметре (ECAL) необходимо создать объекты, описывающие собственно ECAL, трек, кластер и т. д. Для определения состояния трека необходимы переменные, описывающие компоненты импульса трека. Код программы для ECAL и данные хранятся совместно, разработка и развитие их ведется независимо от программ, описывающих работу всех иных частей физической установки. Любые изменения в данных, описывающих ECAL, или в алгоритмах реконструкции производятся безболезненно и незаметно для пользователей в том смысле, что это никак не влияет на пользовательский интерфейс к программе. Можно сказать, что ООП обеспечивает

высокий уровень абстракции в смысле выделения главных признаков объекта, предоставляя пользователю возможность работы с этим объектом, но оставляя «за кадром» внутреннюю имплементацию этого объекта.

Если какие-то различные детекторы физической установки (например, электромагнитный и адронный (HCAL) калориметры) имеют общие свойства, то механизм **наследования** позволяет использовать одни и те же объекты для описания этих свойств, т. е. мы можем определить класс Calorimeter и классы ECAL и HCAL, которые будут наследовать свойства класса Calorimeter.

Возможность понизить сложность программы с помощью концепции **полиморфизма** («один интерфейс, но множество методов») реализуется путем создания единого класса действий, выполнение которых зависит от типа данных, т. е. имеется одно имя для задания общих для класса действий, но какое именно действие будет выполнено, определяется типом данных. (Один из самых популярных и наглядных примеров на данную тему: функция «draw», с помощью которой можно изобразить различные геометрические фигуры, в зависимости от того, что будет задано как объект рисования — окружность, квадрат или треугольник.)

3.1. Модель компьютинга для эксперимента BaBar как пионерский опыт перехода на объектно-ориентированную платформу. Первые попытки использовать ООП для создания приложений для ФВЭ предпринимались уже с середины 80-х годов (например, в проектах Pions [41] (CERN) и REASON [42] (SLAC), в объектно-ориентированной программе GISMO [43] для моделирования и реконструкции событий в ФВЭ, в наборе инструментальных средств GEANT4 [44] для моделирования в ФВЭ); однако впервые полный переход к созданию математического обеспечения для крупного эксперимента в области ФВЭ исключительно на основе ООП был осуществлен коллаборацией BaBar [45] начиная с 1995 г.

Эксперимент BaBar был организован в SLAC (США) на ускорителе PEP-II для исследования нарушения CP-инвариантности в B-мезонной системе [46]. Распады с нарушением CP-инвариантности встречаются очень редко. Поэтому потребовалось собрать огромные объемы данных, что, соответственно, привело к необходимости непрерывной работы детектора BaBar в режиме «фабрики» на ускорителе PEP-II в течение почти полугода каждый год. Ежегодный объем данных — это 10^9 событий, что составляет примерно 25 Тбайт данных. Кроме того, коллаборация BaBar является очень разобщенной географически, что накладывало свои требования к организации компьютинга с точки зрения как интенсивного использования глобальных сетей, так и разработки математического обеспечения для осуществления полной интеграции всех участников эксперимента, удаленных географически от SLAC.

Модель компьютинга, реализованная для эксперимента BaBar, включает в себя центральный информационно-вычислительный комплекс в SLAC и ряд «региональных» центров, расположенных в США и Европе в институтах-

участниках этого эксперимента. В SLAC организован мониторинг и поддержка детектора, съем данных в режиме реального времени и последующий анализ физических данных. В удаленных от SLAC центрах имеются определенные процессорные ресурсы и ресурсы для хранения данных. Между региональными центрами и центром в SLAC обеспечен необходимый сетевой доступ. Основной объем обработки данных производится в SLAC, а региональные центры ведут работы по моделированию данных, разработке кодов программ и физическому анализу. Только небольшая часть базы данных в SLAC тиражируется в региональные центры. Как видно, данная модель во многом сходна с той, которую планируется реализовать для LHC, но в гораздо менее широких масштабах, и в ней не ставилось сложных задач с точки зрения организации управления распределенными данными.

Наибольший интерес организация компьютинга для BaBar представляет в том отношении, что это был первый крупный эксперимент в области ФВЭ, в котором был решительно произведен переход на объектно-ориентированную платформу. К моменту принятия такого решения объектно-ориентированное программирование почти не использовалось в приложениях для ФВЭ, поэтому пришлось преодолеть большие трудности при создании математического обеспечения: в плане как переориентации разработчиков, так и преодоления инерции пользователей, в течение многих лет работавших с языком Фортран.

В результате за довольно короткий срок — в силу вынужденной необходимости и полной безальтернативности — было создано хорошо структурированное математическое обеспечение для эксперимента BaBar, написанное на языке C++ и состоящее из отдельных «пакетов» со строго определенными наборами классов и функций; управление исходными кодами программ осуществляется с помощью средства CVS. Этот набор пакетов поддерживается для всех операционных систем, используемых в институтах-участниках BaBar; все текущие версии доступны для коллаборантов как исходные тексты и как откомпилированные библиотеки и исполняемые модули для соответствующих платформ. Разработка программного обеспечения была организована с очень продуманным уровнем модулярности так, что, например, в программе для распознавания трека и нахождения вершины взаимодействия изменения в любой части программы — методов распознавания образов, алгоритмов фитирования, параметризации калибровки, описания детектора — могут быть произведены без коренного изменения всей программы в целом. Именно использование ООП позволило достичь такого результата.

Переход на объектно-ориентированное математическое обеспечение для реконструкции и анализа потребовал найти решение для управления вводом/выводом получаемых в эксперименте данных как объектов. В подобной ситуации объектно-ориентированные СУБД (ОО СУБД) (ODBMS — Object Oriented Data Base Management System) являются наиболее естественной мо-

делью организации хранения данных, не только обеспечивая хороший интерфейс с C++, но и предоставляя средства для управления большими объемами данных в распределенной гетерогенной среде. BaBar остановил свой выбор на коммерческой ODBMS Objectivity/DB.

Для каждого зарегистрированного в эксперименте физического события предусмотрены следующие типы объектов в базе данных:

• SIM (Simulated Truth) — данные, полученные с помощью моделирования по методу Монте-Карло.

