ТЕХНОЛОГИЯ iPSE СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И ОБРАБОТКИ ДАННЫХ

УДК 004.75;004.453

С. В. Марьин, С. В. Ковальчук

СЕРВИСНО-ОРИЕНТИРОВАННАЯ ПЛАТФОРМА ИСПОЛНЕНИЯ КОМПОЗИТНЫХ ПРИЛОЖЕНИЙ В РАСПРЕДЕЛЕННОЙ СРЕДЕ

Разработана интеллектуальная платформа управления исполнением композитных приложений в распределенной вычислительной среде на основе сервисно-ориентированной архитектуры. Платформа адаптирована для использования в составе программного комплекса HPC-NASIS для многомасштабного моделирования в нанотехнологиях.

Ключевые слова: *интеллектуальное управление, планирование, параметрическая модель, эвристика, сервисно-ориентированная архитектура.*

Введение. Методология электронной науки (eScience) предусматривает обеспечение коллективного доступа исследователей к разнообразному программному инструментарию компьютерного моделирования и обработки данных посредством распределенной вычислительной среды. Это, в свою очередь, требует развития специализированного класса программного обеспечения — платформ распределенных вычислений (ПРВ) для разработки и исполнения *композитных* приложений, состоящих из нескольких взаимодействующих прикладных сервисов, описываемых в форме потока заданий (workflow, WF). При этом ПРВ в общем случае должна обеспечивать не только исполнение композитных приложений на заданном наборе вычислительных систем, но и *управление* процессом исполнения отдельных сервисов в составе WF с целью обеспечения эффективного использования ресурсов и минимизации общего времени решения задачи. Процесс управления сводится к построению расписания, обеспечивающего синхронизацию работы отдельных сервисов в условиях неоднородности вычислительных ресурсов и стохастической изменчивости параметров коммуникационных сетей и вычислительных систем.

В настоящее время исследование проблемы управления композитными приложениями в форме WF в распределенных вычислительных средах связано с развитием нескольких параллельных направлений. В качестве иллюстрации в табл. 1 приведены характеристики ведущих отечественных и зарубежных решений в данной области [1]. В таблице приведены сведения о модели WF (абстрактный, конкретный [2]), а также способе задания потока: графический, текстовый либо автоматический (по неполному пользовательскому описанию). Также приведены

характеристики планировщика, осуществляющего исполнение WF, и указана целевая функция планирования (время исполнения или квоты на использование ресурсов).

Результаты анализа табл. 1 в целом демонстрируют, что в настоящее время еще не сложилось единого подхода к вопросам управления процессом исполнения композитного приложения в распределенной среде.

Таблица 1

Характеристики программных платформ управления композитными приложениями в распределенной вычислительной среде

Название системы	Компоновка WF		Планирование			
	модель	составление	архитектура планировщика	уровень планирования	схема	цель
CAEBeans	WF фиксирован*		Централизованная	Задача	Динамическая	Время
СУС ИСА РАН	Абстрактная	Графическое	Планирование вырожденное: каждому вычислительному компоненту априори поставлен в соответствие ровно один вычислительный ресурс			
GridMD	WF задается непосредственно в тексте запускаемой программы (C++)		Используется планировщик той Грид-системы, на которой запускается приложение			
DAGMan	Абстрактная	Текстовое	Централизованная	Задача	Динамическая	Время
Pegasus	Абстрактная	Текстовое Автоматически	Централизованная	Задача WF	Статическая от пользователя Динамическая	Время
Triana	Абстрактная	Графическое	Распределенная	Задача	Динамическая	Время
ICENI	Абстрактная	Текстовое Графическое	Централизованная	WF	Динамическая с предсказанием	Время Квоты
Taverna	Абстрактная Конкретная	Текстовое Графическое	Централизованная	Задача	Динамическая	Время
GrADS	Абстрактная	Текстовое	Централизованная	Задача WF	Динамическая с предсказанием	Время
GridFlow	Абстрактная	Графическое Текстовое	Иерархическая	Задача	Статическая	Время
UNICORE	Конкретная	Графическое	Централизованная	**	Статическая от пользователя	**
Gridbus workflow	Абстрактная Конкретная	Текстовое	Иерархическая	Задача	Статическая от пользователя Динамическая	Квоты
Askalon	Абстрактная	Графическое Текстовое	Распределенная	WF	Динамическая Динамическая с предсказанием	Время Квоты
Karajan	Абстрактная	Графическое Текстовое	Централизованная	**		
Kepler	Абстрактная Конкретная	Графическое	Централизованная	**		

