**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2014 Т. 6 № 5 С. 635−642**

Ки&М

МОДЕЛИ В ФИЗИКЕ И ТЕХНОЛОГИИ

УДК: 004.414.23, 519.876.5

Моделирование грид-облачных сервисов проекта NICA как средство повышения эффективности их разработки

В. В. Кореньков[a], А. В. Нечаевский, Г. А. Ососков, Д. И. Пряхина, В. В. Трофимов, А. В. Ужинский

Объединенный институт ядерных исследований,
Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: [a] symsim@jinr.ru

Получено 10 октября 2014 г.

Описана новая система моделирования грид- и облачных сервисов, ориентированная на повышение эффективности разработки системы хранения и обработки данных ускорительного комплекса НИКА. В системе реализован подход учета качества работы уже функционирующей системы при проектировании ее дальнейшего развития за счет объединения самой программы моделирования с системой мониторинга реального (или модельного) грид-облачного сервиса через специальную базу данных. Приведен пример применения программы для моделирования достаточно общей облачной структуры, которая может быть также использована и вне рамок физического эксперимента.

Ключевые слова: имитационное моделирование, грид, облака, хранение данных, оптимизация, мониторинг

Grid-cloud services simulation for NICA project, as a mean of the efficiency increasing of their development

V. V. Korenkov, A. V. Nechaevskiy, G. A. Ososkov, D. I. Pryahina, V. V. Trofimov, A. V. Uzhinskiy

Joint institute for nuclear researches, Laboratory of Information Technologies, Joliot-Curie, 6, Moscow reg., Dubna, 141980, Russia

Abstract. — A new grid and cloud services simulation for NICA accelerator complex data storage and processing system are described. This system is focused on improving the efficiency of the grid-cloud systems development by using work quality indicators of some real system to design and predict its evolution. For these purpose the simulation program are combined with real monitoring system of the grid-cloud service through a special database. An example of the program usage to simulate a sufficiently general cloud structure, which can be used for more common purposes, is given.

Keywords: simulation, grid, cloud, data storage, optimization, monitoring

Citation: *Computer Research and Modeling*, 2014, vol. 6, no. 5, pp. 635–642 (Russian).

Работа выполнена при поддержке гранта РФФИ № 14-07-00215.

Введение

В различных областях деятельности для обработки информации существует множество вычислительных систем различного масштаба. Наибольший интерес для исследования представляют грид- и облачные вычислительные системы, обрабатывающие сверхбольшие объемы данных, примером которых может служить грид-облачная система распределенной обработки данных, разрабатываемая в Объединенном институте ядерных исследований для эффективной поддержки компьютинга на ускорительном комплексе НИКА [Сисакян А.Н., Сорин А.С., 2011]. Этот комплекс, создаваемый в ОИЯИ, 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, которые подлежат хранению и анализу в разрабатываемом в настоящее время в ОИЯИ Tier1 центре Всемирной сети распределенных вычислений — Worldwide LHC Computing Grid (WLCG) [The Worldwide..., 2014].

В настоящее время при проектировании грид-систем используется подход, когда задача создания модели и формулировки рекомендаций по построению выполняется однократно при проектировании системы. Однако эксперименты продолжаются годами и десятилетиями, одновременно с эксплуатацией системы происходит ее развитие, не только качественное, но и количественное. При эволюции WLCG произошло качественное изменение систем хранения информации, а вместо планируемых трех уровней обработки данных появилось четыре. Таким образом, даже при значительных усилиях, вложенных на этапе проектирования в понимание конфигурации систем и их количественных характеристик, невозможно развивать систему без дополнительных исследований. Разработчики и эксплуатирующие организации сталкиваются с проблемой прогнозирования поведения системы после проведения планируемых модификаций.

Моделирование системы позволяет ответить на ряд вопросов. 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 (устойчивость к повреждениям и удалениям) на протяжении всего жизненного цикла проекта, обеспечить распределение ресурсов между различными группами пользователей, выбрать алгоритмы обработки и запуска задач и многое другое.

Таким образом, требуется создание методологии и программного окружения, позволяющего моделировать системы на постоянной основе, прогнозировать поведение системы при значительных изменениях.