• RAW — так называемые «сырые» данные, полученные в эксперименте или с помощью моделирования. Эти данные не подлежат изменениям. Размер данных — 25 Кбайт на одно событие.

• REC (Reconstructed Data) — данные, полученные в результате использования программ реконструкции событий. Эти данные могут быть получены повторно. Размер данных — 100 Кбайт на одно событие.

• ESD (Event Summary Data) — скомпрессированные реконструированные данные (примерно аналогичные традиционным DST-данным). Размер данных — 10 Кбайт на одно событие.

• AOD (Analysis Object Data) — данные для конечного физического анализа. Размер данных — 1 Кбайт на одно событие.

• TAG — скомпрессированные данные-признаки для классификации событий, позволяющие осуществлять единичные выборки данных. Размер данных — менее 100 байт на одно событие.

• HDR (Event Header) — указатели на события, содержащие ссылки на информацию о событиях; их размер — менее 64 байт на одно событие.

Объекты в базе данных доступны через некоторые «наборы» («collections»), содержащие ссылки на события, хранимые в базе данных. Новые наборы могут создаваться по мере появления новых данных и могут быть доступны по своему имени, находящемуся в так называемом «словаре». Для увеличения эффективности доступа находящиеся в базе данные могут быть реплицированы в региональные центры коллаборации. Objectivity/DB предоставляет поддержку для организации распределенной базы данных. Достаточно большой суммарный объем базы данных (примерно 100 Тбайт) не дает возможности разместить всю базу на дисках, поэтому основная часть данных находится на лентах роботосистемы SLAC емкостью 600 Тбайт и по мере необходимости перемещается на диски. Наиболее часто используемые данные хранятся на дисках быстрого доступа.

Как показал успешный опыт адаптации Objectivity/DB в эксперименте BaBar, этот вариант объектно-ориентированной базы данных вполне удовлетворяет высоким требованиям к базам данных в ФВЭ с точки зрения функциональности, производительности, масштабируемости и надежности [47].

Для того чтобы в полной мере использовать возможности C++-объектов, получаемых в результате реконструкции событий, был разработан набор инструментальных средств Beta для физического анализа данных. Beta обеспечивает достаточный уровень абстракции для физиков, чтобы дать возможность эффективно производить нахождение вершин, идентификацию частиц и т. п. Beta разработан с учетом новизны применения языка C++ для физиков, что позоляет использовать свои возможности не только опытным программистам на C++, но и предоставляет простой интерфейс для начинающих пользователей. Для этого были сформулированы четыре базовых понятия.

• «Кандидаты» («candidate») — те события, в которых треки заряженных частиц предположительно могут содержать пионы, а нейтральные кластеры — фотоны. В результате нахождения вершины для двух кандидатов образуется новый кандидат, указывающий на «частицу», вызвавшую образование этих двух треков. Для всех имеющихся кандидатов любого рода в Beta обеспечивается независимый и однородный доступ для пользователей.

• «Операторы» («operator»), с помощью которых можно производить определенные действия над кандидатами. Так, например, нахождение вершины — это оператор.

• «Искатели» («finders»), с помощью которых происходит поиск частиц распада (например, нейтральных K- и D-мезонов).

• «Ассоциаторы» («associators») — для установления соответствия между кандидатами (например, между SIM-данными и реконструированными треками или треками заряженных частиц и кластерами).

Пионерский опыт эксперимента BaBar по переходу к ООП оказался очень полезным для всех новых крупных экспериментов по ФВЭ с точки зрения как создания математического обеспечения нового поколения*, так и использования ОО СУБД в ФВЭ.

3.2. Создание объектно-ориентированного программного окружения для экспериментов на LHC. *3.2.1. Проект LHC++.* В течение многих лет в CERN создавался, развивался и активно использовался в физических исследованиях набор прикладных программ, хорошо известный физикам как CERNLIB [49] и написанный на языке Фортран-77. Предполагалось перевести эти программы на Фортран-90 и продолжать дальнейшую поддержку и развитие этого математического обеспечения. Однако тенденция к переходу на ООП привела к организации в 1995 г. проекта LHC++ [50] по созданию «C++-эквивалента CERNLIB».

*Можно отметить, что самым популярным и эффективным для физиков до сих пор считается курс лекций Пауля Кунца (P. Kunz) из SLAC «C++ для физики частиц» [48]. Автора на протяжении ряда последних лет постоянно приглашают во многие научные центры мира, в том числе и в CERN, для прочтения этой серии лекций по C++, пользующейся неизменным успехом.

Таблица 3. Структура LHC++

Назначение	Название компонента
Моделирование детекторов	GEANT4
Анализ данных	IRISExplorer – HEPExplorer
Пользовательская графика	MasterSuite – HEPInventor – HepVis
Базовая графика	OpenInventor – OpenGL
Математические библиотеки для ФВЭ	HEPFitting – GEMINI – CLHEP
Базовые математические библиотеки	NAG C library (с C++-заголовками (headers))
Гистограммирование	HTL
Долговременное хранение данных (Persistency)	Objectivity/DB
Поддержка C++	Стандартные библиотеки

К настоящему моменту LHC++ состоит из ряда как коммерческих стандартных компонентов, так и специфических приложений для ФВЭ. Основные компоненты LHC++ представлены в табл. 3. Все базовые функции реализованы как библиотеки классов C++, а интеграция осуществлена посредством более сложной системы визуализации.

Коммерческие компоненты: OpenGL [51, 52] — промышленный графический стандарт, OpenInventor [53] — набор инструментальных средств для интерактивного графического программирования, MasterSuite [54] — набор инструментальных средств для визуализации данных (фактически — надстройка к OpenInventor), IRISExplorer [55] — набор инструментальных средств для визуализации научных данных, Objectivity/DB [56, 57] — ОО СУБД, рассматриваемая как возможный вариант организации баз данных петабайтных объемов с доступом к системам массовой памяти, стандартная математическая библиотека NAG C [58], GEMINI [59] — пакет минимизации и анализа ошибок.