* В системе CAEBeans WF задается разработчиком конечного комплекса и остается фиксированным для пользователя.

** Архитектура системы подразумевает явную реализацию части стратегии планирования разработчиком конечного комплекса.

Так, часть решений требует явного задания расписания исполнения или использует результаты статического планирования. Вместе с тем ряд платформ (например, Askalon, GrADS, ICENI) позволяет не только динамически планировать процесс исполнения, но и прогнозировать время исполнения с целью дальнейшего мониторинга хода решения задачи. Однако адекватность и достоверность такого прогноза дискуссионны в силу того, что он основывается только на экстраполяции фактических данных измерений времени расчетов и не использует в полной мере априорных знаний предметной области о производительности отдельных предметно-ориентированных сервисов в составе композитного приложения.

В настоящей работе предложен новый подход к управлению процессом исполнения композитного приложения в распределенной среде в условиях неопределенности с использо-

ванием экспертных знаний в форме параметрических моделей производительности сервисов заданной предметной области.

Модель процесса исполнения композитного приложения в распределенной вычислительной среде. Подход к решению задачи управления композитными приложениями развивается в рамках концепции iPSE [3]. Концепция предусматривает такой способ описания сервисов в распределенной среде, когда уже на этапе создания сервисной оболочки разработчики прикладных сервисов предоставляют информацию не только об интерфейсах их взаимодействия, но и о характеристиках их производительности. Фактически эта информация также представляет собой экспертное знание, заданное в форме уравнения (параметрической модели) или табличной функции (профиля приложения). Эффективное взаимодействие сервисов в этом случае организуется самой оболочкой управления, которая выполняет операцию логического вывода (строит субоптимальное расписание) на основе знаний о производительности, заложенных в функциональных сервисах, и данных о функционировании распределенной системы в целом, получаемых посредством ее мониторинга в режиме реального времени. Это позволяет выбрать субоптимальную схему исполнения WF за счет управления распределением отдельных сервисов на ресурсах, способами их распараллеливания и маршрутами передачи данных.

Формальный механизм построения описания композитного приложения сводится к последовательности преобразований описания абстрактного WF в конкретный (или частично-конкретный) WF. В качестве модели абстрактного WF выступает ориентированный ациклический граф

$$W_a = \{w_a = (V_a, E_a)\},$$

где множество вершин V_a — решаемые подзадачи, а множество ребер E_a — зависимости между ними по данным. Промежуточным этапом построения приложения является частично-конкретный WF:

$$\begin{aligned} &W_i = \{(w_i = (V_i, E_i), state, resource)\}, \\ &state : V_i \to \{done, running, scheduled, not\_scheduled\}, \\ &resource : V_i \to C \cup \{\varnothing\}, \end{aligned} \tag{1}$$

где $state$ — функция отображения множества решаемых подзадач на множество состояний планирования, включающего такие состояния, как „выполнено“, „запущено“, „спланировано“, „не спланировано“; $resource$ — функция отображения множества решаемых задач на множество доступных ресурсов C (в случае, если задача находится в состоянии, отличном от „не спланировано“); i — шаг частично-конкретного WF.