Объединив моделирование и мониторинг в рамках одного программного пакета, можно добиться существенного снижения эксплуатационных затрат и вложений в увеличение мощности с целью сохранения скорости получения результата экспериментов, при постоянном повышении потока данных.

Анализ средств моделирования

Говоря о том, какую технологию моделирования применить, следует учесть, что возможность применения аналитических моделей для рассматриваемых задач ограничена по следующим соображениям. Существует несколько подходов при аналитическом моделировании грид- и облачных систем, которые можно сгруппировать в два типа:

– система рассматривается как многоканальная система массового обслуживания, с состояниями, управляемыми марковским процессом, с ограничениями на распределения входных потоков и на дисциплины обслуживания, вызванными теоретическими предпосылками;

– система рассматривается как динамическая стохастическая сеть, описываемая системами уравнений, позволяющими учитывать как маршрутизацию, так и распределение ресурсов

в сети, причем изучению подлежат равновесные и неравновесные состояния сети [Попков, 2003].

Оба подхода выдают результат моделирования, как правило, в виде асимптотических распределений и в силу ограниченных теоретических предпосылок не могут быть применены для моделирования конкретных сложных компьютерных сетей многоуровневой архитектуры с реальными распределениями входных потоков заданий, сложной многоприоритетной дисциплиной их обслуживания и динамическим распределением ресурсов. Поэтому авторы считают правильным использовать имитационное моделирование.

На сегодняшний день существуют различные программные инструменты имитационного моделирования грид-систем и облаков [Кореньков, Нечаевский, 2009; Кореньков, Муравьев, Нечаевский, 2014]. Например, GridSim [GridSim..., 2012] — 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. Предлагаемое нами программное приложение основано на расширении классов GridSim и их объединении в программу, которая моделирует обработку потока заданий грид-структурой, обладающей заданными ресурсами и дисциплиной их резервирования и использования.

В последнее время выдвигается идея интеграции в грид-структуры центров, построенных по принципу облачных вычислений, а также реализации служб грид на оборудовании «облачных» центров. Поэтому методы и средства, которые разрабатываются в рамках проекта, допускают моделирование объединения в грид-структуру центров, имеющих облачную архитектуру.

В качестве планировщика потока заданий использованы алгоритмы ALEA [Klusacek et al., 2008], позволяющие выбрать оптимальную дисциплину планирования потока заданий на компьютерных кластерах, объединенных в структуру. Анализировать ход выполнения моделируемого эксперимента можно по таким критериям:

- количество процессорных элементов — запрошенных, используемых и доступных;
- загруженность кластеров в процентном соотношении по каждому часу и дню;
- количество задач, которые ждут выполнения и выполняющихся;
- среднее использование процессорных элементов кластеров за каждый час;
- процент отказов ресурсов грид-системы.

Эти критерии наиболее рационально подходят для анализа алгоритмов распределения и выполнения заданий. Следует, однако, подчеркнуть, что планировщик заданий ALEA не предназначен для моделирования обработки данных.

Описание подхода к моделированию

Постоянное развитие современных грид-систем требует непрерывных корректировок большинства параметров моделирования. Это необходимо для прогнозирования поведения системы при значительных ее изменениях. Для корректировки параметров предлагается использовать статистику эксплуатации системы, получаемую на основе имеющихся программных средств ее мониторинга.

В связи с этим возникают две проблемы:

- обеспечение совпадения исходных данных для модели с реальными;
- проверка адекватности моделирования, т. е. доказательство того, что моделирование произведено корректно и поведение модели не отличается от поведения реальной системы.

Подход авторов состоит в следующем.

1. Если речь идет о модернизации существующей установки обработки, то правильно использовать накопленные данные. К примеру, в проекте WLCG имеются как глобальные, так и специализированные под конкретные эксперименты системы мониторинга и аккаунтинга (http://dashb-wlcg-transfers.cern.ch/ui/, http://dashb-atlas-task-prod.cern.ch/templates/task-prod,

http://panda.cern.ch:25880/server/pandamon/query и пр.). При этом результаты моделирования обработки потока заданий должны совпадать в пределах погрешности с результатами мониторинга прохождения того же потока заданий в системе.