Специфические приложения для ФВЭ: CLHEP [60] — библиотека классов базового уровня, необходимых для разработки приложений для ФВЭ; HTL [61] — библиотека классов для построения гистограмм; HEPInventor [62] — графическая библиотека классов (надстройка к MasterSuite) для обеспечения интерфейса между HTL-структурами данных и графикой; HepVis [63] (расширение OpenInventor) — набор объектов, обеспечивающий гра-

фическое представление процессов в физических экспериментах на коллайдерах; HEPExplorer [64] (HEP-specific IRISExplorer) — инструментальное средство для анализа экспериментальных данных; HEPFitting [65] — приложение для обработки данных, основанное на пакете GEMINI и позволяющее осуществлять фитирование загружаемых данных.

Таким образом, программное окружение LHC++ предоставляет базовый набор библиотек классов для создания приложений в ФВЭ, набор стандартных математических библиотек, набор графических библиотек и средств визуализации, инструментарий для анализа данных, средства гистограммирования, генерации событий и моделирования детекторов и организацию хранения данных.

3.2.2. Проект ROOT. Имевшийся богатый и плодотворный опыт создания таких хорошо известных физикам средств интерактивного анализа данных, как PAW и PIAF, и пакета моделирования GEANT позволил небольшой группе разработчиков во главе с Р. Браном (R. Brun) решиться на создание объектно-ориентированного аналога перечисленных выше средств, не привлекая при этом коммерческих программных продуктов. Так возник проект ROOT [66], первоначальный вариант которого был создан для эксперимента NA49 [67]. К настоящему моменту ROOT является мощным инструментальным средством, в котором успешно интегрированы все необходимые для обработки и анализа физической экспериментальной информации функции. ROOT принят как базовая программная среда одним из планируемых на LHC экспериментов — ALICE [68, 69], рядом физических экспериментов в США (STAR [70, 71], CDF [72, 73]), экспериментом HERA-B [74] в Германии, немецким Исследовательским центром по физике тяжелых ионов GSI [75] (Дармштадт). Для эксперимента ATLAS была создана программа быстрого моделирования ATLFast++ [76] на базе ROOT, в эксперименте D0 (Fermilab) ROOT используется в системе мониторинга событий в режиме реального времени [77]. Разработаны интерфейсы ROOT-Oracle [78] (только для среды Linux) и ROOT-Objectivity/DB [79].

4. ОРГАНИЗАЦИЯ ХРАНЕНИЯ ДАННЫХ И ДОСТУПА К ДАННЫМ

Проблема организации хранения и доступа к данным для экспериментов, планируемых на LHC, является очень сложной и многоплановой, поскольку надо не только организовать долговременное надежное хранение огромных объемов накапливаемых экспериментальных данных, но и обеспечить к ним достаточно быстрый и прозрачный для пользователя доступ из географически очень удаленных научных центров.

4.1. Системы управления массовой памятью. Наиболее развитой в настоящее время системой управления массовой памятью является система HPSS (High Performance Storage System), разработанная фирмой IBM Goverment Systems [80]. HPSS успешно эксплуатируется в течение нескольких лет в ЦЕРН. Система HPSS состоит из целого комплекса программ, обеспечивающих управление огромным иерархическим массивом, и сервисов для работы с этим хранилищем данных. HPSS является распределенной системой в том смысле, что управляемые объекты располагаются физически в различных местах, и доступ к ним осуществляется по сети. HPSS снабжена также средствами масштабируемости: т. е. новые объекты управления можно добавлять по мере необходимости. К управляемым объектам относятся как устройства физического хранения данных, так и серверы, поставляющие эти данные. В HPSS реализована также возможность передачи данных между устройствами хранения без промежуточной буферизации в памяти управляющего компьютера. HPSS — одна из самых развитых и, следовательно, дорогих систем. Существует значительное число иных подобных систем (например, Unitree [81], OMNISTORAGE [82], EMC [83]), но их функциональное наполнение не столь развито, как в HPSS. Существуют также определенные попытки создания некоммерческих продуктов систем технологии массовой памяти (Castor [84, 85], Enstore [86]). В настоящее время находится в рабочей стадии проект EuroStore [87] по созданию интегрированной системы, включающей в себя распределенно-параллельную файловую систему PFS (Parallel File System) и систему управления массовой памятью HSM (Hierarchical Storage Management). Проект EuroStore является специализированным в том смысле, что ориентирован на требования к организации хранения данных и доступа к ним, предъявляемые в ФВЭ, в особенности для организации компьютинга экспериментов, планируемых на LHC.

4.2. Выбор СУБД. Переход на ООП в создании математического обеспечения для ФВЭ привел к необходимости выбора соответствующей ОО СУБД, которая должна отвечать требованиям масштабируемости (до размера в несколько сотен Пбайт), обладать возможностями работы в распределенной гетерогенной среде, иметь интерфейс к системам массовой памяти и быть *устойчивой (persistent)*. Таким требованиям соответствует, например, ОО СУБД Objectivity/DB, которая успешно используется в эксперименте BaBar [88]. В ходе проекта RD45 [89] отделения информационных технологий в CERN также установлено на практике успешное использование Objectivity/DB при работе с базами данных объемами порядка терабайт [90]. Objectivity/DB на данный момент применяется для организации баз данных для множества экспериментов, в том числе для CMS [91], ATLAS [92], LHCb [93].

Реляционные базы данных, которые традиционно использовались в компьютинге для экспериментов по ФВЭ, такие как ORACLE и MySQL, в настоящее время претерпевают модификации в направлении ОО-технологии, и

не исключается возможность их будущего использования, в том числе и в компьютинге для LHC.

4.3. Средства управления данными в проекте EU Data Grid. В европейском проекте EU Data Grid, о котором уже было кратко упомянуто в данном обзоре, делается попытка реализовать некоторую модель управления данными для возможного дальнейшего использования ее в структуре региональных центров для LHC.

В проекте EU Data Grid в рамках пакета по созданию средств управления данными подразумевается несколько слоев сервисов. К сервисам высокого уровня относятся управление реплицированием (Replica Management), оптимизация запросов и управление шаблоном доступа (Query Optimization & Access Pattern Management), к сервисам промежуточного слоя — организация пересылки, поиска и доступа к данным. Центральные сервисы — это собственно системы управления хранением данных (HPSS, Castor или просто локальные файловые системы) и системы управления метаданными*. Согласно управлению репликой копии файлов или метаданных будут помещаться в распределенный иерархический кэш. Для выполнения этой задачи необходимо обращение к блоку пересылки данных в промежуточном слое, который, в свою очередь, будет использовать средства доступа к данным или указатели к метаданным, хранящимся под управлением тех или иных систем управления хранением данных или метаданных. Все перечисленные компоненты должны обеспечивать надлежащие механизмы безопасности.