Для составления расписания используется процедура планирования, которая может быть представлена в виде функции следующего вида:

$$sched : W_i \times T_0' \times H \to W_i, \tag{2}$$

где T_0' — множество, содержащее характеристики времени исполнения основных сервисов в составе WF, H — характеристики распределенной среды. Ход исполнения WF в целом может быть представлен в виде последовательности частично-конкретных WF:

$$\begin{aligned} &W_c(w_a \in W_a, sched, t_0', h \in H) = \{(w_i)\}, \\ &w_0 = \{w_a, state(v) = not\_scheduled, resource(v) = \varnothing\}, \\ &w_i = sched(w_{i-1}, t_0', h), i > 0, \end{aligned} \tag{3}$$

при этом функция оценки времени окончания счета на вычислительном ресурсе t_0' (как основная характеристика процесса синхронизации) представляет собой отображение вида

$$t_0' : C \to R^+. \tag{4}$$

Значения t_0' могут быть получены различными способами, в том числе путем профилировки. Однако в рамках концепции iPSE они интерпретируются как априорные знания предметной области, формой представления которых являются параметрические модели производительности, ассоциированные с доступными вычислительными сервисами предметной области. На рис. 1 приведены графики, иллюстрирующие основные аспекты построения параметрических моделей производительности на примере трех сервисов в области квантовой химии, реализуемых вычислительными пакетами GAMESS, ORCA и MOLPRO (*1*, *2* и *3* соответственно).