2. Для новых установок эта проблема разрешается выдвижением гипотез о типах потоков входной информации, их параметрах и процедурах их обработки с последующим моделированием как самих входных потоков, так и процессов их обработки. Такие гипотезы можно сформулировать на основании данных мониторинга подобных систем (оценивая интенсивность и основные характеристики потоков заданий и файлов). Обработка результатов моделирования заключается в анализе распределения времени событий, которые генерируются при обработке входного потока данных. Затем эти распределения сравниваются с результатами, полученными из мониторинга существующей системы.

Таким образом, модель должна рассматриваться как неотъемлемая часть системы обработки данных, а данные мониторинга — как входные для моделирования. Это позволит принимать более обоснованные проектные решения при развитии системы.

Система мониторинга и учета ресурсов предназначена для отслеживания текущего состояния ресурсов, заданий и других объектов в грид-системе. Среди основных задач мониторинга отметим следующие:

- непрерывное наблюдение за состоянием грид-сервисов;
- получение информации о вычислительных ресурсах (количество вычислительных узлов для выполнения задач, архитектура вычислительной системы, установленное программное обеспечение, доступные специализированные программные пакеты), потребленное процессорное время и прочее;
- данные о доступе виртуальных организаций к ресурсам и использовании ими квот на вычислительные ресурсы;
- мониторинг выполнения вычислительных заданий и задач (запуск, изменение состояния, коды завершения и т. п.).

Среди параметров мониторинга, необходимых для последующего моделирования, наиболее существенными являются следующие:

1) число задач (симуляция, анализ, реконструкция), поступающих в систему;
2) объем используемой оперативной памяти;
3) использованное процессорное время;
4) число обработанных событий;
5) время расчета задачи;
6) объем используемых данных.

Схема программы SyMSim (Synthesis of Monitorng and Simulation — Синтез мониторинга и моделирования), реализующей идею синтеза процессов мониторинга и моделирования, представлена на рисунке 1.

Данные мониторинга реальной грид-системы поступают в базу данных следующим образом: задание отправляется на сервер (*1*), далее сайт-пилот запрашивает задание на обработку (*2*), сервер отправляет задание на исполнение (*3*), информация о выполнении задания поступает в базу данных StatDB (*4*). На основе данных мониторинга пользователь задает входные параметра модели (*5*) и потока заданий (*6*), далее модель обрабатывает задания (*7*, 8). Результаты работы модели доступны пользователю для дальнейшего анализа.

Для иллюстрации возможностей разработанной программы ниже приведен пример ее применения для оптимизации простой облачной структуры.

Пример использования программы SyMSim

Предполагается что моделируемая структура предназначена для обработки данных физического эксперимента, но другие структуры, связанные с хранением и обновлением больших массивов цифровой информации, также могут быть смоделированы.