Стержневой проблемой управления данными в Data Grid-структуре является гетерогенность репозиториев данных. Задача управления данными должна решаться для различных систем управления путем хранения данных центрального сервиса Data Grid: это могут быть системы управления массовой памятью HPSS, Castor, Unitree или Enstore, распределенно-параллельные дисковые системы, например, DPSS [94], распределенные файловые системы AFS или NFS, а также базы данных. При такой гетерогенности организации хранения данных очень сложным является решение проблемы наименования и доступа к данным в столь различных системах хранения данных. При иерархической организации управления памятью обеспечивается автоматический и прозрачный доступ к хранилищу данных, состоящему из лент, промежуточного дискового хранилища данных и дисков быстрого доступа. В такой иерархической системе данные переносятся сначала с лент на локальный дисковый кэш до начала grid-переноса данных. При этом запросы должны группироваться таким образом, чтобы достичь оптимального монтирования лент, что требует организации внутренних каталогов и механизмов переноса данных с ленты на диск.

*Метаданные содержат в себе информацию о структуре хранимых данных.

Репликация (тиражирование) данных может рассматриваться как процесс управления копиями данных, а также это есть стратегия кэширования, при которой идентичные файлы доступны в нескольких местах grid-инфраструктуры. Главной целью репликации является достижение более быстрого доступа к данным за счет их местонахождения в локальном кэше или в ближайшем местонахождении копии данного файла: т. е. не приходится осуществлять перенос файла по всей глобальной сети для каждого единичного запроса. Каждая реплика должна синхронизоваться с другими репликами. Качество реплики зависит от протоколов обновления и сетевых параметров grid-структуры. Должна быть также выработана стратегия обновления и создания реплик. Понятно, что создание реплик особенно актуально при огромных объемах данных (порядка нескольких Пбайт).

Репликация метаданных требует использования механизма связи клиент–сервер на каждом grid-узле. Инструментальный набор средств Globus предоставляет две возможности: сокеты и коммуникационную библиотеку более высокого уровня (Nexus). В подсистеме коммуникации должны быть реализованы различные протоколы репликации (синхронные и асинхронные методы обновления). Replica Manager обеспечивает сервисы доступа высокого уровня и оптимизирует глобальную пропускную способность с использованием grid-кэшей: анализ запроса пользователя приводит к оптимальному решению реализации этого запроса, а в соответствии с анализом множества запросов принимается решение о создании или уничтожении реплики. Replica Manager осуществляет глобальное кэширование, а за создание локальных кэшей ответственны системы массовой памяти.

Связующим элементом в grid-системе является сервис управления метаданными (каталогами с именем и месторасположением единичных или реплицированных файлов, информацией по мониторингу (статус ошибок, пропускная способность и т. п.), информацией по конфигурации grid (описание сетей, коммутаторов, кластеров, узлов и математического обеспечения), стратегиями гибкого динамического управления). Именно этот сервис служит интеграции разнообразных, децентрализованных и гетерогенных составляющих grid.

Многие аспекты обеспечения безопасности в grid-инфраструктуре тесно связаны с управлением данными, в особенности организация grid-кэшей и синхронная стратегия реплицирования.

В распределенной и включающей в себя реплики системе хранения данных запрос оптимизируется за счет существования нескольких копий файлов. Оптимальная схема выполнения запроса зависит от ряда динамических и статических факторов, таких как размер файла, к которому требуется доступ; уровень загрузки на сервере данных для обслуживания запрашиваемого файла; метод/протокол, по которым будет осуществлен доступ к файлу и перенос файла; пропускная способность сети, расстояние и трафик внутри grid; стратегия управления удаленным доступом. Оптимизация запросов может про-

изводиться на разном уровне гранулярности. Например, можно запрашивать не весь файл, а отдельные объекты, содержащиеся в файле, по их идентификаторам (object identifiers — OID) согласно картам размещения информации.

5. ОРГАНИЗАЦИЯ КОМПЬЮТИНГА ДЛЯ ЭКСПЕРИМЕНТА CMS

Рассмотрим подход к организации компьютинга для LHC на примере эксперимента CMS. Коллаборация CMS создает одну из двух основных экспериментальных установок на LHC — компактный мюонный соленоид (CMS — Compact Muon Solenoid) [95]. В коллаборацию CMS входит более 1600 ученых из 150 научных центров мира, в том числе и большая группа сотрудников ОИЯИ.

Определение стратегии компьютинга CMS проводилось в контексте координации работ всех специалистов, работающих в различных научных центрах, и с учетом стремительно меняющейся ситуации в компьютерной сфере. В результате еще в 1996 г. был создан проект по компьютингу CMS [96]. Правильно выбранная стратегия компьютинга позволила успешно решать задачи проектирования установки CMS и создала основу для ее успешной эксплуатации в дальнейшем. Разработанные в CERN предложения по компьютингу CMS касаются как стадии конструирования установки, так и периода ее эксплуатации [97]. Компьютерные требования на этих двух этапах существенно разнятся.

На этапе конструирования установки CMS основной задачей является, прежде всего, моделирование физических процессов и установки, включая генерацию событий, трекинг частиц, моделирование радиационного фона и электронных шумов, моделирование триггеров различного уровня, реконструкцию и анализ событий, а также визуализацию событий и процессов. Перечисленные задачи были решены в общей программе моделирования CMSIM [98], написанной на языке Фортран, и на данном этапе решаются в новом объектно-ориентированном пакете ORCA [99–101], написанном на языке C++ и работающем в программном окружении LHC++. Соответственно во всех научных организациях, участвующих в работах по моделированию, необходимо наличие работающих актуальных версий CMSIM и ORCA и соответствующего унифицированного программного окружения. Для интерактивного анализа данных, визуализации физических событий и детектора в коллаборации CMS создан специализированный пакет IGUANA (Interactive Graphical User Analysis) [102, 103], который базируется на четырех графических пакетах: X11, Qt [104, 105], OpenGL и OpenInventor, а также использует некоторые средства HepVis.