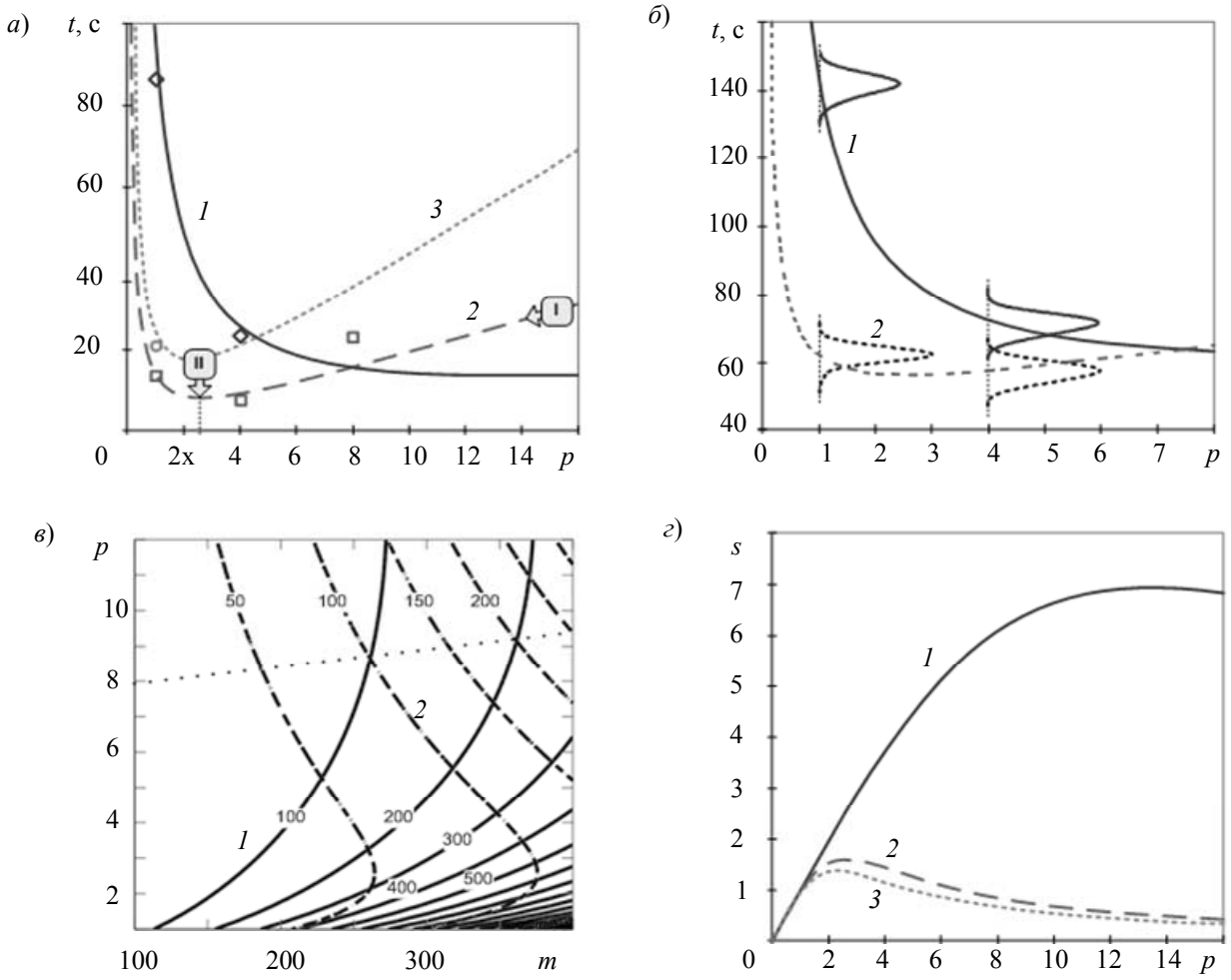

Рис. 1

На рис. 1, *а* приведен пример модели зависимости времени решения задачи t от количества вычислительных ядер p, на которых она распараллелена. Из соображений минимизации времени работы и с учетом ограничения числа вычислителей может быть произведен предварительный выбор вычислительного сервиса (I на рис. 1, *а*): использование пакета ORCA (I), функционирующего на двух вычислителях (II). На рис. 1, *б* представлено распределение времени работы пакетов (реализуемых сервисами), полученное на основании экспериментов в среде распределенных вычислений. Время исполнения учитывает накладные расходы на запуск сервиса в распределенной среде, что выражается в параллельном сдвиге графиков вдоль

оси ординат, по сравнению с рис. 1, *а*. Кроме того, пересечение распределений при $P = 4$ свидетельствует о неоднозначности решения, построенного по детерминированным моделям производительности. На рис. 1, *в* приведен график, иллюстрирующий зависимость времени работы пакетов (составляющих основу сервисов) от двух величин: количества базисных функций *m* (параметр предметной области) и количества вычислительных ядер (технический параметр). Все пространство изменения этих переменных можно разделить на области, характеризующиеся минимизацией времени при использовании какого-либо из пакетов (что и является критерием выбора). На рис. 1, *г* приведены графики производного параметра *s* (параллельного ускорения), получаемого в процессе моделирования. Как можно заметить, выбор по этому параметру (максимизация ускорения) привел бы к иным результатам (выбору пакета *1* — GAMESS). Как следствие, одной из задач, решаемых при построении схемы выполнения, является корректное определение критериев оптимизации в соответствии с потребностями пользователя.

Интеллектуальная технология планирования процесса исполнения композитного приложения. Параметрические модели производительности позволяют эффективно описывать лишь характеристики отдельных прикладных сервисов в составе композитного приложения. Определение времени работы WF в целом требует использования специфических подходов, основанных на численном построении алгоритмов планирования на основе различных эвристик, входными данными для которых, в соответствии с (2), (4), являются значения времени работы отдельных сервисов. Для исследования эффективности решения задачи управления процессом исполнения композитного приложения в распределенной среде рассмотрены эвристические алгоритмы планирования MaxMin, MinMin и Sufferage [4]. В результате анализа, проведенного посредством имитационного моделирования, было установлено, что в реальных распределенных системах, вследствие наличия стохастических факторов в изменчивости характеристик вычислительных ресурсов и коммуникационных каналов, возможно только интервальное сопоставление различных сценариев исполнения; при этом в зависимости от конкретного состояния среды может выигрывать та или иная эвристика. Таким образом, нельзя однозначно декларировать целесообразность использования того или иного алгоритма планирования, и необходимо в каждом конкретном случае рассматривать конкурирующие эвристики, вводя при этом критерии их ранжирования. Это позволяет обосновать общую процедуру планирования исполнения композитного приложения в распределенных вычислительных средах в рамках концепции iPSE (рис. 2).