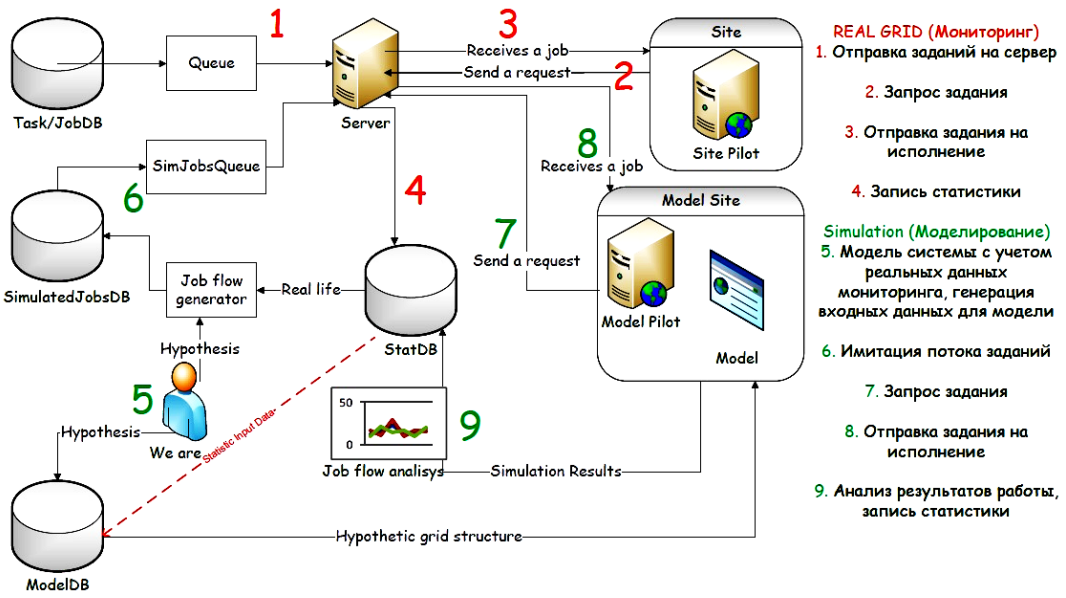

Рис. 1. Схема программы SyMSim моделирования системы распределенных вычислений с учетом данных мониторинга этой системы

Постановка задачи

Исследуемая структура облачного хранилища состоит из ленточного робота, дискового массива, кластера процессоров. Речь идет о хранении данных в роботизированных библиотеках с тысячами картриджей с магнитными лентами, которые робот автоматически достает или устанавливает в один или несколько драйвов, т. е. устройств чтения-записи. Предположим, что каждый слот — это отдельный процессор. Также структура, которая рассматривается в качестве примера, имеет генератор данных (детектор), буфер для хранения данных, сервер, который передает информацию, и очередь заданий на обработку. Схема прохождения заданий через SyMSim представлена на рисунке 2.

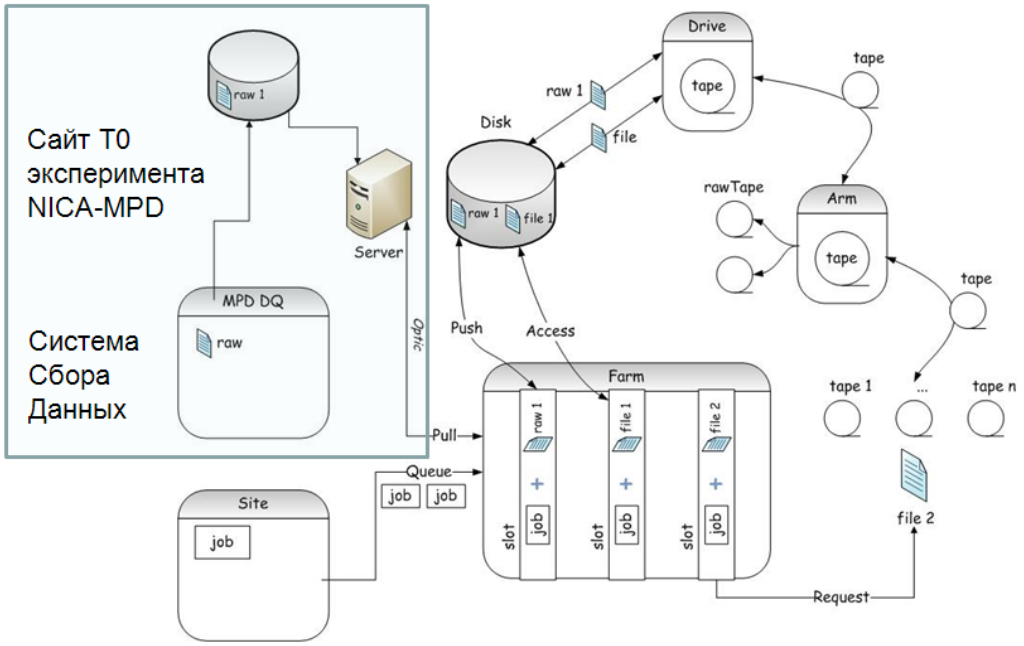

Рис. 2. Схема прохождения заданий через SyMSim

Задание начинает выполняться, если есть свободный слот и все файлы доступны на дисковом хранилище облака. Если файл хранится в роботизированной библиотеке, задание резервирует слот, но выполнение задерживается до момента его загрузки на диск. Процесс перемещения файла из библиотеки в дисковое хранилище включает в себя операцию помещения ленточного картриджа на драйв, которую выполняет рука робота, монтирования файловой системы картриджа на драйве и записи файла на диск.