Проведение тестовых сеансов работы прототипов отдельных частей установки CMS предполагает обработку данных, полученных в этих сеансах.

Механические и конструкционные конструкторские разработки, разработка электроники и контроль за качеством готовых частей прототипов и установки ведутся также с привлечением компьютерных средств.

С точки зрения обеспечения информационного сервиса предполагается наличие www-серверов CMS в основных компьютерных центрах коллаборации.

Немаловажной для полноценной работы является создание унифицированной среды. С середины 90-х годов коллаборацией CMS была признана базовой платформа Unix: вначале это была операционная система SunOS, затем коммерческая система Solaris (на платформе Sun-станций). В последние годы прослеживается тенденция активного использования операционной системы Linux Red Hat в связи с ориентацией на массовое создание PC-ферм в CERN и организациях-участницах экспериментов.

В качестве языка программирования предпочтение отдано языку C++ в связи с полным переходом на объектно-ориентированную платформу. Произошел также переход от CMZ-технологии сборки библиотек к технологии CVS. Для сборки больших пакетов используется новое программное средство SCRAM [106].

Прослеживается общность и возможная преемственность в разработке математического обеспечения для экспериментов по ФВЭ, если остановиться, в частности, на иерархии типов данных, принятой в эксперименте CMS и сравнить ее с описанной выше иерархией данных для эксперимента BaBar. В проекте MONARC, в соответствии с требованиями эксперимента CMS, были предложены пять базовых типов данных и смоделирован будущий процесс формирования этих данных на рабочей стадии установки. Во-первых, это так называемые «сырые» данные (Raw Data), которые будут получать непосредственно на работающей установке, а также с помощью моделирования. Размер этих данных — 1 Мбайт на событие. Далее следуют реконструированные данные — ESD (Event Summary Data) — размером 100 Кбайт на событие, физические данные для анализа — AOD (Analysis Object Data) размером 10 Кбайт на событие, ntuple-подобные данные — DPD (Derived Physics Data) — размером 1 Кбайт на событие и так называемые «таггированные» (или «меченые») данные (TAG) размером от 100 до 500 байт на событие, которые будут использоваться для выборки событий. (Если обратиться к привычной для физиков терминологии, то ESD-данные подобны мини-dst, AOD-данные — микро-dst, а TAG-данные — нано-dst.)

Практически во всех организациях-участницах эксперимента CMS имеются на данный момент определенные процессорные мощности и средства хранения данных с обязательной поддержкой унифицированного с точки зрения требований CMS программного окружения, ведется информационная поддержка работ по тематике CMS. В последние несколько лет основной упор с точки зрения развития компьютинга для CMS делается на подготовку к рабо-

чей стадии эксперимента, т. е. на создание прототипов региональных центров различного уровня и на освоение и развитие grid-технологий. Так, например, ведутся работы по массовому моделированию событий для триггера высокого уровня установки CMS. Такое моделирование периодически проходит как непосредственно в CERN, так и в ряде других европейских и американских институтов, претендующих на роль вычислительных региональных центров для LHC. Полученные модельные данные передаются в CERN для включения в объектно-ориентированную базу данных (Objectivity/DB). Эта база данных создается для выбора базовых единиц информации, оптимизации алгоритмов триггера и реконструкции событий. База данных CMS в CERN с начала 2000 г. наполняется модельными физическими событиями, подобными тем, которые предполагается исследовать на действующей установке CMS. В процессе генерации событий и последующей передачи данных в CERN происходит проверка работоспособности локальных ресурсов, корректности работы программного окружения, отрабатываются процедуры удаленного обмена большими объемами данных.

5.1. Поддержка компьютинга CMS в ОИЯИ. Поскольку в течение последних пяти лет в ОИЯИ была организована достаточно полная поддержка компьютинга для эксперимента CMS [39, 107, 108], остановимся более подробно на том, как решалась и решается проблема поддержки компьютинга CMS в ОИЯИ для того, чтобы обеспечить возможность выполнения (и продолжения уже начатых в CERN) работ в ОИЯИ в условиях, максимально приближенных к условиям компьютерной инфраструктуры CERN.

ОИЯИ отвечает за проектирование, изготовление и ввод в эксплуатацию торцевых адронных калориметров (Endcap HCAL) и передней мюонной станции ME1/1. ОИЯИ также принимает участие в работах по созданию предливневого детектора (Endcap Preshower) и программ физического анализа экспериментальных данных [109].

Если иметь в виду использование компьютерных средств, то ведутся работы по механическому проектированию, разработке электроники для указанных детекторов, их моделированию, изучению физических процессов и обработке данных с тестовых сеансов на прототипах этих детекторов. Для проведения этих работ в ОИЯИ необходимо было обеспечить доступ к экспериментальной информации, записанной на тестовых сеансах.

В 1997 г. для проведения перечисленных работ по тематике CMS был создан кластер архитектуры NIS+ из трех SunSparc-станций [107, 110], программное окружение которого соответствует программному окружению на CERN-кластере cms.cern.ch. На этом кластере сотрудниками ОИЯИ проводилась обработка данных с прототипов детекторов CMS и осуществлялись работы по моделированию физических процессов и установки. Кластер также использовался как архивный сервер для электронных и механических разработок.

В 1996 г. в ОИЯИ был создан информационный www-сервер [111, 112], который был принят как официальный web-сервер коллаборации RDMS (Russia-Dubna Member States).

Таким образом, в ОИЯИ была организована полноценная поддержка компьютинга для начальной стадии конструирования установки и тем самым подготовлена база для дальнейшего участия ученых ОИЯИ в эксперименте CMS.

В соответствии со все более широким использованием ферм персональных компьютеров для физических экспериментов в Лаборатории информационных технологий ОИЯИ в 2000 г. была создана специализированная Linux PC-ферма [39]. Программная среда фермы является полностью унифицированной с точки зрения требований экспериментов CMS и ALICE. На этой ферме начата работа в сеансах массовой генерации событий для триггера высокого уровня эксперимента CMS [39].