Процедура включает в себя следующие этапы:

— формализация композитного приложения: формирование структуры абстрактного WF исходя из пользовательского описания, состава данных Ξ и ограничений на режимы исполнения отдельных сервисов;

— определение актуальных параметров распределенной среды (состава и текущих характеристик доступных ресурсов) с использованием инструментов мониторинга вычислительных ресурсов;

— формирование набора активных фактов: оценка характеристик производительности отдельных прикладных сервисов по параметрическим моделям (как форме представления знаний, ассоциированных с элементами WF), а также определение накладных расходов, связанных с вызовом сервисов (T_C), передачей (T_N) и конвертированием (T_D) данных;

— имитационное моделирование сценариев исполнения WF на основе набора конкурирующих эвристик: с использованием априорных знаний о стохастической изменчивости параметров распределенной среды методом Монте-Карло генерируются модельные ансамбли вариантов исполнения композитного приложения;

— интервальное оценивание: по каждой конкурирующей эвристике строится распределение времени исполнения, после чего численно проверяется гипотеза о сходстве-различии результатов для эвристик; в результате выбирается отделимая эвристика, с минимальным средним временем исполнения и ограничением на разброс в сторону увеличения времени исполнения. В том случае, если сценарии исполнения статистически неразделимы, к реализации предлагается схема с минимальным средним временем исполнения.