Бюджетные ограничения по стоимостям драйвов и слотов в условных единицах отражены ниже в таблице 1. Общий бюджет ограничен суммой в 1000 условных единиц.

В качестве критериев оценки проектируемого кластера выбираются следующие параметры: время прохождения тестового потока из 99 заданий и уровень загрузки процессоров облака.

Моделирование должно помочь разработчику облачного кластера при заданной стоимости драйвов и слотов найти оптимальное их соотношение, которое минимизировало бы время прохождения и не давало упасть загрузке процессоров при указанном ограниченном бюджете.

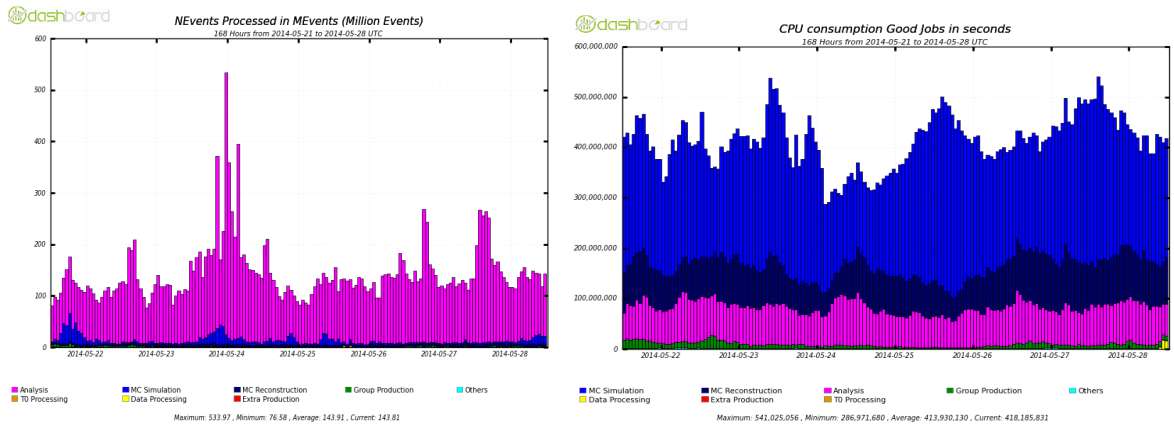

Рис. 3. Динамика изменения параметров мониторинга для эксперимента Atlas

На основе доступной статистики по эксперименту Atlas (динамика распределения числа физических событий (в млн) по группам решаемых задач (анализ, симуляция, обработка) и динамика распределения загрузки процессоров (число выполняемых заданий в сек.) по типам решаемых задач (см. рис. 3)), видно, что задачи делятся на три типа: моделирование, реконструкция и анализ. Для определения входных параметров модели выполняется статистический анализ данных эксперимента Atlas, результаты которого позволяют сделать предположение о распределении потока заданий. Параметры потока заданий, который необходимо генерировать, заносятся в веб-форму (см. рис. 4). Далее генерируется поток заданий трех различных типов с параметрами (количество событий, процессорное время и память), близкими к подобным параметрам для эксперимента Atlas, и результат записывается в базу данных.

Описание процесса моделирования

Каждый вычислительный эксперимент заключался в «выполнении» одного и того же потока заданий при различных параметрах облака. Определялась зависимость времени выполнения потока заданий от конфигурации системы обработки. Количество слотов и драйвов варьировалось с целью определить соотношение, когда добавление оборудования более не ускоряет процесс обработки заданий. Уровень загрузки кластера определялся как $\mathbf{W = T_{100}/Ta}$, где T_{100} — процессорное время выполнения пакета, а Ta — астрономическое время. Результаты этих экспериментов отображены в таблице 1.