6. ПРОЕКТ ПО СОЗДАНИЮ РЕГИОНАЛЬНОГО ВЫЧИСЛИТЕЛЬНОГО ЦЕНТРА ДЛЯ LHC В РОССИИ

Проблема продолжения сотрудничества российских институтов в проектах на LHC после запуска ускорителя и экспериментальных установок напрямую связана с необходимостью создания условий для обработки и анализа экспериментальной информации непосредственно в России. Решить подобную задачу возможно лишь силами всех российских институтов, участвующих в проектах на LHC. Именно поэтому российские институты с 1999 г. начали работу над совместным проектом в этом направлении.

Специалисты ряда российских институтов, участвующих в проектах на LHC, создали совместный проект «Российский информационно-вычислительный комплекс для обработки и анализа данных экспериментов на большом адронном коллайдере» (РИВК-БАК) [113]. Проект был разработан в соответствии с меморандумом о создании РИВК-БАК, подписанном директорами ведущих российских физических институтов-участников LHC. Целью проекта является создание в России регионального комплекса для обработки данных экспериментов на LHC. Проект рассчитан на период до 2006 г. На начальном этапе (до 2002 г.) планируется разработка концепции комплекса и создание его прототипа. Для проведения работ были сформированы рабочие группы проекта по направлениям деятельности: создание ферм и кластеров персональных компьютеров; архивирование данных; развитие региональной сети РИВК-БАК и организация канала связи с CERN, а также по сопровождению унифицированного программного обеспечения. В ИТЭФ, ИФВЭ, НИИЯФ МГУ и ОИЯИ созданы фермы персональных компьютеров, ориентированные на LHC. Таким образом, положено начало для отработки прототипа российского регионального центра [39, 114].

ОИЯИ на протяжении уже нескольких лет является активным участником трех проектов на LHC: ALICE, ATLAS и CMS. Более 200 сотрудников института занимаются проектированием и изготовлением детекторов, участвуют в разработке физических исследований и программного обеспечения для этих установок. Как известно, вклад российской стороны в эти эксперименты очень значителен, и теперь, когда близится завершение строительства ускорителя и экспериментальных установок, важнейшим моментом является обеспечение дальнейшего полноценного участия российских ученых в экспериментах на LHC после запуска ускорителя, что может быть достигнуто лишь при достаточно полной поддержке компьютинга LHC в России.

7. ОПЫТ РАБОТЫ В ОИЯИ С СИСТЕМАМИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ И БАЗАМИ ДАННЫХ

В контексте участия ОИЯИ в создании в России регионального вычислительно-информационного комплекса для LHC важным моментом является тот факт, что в ОИЯИ накоплен значительный опыт работы по интеграции компьютерных ресурсов и повышению эффективности их использования.

Так, например, именно в ОИЯИ впервые в России использовалась система распределенных вычислений Condor [115]. (Следует заметить, что несмотря на то, что эта система была разработана более 10 лет назад, она и по сей день является достаточно удачным и не утратившим актуальность приложением в современной технологии эффективного использования ресурсов: именно система Condor нашла сейчас свое новое применение во многих проектах, относящихся к реализации grid-структур.) Это система пакетной обработки, в которой свободное процессорное время и другие свободные ресурсы серверов и рабочих станций, расположенных по всему миру, предоставляются всем участникам. Естественно, администратор каждой из станций имеет полную возможность сформулировать свое собственное понимание «свободных ресурсов». В результате участники этой динамически развивающейся системы, не теряя ничего, приобретают возможность резко ускорить обработку своих заданий, когда это становится необходимым. В 1994 г. пул рабочих станций SunSparc ОИЯИ был включен в европейский пул ресурсов с центром администрирования в Амстердаме. Этот пул, в свою очередь, был составной частью объединенного пула с центром администрирования в Университете штата Висконсин (США). К сожалению, неудовлетворительное состояние российских внешних коммуникаций на тот момент не позволило эффективно внедрить использование системы Condor, особенно для задач, требовавших большого количества обменов при обработке данных, хотя для вычислительных задач, где не требуется передача больших массивов информации, использование этой системы было вполне приемлемо даже при несовершенных

телекоммуникациях. В результате создания небольшого пула в ОИЯИ был получен доступ к 250 рабочим станциям в США и Европе, которые использовались для расчетов в физике высоких энергий.

Хотелось бы также упомянуть о том, что инсталляция порта математического обеспечения для www-сервера и нескольких клиентов была выполнена в ОИЯИ одной из первых в России [115], далее был создан www-сервер, содержавший сведения об ОИЯИ, и было издано первое в России руководство по работе со Всемирной паутиной [116].

На протяжении ряда последних лет ОИЯИ является ведущей организацией проекта БАФИЗ [117,118] по созданию и развитию распределенной сети баз данных и знаний в области ядерно-физических исследований. В рамках этого проекта еще в 1995 г. был разработан web-интерфейс для доступа к системам управления базами данных (эта разработка была одной из первых в этой области) [119].

С 1997 г. вычислительные мощности центральных серверов ОИЯИ составили основу суперкомпьютерного центра [120]. Ядром СКЦ являются параллельная ЭВМ SPP-2000 и автоматизированная ленточная библиотека ATL-2640, допускающая две технологические схемы использования: с помощью программных средств OMNIBACK (автоматическое резервное копирование для институтских ЭВМ) и OMNISTORAGE (управление мигрирующей файловой системой).

К настоящему моменту информационно-вычислительная интегрированная инфраструктура ОИЯИ включает в себя сотни серверов и несколько тысяч компьютеров, соединенных высокоскоростной многоуровневой локальной сетью технологии ATM. С точки зрения архитектурных решений в ОИЯИ имеется множество подсистем: кластеры, многомашинные комплексы, системы пакетной обработки, системы SMP, MPP и специализированной архитектуры. Поскольку информационно-вычислительная инфраструктура ОИЯИ характеризуется наличием оборудования различных фирм, разнообразием архитектуры и, соответственно, разнообразием операционных систем, то в контексте будущего использования grid-технологий необходимо осуществить интеграцию различных элементов суперкомпьютерного центра ОИЯИ и кластерных решений, включая использование дисковых кэшей и системы массовой памяти.