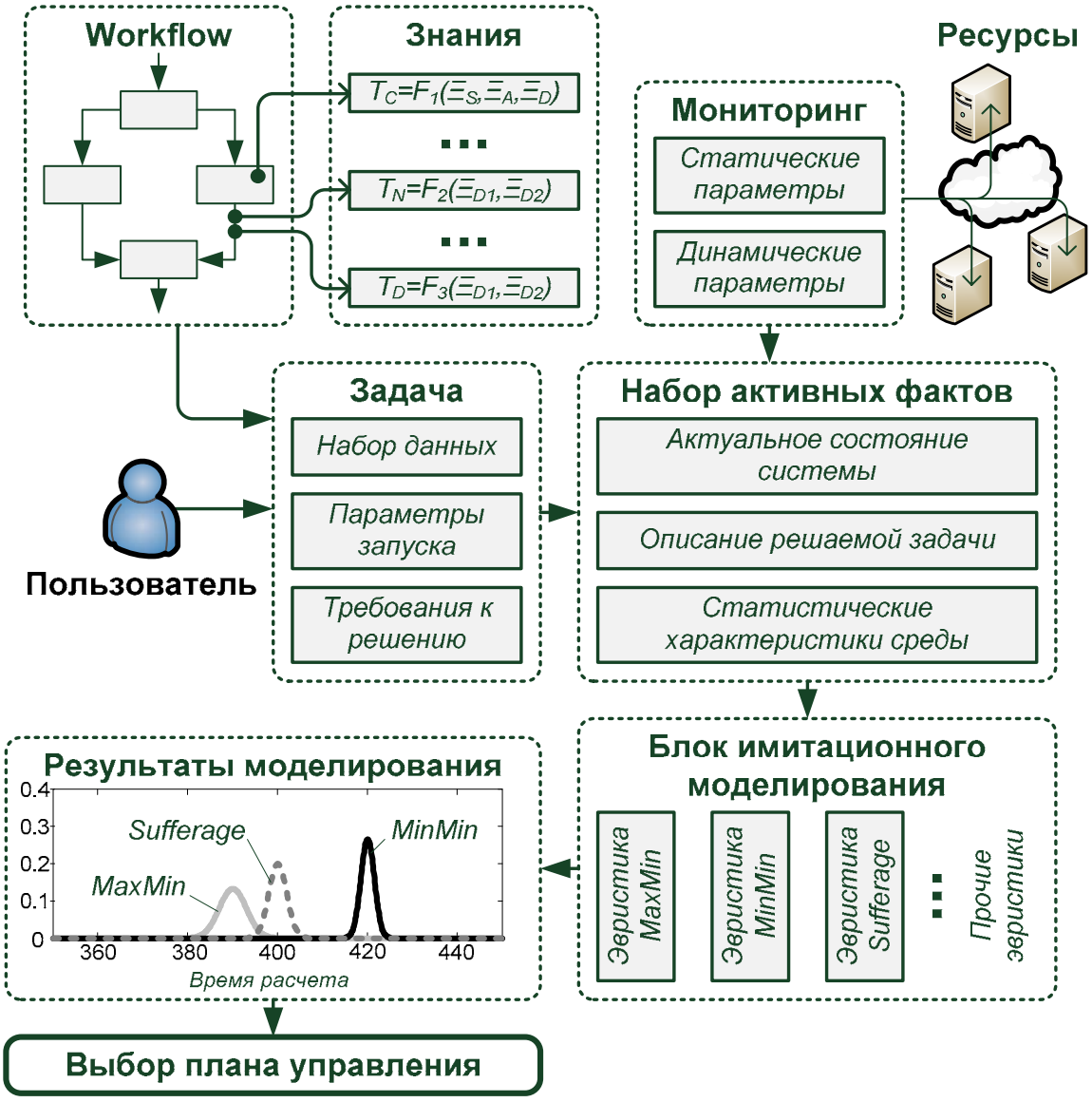

Рис. 2

Таким образом, предложенная интеллектуальная технология позволяет совокупно учесть стохастическую изменчивость характеристик распределенной среды и априорные знания о производительности прикладных сервисов в ходе планирования исполнения композитного приложения.

Архитектура и реализация сервисно-ориентированной платформы. Общая архитектура интеллектуальной платформы управления композитными приложениями приведена на рис. 3. Основная работа с компонентами (системными сервисами) в составе платформы осуществляется через интерфейс управляющего ядра, предназначенного для осуществления операций с пользовательскими WF и консолидации работы прочих системных сервисов. В интерфейс управляющего ядра входят базовые команды работы с WF: компоновка и доопределение описания композитного приложения в форме абстрактного

WF; запуск и остановка выполнения WF; получение информации о текущем состоянии WF, включая идентификаторы файлов входных и выходных данных в соответствующем хранилище.