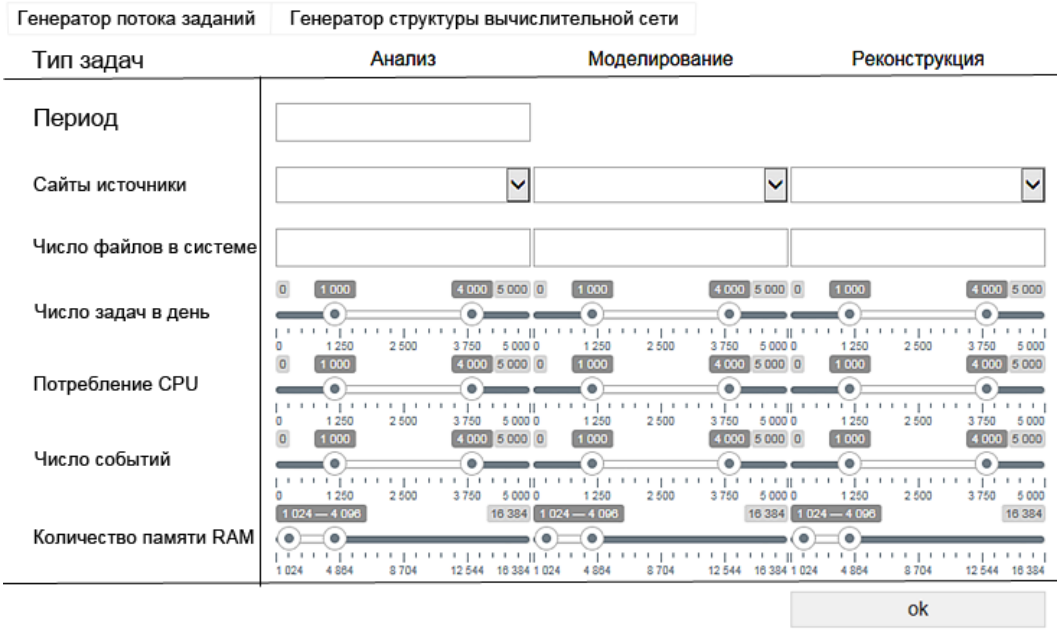

Рис. 4. Веб-интерфейс для генерации потока заданий

Таблица 1

№	Кол-во слотов	Цена слота	Драйвы	Цена драйва	Сумма затрат	Время счета	Загрузок лент	Уровень загрузки
1	23	30	6	50	990	50123	34	0,591
2	21	30	7	50	980	52659	31	0,616
3	20	30	8	50	1000	49101	27	0,694
4	18	30	9	50	990	47419	22	0,799
5	16	30	10	50	980	48517	20	0,878
6	15	30	11	50	1000	50441	16	0,901
7	13	30	12	50	990	52689	13	0,995

Мы видим, что при большом количестве процессоров загрузка кластера падает, поскольку процессоры простаивают в ожидании монтирования кассет с данными на драйвы. Следовательно, надо выбирать оптимальное соотношение количества процессоров и драйвов.

На рисунке 5 отображен вариант 4 конфигурации хранилища из таблицы 1. Видно, что конфигурация, обеспечивающая минимальное время исполнения, должна состоять из 18 вычислительных процессоров и 9 драйвов-загрузчиков.

Результаты моделирования по критерию минимального времени прохождения задания при достаточно высокой загрузке процессоров могут служить обоснованием выбора конфигурации облачного кластера и аргументом в пользу покупки или отклонения более дорогого оборудования, хотя не следует забывать, что на выбор конфигурации также влияют и другие соображения: надежность, перспективы развития, величина резерва и т. д.

Заключение

Разработка грид-облачных систем сбора, передачи и распределенной обработки информации требует тщательного моделирования, которое будет эффективным, только если оно будет учитывать качество работы уже функционирующей системы в прогнозах на ее дальнейшее развитие. Новизна подхода, предложенного авторами, состоит в соединении процессов моделирования и мониторинга в рамках одного проекта.