ЗАКЛЮЧЕНИЕ

Во второй половине 90-х годов в организации компьютинга для физических экспериментов произошли большие перемены: во-первых, наметилась четкая ориентация на использование ферм и кластеров персональных ЭВМ, во-вторых, произошел переход на ООП в разработке математического

обеспечения. Наконец, осмысление грандиозности задач, которые ставятся в организации компьютинга для строящихся на LHC установок, привело к организации специальных проектов для моделирования этого компьютинга и создания прототипов региональных центров в Европе и США. Реализация этих проектов неизбежно требует привлечения, использования и развития новейших информационно-сетевых технологий, и именно физика частиц предоставляет сейчас широкие возможности для испытаний grid-систем — мощных управляемых распределенных систем информации.

Дальнейшее развитие в России компьютинга для физических экспериментов, ориентированных на LHC, можно рассматривать как расширение и развитие существующих (см., например, [120]) вычислительных мощностей с созданием инфраструктуры, соответствующей требованиям к компьютингу этих экспериментов:

- развитие внешних сетевых коммуникаций и локальных сетей;
- создание ферм и кластеров персональных ЭВМ;
- создание хранилищ данных требуемой емкости;
- дальнейшее развитие www-информационного сервиса;
- наличие адекватного CERN оборудования и математического обеспечения для проведения телеконференций [121].

Наиболее важной, безусловно, является задача организации локальных сетей и внешних коммуникаций. Без хорошо организованных и быстрых локальных сетей в российских ядерно-физических институтах и в отсутствие быстрой связи с CERN (не менее 155 Мбит/с в самое ближайшее время и порядка 1 Гбит/с к 2005 г.) полноценное (и адекватное российскому вкладу в строительство физических установок на LHC) участие в анализе и обработке данных на действующей стадии установок на LHC представляется невозможным.

Немаловажным является освоение новых программных технологий, в том числе переход к объектно-ориентированному программированию, работа с объектно-ориентированными базами данных и современными средствами визуализации, что уже потребовало переориентации специалистов, многие годы использовавших язык Фортран, к программированию на C++, а также привлечения молодых специалистов, владеющих объектно-ориентированным подходом.

Принципиально новой и особенно актуальной становится задача освоения grid-технологий [122]. Именно освоение и применение этих технологий приведет в конечном итоге развитые страны мира к созданию глобального информационного общества XXI века. В этом контексте развитие и использование grid-технологий в России актуально не только для российского ядерно-физического сообщества, ибо возможное отставание России в области информационно-сетевых технологий может крайне негативно сказаться на многих сторонах жизни. Физика высоких энергий на данный момент дает

уникальный шанс приобщиться к этим новым и прогрессивным технологиям. В обзоре мы уделили большое внимание описанию нескольких (из множества подобных) grid-проектов в США, Европе и Японии, чтобы дать представление о степени сложности и масштабности решаемых в них задач и мотивировать необходимость организации подобных проектов в России.

СПИСОК ЛИТЕРАТУРЫ

1. *Evans L.R.* CERN AC/95-02 (LHC). Geneva, 1995.
2. GRID: a Blueprint to the New Computing Infrastructure / Ed. by Foster J., Kesselman K. San Francisco: Morgan Kaufman Publishers, 1999.
3. *Коваленко В., Корягин Д.* // Открытые системы. 1999. Т.11–12. С.10.
4. *Васенин В.* // Открытые системы. 2000. Т.12. С.36.
5. *Шевель А.* // Открытые системы. 2001. Т.2. С.36.
6. *Кореньков В., Тихоненко Е.* // Открытые системы. 2001. Т.2. С.30.
7. *Nowak M.* // Proc. of 1999 CERN School of Computing, Geneva, 2000. P.79.
8. http://www.cern.ch/MONARC
9. *Aderholz M. et al.* KEK Preprint 2000-8, CERN/LCB 2000-001. 2000.
10. http://www.pcmarkt.ch/spec.shtml
11. http://www.EU-DataGrid.org/grid/presentations/ecfa-june-_202000.ppt
12. http://www.EU-DataGrid.org/
13. http://www.globus.org
14. http://www.phys.ufl.edu/˜ avery/mre/
15. http://www.cacr.caltech.edu/ppdg
16. http://www-itg.lbl.gov/Clipper/
17. http://www.nile.cornell.edu/
18. http://www.cs.wisc.edu/condor
19. http://pcbunn.cithep.caltech.edu/
20. *Bunn J.* CMS Note IN/1999-044, CERN. Geneva, 1999.
21. *Жучков А.В., Ильин В.А., Кореньков В.В.* // Тр. Всерос. конф. «Высокопроизводительные вычисления и их приложения». М., 2000. С.227.
22. *Кореньков В.В., Тихоненко Е.А.* // Сб. тез. Всерос. конф. «INTERNET в научных исследованиях». М., 2000. С.86.
23. *Yeh G.P.* Fermilab-Conf-00/053. Fermilab, 2000.
24. http://www-cdf.fnal.gov/cd/gp.html
25. *Андреев А., Воеводин В., Жуматий С.* // Открытые системы. 2000. Т.5–6. С.15.
26. http://www.linux.org
27. http://www.redhat.com
28. *Geist P.* PVM: Parallel Virtual Machine. MIT Press, 1994.

29. http://www-unix.msc.anl.gov/mpi/index.html
30. http://hepwww.ph.qmw.ac.uk/HEPpc/
31. http://www.platform.com/platform/platform.nsf/webpage/LSF?OpenDocument
32. http://www.genias.de/products/codine/description.html
33. http://www.shef.ac.uk/uni/projects/nqs
34. htpp://pbs.mrj.com/
35. *Balashov V., Lomov A.* // CERN Comp. Newsletters. 1993. V.214. P.13.
36. http://www.microprocessor.sscc.ru
37. *Alpert M. et al.* FERMILAB-TM-2109. Fermilab, 2000.
38. http://datafarm.apgrid.org/
39. *Кореньков В.В., Мицын В.В., Тихоненко Е.А.* Сообщение ОИЯИ Р11-2001-24. Дубна, 2001.
40. *Stroustrup B.* The C++ Programming Language. Massachusetts: Addison-Wesley, 1997.
41. *Betttels J., Myers D.R.* CERN-DD/86/6. Geneva, 1986.
42. *Attwood W.* SLAC-PUB-5242. Stanford, 1990.
43. *Attwood W.* // Int. J. Mod. Phys. C. 1992. V.3. P.459.
44. *Dell'Acqua A.* CERN/DRDC/94-29. Geneva, 1994.
45. BaBar TDR — SLAC-R-95-457. Stanford, 1995.
46. *Geddes N.* // Comp. Phys. Commun. 1998. V.110. P.38.
47. *Robertson L.* // Comp. Phys. Commun. 1998. V.110. P.6.
48. ftp://ftp.slac.stanford.edu/user/pfkeb/c++class
49. http://wwwinfo.cern.ch/asd/index.html
50. http://wwwinfo.cern.ch/asd/lhc++
51. http://www.sgi.com/software
52. *Davis T.* OpenGL Programming Guide: The Official Guide to Learning OpenGl. Massachusetts: Addison-Wesley, 1993.
53. http://www.sgi.com/Technology/Inventor.html
54. http://www.tgs.com
55. IRIS Explorer User Guide. 1995.
56. http://www.objy.com
57. http://wwwinfo.cern.ch/asd/lhc++/Objectivity/index.html
58. http://www.nag.co.uk/
59. http://wwwinfo.cern.ch/asd/lhc++/Gemini
60. http://wwwinfo.cern.ch/asd/lhc++/clhep/index.html
61. http://wwwinfo.cern.ch/asd/lhc++/htlguide/htl.html
62. http:/home.cern.ch/~ couet/HEPInventor.doc/
63. http://www.cern.ch/Physics/Workshops/hepvis/
64. http://wwwinfo.cern.ch/asd/lhc++/HepExplorer/index.html
65. http://wwwinfo.cern.ch/asd/lhc++/HepFitting/

66. http://root.cern.ch/
67. http://b.home.cern.ch/b/billmei/www/Bsp
68. http://AliSoft.cern.ch/offline
69. *Rademakers F.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.185.
70. http://www.star.bnl.gov/STARAFS/comp/root/index2.html
71. *Fine V., Nevski P.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.143.
72. http://www-b0.fnal.gov:8000/consumer/framework/
73. *Sexton-Kennedy E. et al.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.161.
74. http://www.mpi-hd.mpg.de/herab/clue/
75. http://www.gsi.de/computing/root/
76. http://root.cern.ch/root/Atlfast.html
77. *Snow J. et al.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.165.
78. http://www.gsi.de/computing/root/OracleAccess.htm
79. http://www.phenix.bnl.gov/WWW/publish/onuchin/rooObjy/
80. http://hpcf.nersc.gov/storage/hpss/
81. http://www.unitree.com/
82. http://www.managementsoftware.hp.com/prodcategories/storagemgmt/index.asp
83. http://www.technoserv.ru
84. http://wwwinfo.cern.ch/pdp/castor/
85. *Barring O., Baud J., Durand J.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.365.
86. http://www-isd.fnal.gov/enstore/index.html
87. http://www.quadrics.com/eurostore
88. *Quarrie D. et al.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.398.
89. http://wwwcn.cern.ch/asd/cernlib/rd45/index.html
90. RD45. A Persistent Object Manager for HEP. CERN/DRDC, 94-30. Geneva, 1994.
91. *Silvestris L., Innocente V.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.423.
92. *Rolli S. et al.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.436.
93. LHCb Computing Group. // Proc. of CHEP'2000, Padova, Italy, 2000. P.431.
94. http://www-itg.lbl.gov/DPSS
95. CMS Collaboration. The Compact Muon Solenoid. Technical Proposal. CERN/LHCC 94-38. Geneva, 1994.
96. CMS Collaboration. CMS Computing Technical Proposal. CERN/LHCC 96-45. Geneva, 1996.
97. *Taylor L.* CMS IN/1999-032, CERN. Geneva, 1999.
98. http://cmsdoc.cern.ch/cmsim/cmsim.html
99. http://cmsdoc.cern.ch/orca
100. *Stickland D.* CMS IN/1999-035, CERN. Geneva, 1999.
101. *Innocente V., Stickland D.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.56.
102. http://cmsdoc.cern.ch/cmsoo/projects/IGUANA

103. *Alverson G., Gaponenko I., Taylor L.* CMS IN-1999-042, CERN. Geneva, 1999.

104. *Kalle Dalheimer M.* Programming with Qt. Köln. O'Reilly Verlag GmbH&Co.KG, 1999.

105. http://www.troll.no/qt/opengl.html

106. http://cmsdoc.cern.ch/cgi-cms/scrampage

107. *Golutvin I. et al.* JINR Commun. D11-98-122. Dubna, 1998.

108. *Pose R., Tikhonenko E.* CMS Document 1996-213, CERN. Geneva, 1996. P.118.

109. RDMS. Participation in CMS Collaboration: RDMS Project, CMS Document 1996-085, CERN. Geneva, 1996.

110. *Kadykov V. et al.* CMS Document 1997-168, CERN. Geneva, 1997. P.239.

111. *Tikhonenko E.* CMS Document 1996-213, CERN. Geneva, 1996, P.121.

112. http://sunct2.jinr.dubna.su

113. http://theory.npi.msu.su/ ilyin/RIVK-BAK

114. *Kodolova O., Tikhonenko E.* CMS Conf. talk 2000-030, CERN. Geneva, 2000.

115. *Кореньков В.В., Мицын В.В., Окрашец К.Ф.* // Краткие сообщ. ОИЯИ. 1995. № 2[70]. С.5.

116. *Окрашец К.Ф.* Сообщение ОИЯИ Р10-95-51. Дубна, 1995.

117. http://dbserv.jinr.ru

118. *Кореньков В.В., Никонов Э.Г.* // Тр. Всероссийской науч. конф. «Математика. Компьютер. Образование». М., 2000. Ч.1. С.259.

119. *Окрашец К.Ф.* Сообщение ОИЯИ Р10-95-50. Дубна, 1995.

120. *Korenkov V.* // Proc. of Intern. Conf. HIPER'98, Zurich, Switzerland, 1998. P.224.

121. http://vrvs.cern.ch

122. *Avery P.* // Proc. of CHEP'2000, Padova, Italy, 2000. P.648.