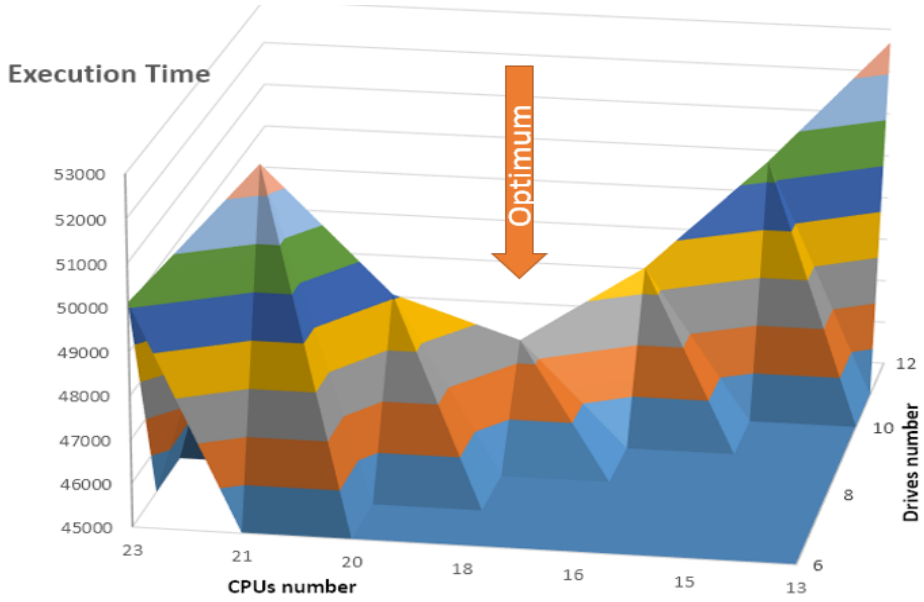

Рис. 5. Зависимость времени выполнения заданий от количества драйвов и слотов

Предложенный подход к моделированию и анализу вычислительных грид-облачных структур инфраструктуры эксперимента НИКА основан на учете данных мониторинга, используемых затем для динамической коррекции параметров моделирования.

Дальнейшее развитие системы предполагает внесение дополнений для моделирования оригинальных алгоритмов назначения пулов dCache. Структура моделирующей программы достаточно общая, чтобы в дальнейшем заменять упрощения, допущенные на предварительном этапе, на более реальные условия. Также необходимо провести полномасштабные испытания модели с целью выявления ее ошибок.

Список литературы

Кореньков В. В., Муравьев А. Н., Нечаевский А. В. Пакеты моделирования облачных инфраструктур // Системный анализ в науке и образовании. — 2014. — Вып. 2. — Дубна,

Попков Ю. С. Макросистемы и grid-технологии: моделирование динамических стохастических сетей // Проблемы управления. — 2003. — № 3.

Сисакян А. Н., Сорин А. С. Многоцелевой детектор–MPD для изучения столкновений тяжелых ионов на ускорителе NICA (Концептуальный дизайн-проект), версия 1.4. [электронный ресурс]. — 2011. — URL: http://nica.jinr.ru/files/CDR_MPD/MPD_CDR_ru.pdf (дата обращения: 16.08.2014).

GridSim: A Grid Simulation Toolkit For Resource Modelling And Application Scheduling For Parallel And Distributed Computing [электронный ресурс] // The University of Melbourne, Australia. — 2014. — URL: http://www.gridbus.org/gridsim/ (дата обращения: 06.09.2014).

Klusacek D., Matyska L., and Rudova H. Alea — Grid scheduling simulation environment // In 7th International Conference on Parallel Processing and Applied Mathematics (PPAM 2007). Vol. 4967 of LNCS. P. 1029–1038. Springer, 2008.

Korenkov V. V., Nechaevskiy A. V. DataGrid simulation packages // System Analysis in Science and Education (Online), ISSN: 2071–9612, Issue 1, 2009.

The Worldwide LHC Computing Grid [электронный ресурс] // CERN, Switzerland. — 2014. — URL: http://home.web.cern.ch/about/computing/worldwide-lhc-computing-grid (дата обращения: 17.09.2014).