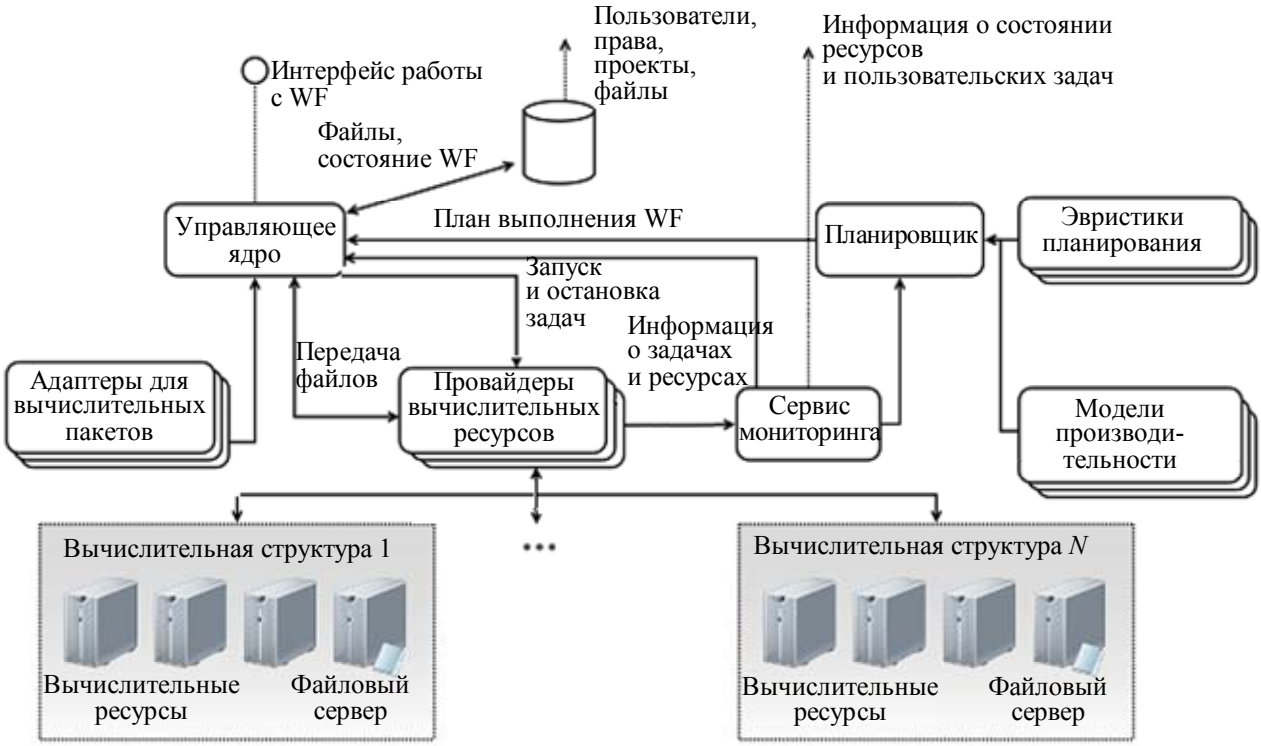

Рис. 3

Основным содержательным элементом платформы является сервис планирования (планировщик), предназначенный для составления расписания запусков, т.е. для отображения списка текущих задач, поступивших от управляющего ядра, на вычислительные ресурсы, информация о состоянии которых поступает от сервиса мониторинга. При планировании используются знания о вычислительных сервисах в форме параметрических моделей производительности и результаты имитационного моделирования в соответствии с процедурой, описанной в предыдущем разделе. Результатом планирования является расписание исполнения отдельных сервисов в составе WF.

Согласно полученному расписанию управляющим ядром осуществляются запуск заданий на соответствующих вычислительных ресурсах, контроль их исполнения, а также пред- и постобработка данных, выполняемая адаптерами вычислительных пакетов. К типичным задачам предобработки относится формирование входного файла для конкретного вычислительного пакета в соответствии с пользовательским описанием в терминах предметной области. К постобработке относится, например, конвертирование данных в необходимый формат для использования другими сервисами в составе WF или более удобного представления пользователю.

Для учета специфики запуска и сбора информации о ходе выполнения в конкретных вычислительных средах используется расширяемый набор провайдеров вычислительных ресурсов. Каждый провайдер — это подпрограмма, поставляемая вместе с платформой или написанная системным программистом, реализующая базовую функциональность взаимодействия с определенным набором вычислительных ресурсов. Такими ресурсами могут быть отдельные кластеры, группа кластеров или ресурсы Грид. После запуска задачи на конкретном вычислительном ресурсе информация о ее состоянии периодически запрашивается сервисом мониторинга. Кроме информации о состоянии задач в сервис мониторинга также поступают данные о

конфигурации и текущей загруженности вычислительных ресурсов. Эти данные передаются другим сервисам: планировщику для использования в процессе выбора ресурсов и построения расписания, управляющему ядру для выполнения необходимых действий при смене статуса задачи (например, при ее окончании или сбое), а также компонентам, взаимодействующим с пользователем для отображения хода выполнения задачи и загруженности ресурсов.

В табл. 2 приведены результаты экспериментального исследования производительности разработанной платформы в составе высокопроизводительного программного комплекса HPC-NASIS [5] для квантово-механических расчетов и моделирования наноразмерных атомно-молекулярных структур. Расчеты выполнялись в режиме метакомпьютинга (выделенные кластеры под управлением HPC-NASIS) и в среде Грид Национальной нанотехнологической сети [6]. Представлены статистические характеристики (среднее время M_x, СКО S_x, коэффициент вариации $V_x = S_x / M_x$) по отдельным составляющим накладных расходов платформы управления и среды распределенных вычислений в целом.

Таблица 2

Статистические характеристики составляющих времени накладных расходов (секунды) при исполнении композитного приложения HPC-NASIS

Временные характеристики запуска приложения	Режим метакомпьютинга			Режим Грид		
	M_x	S_x	V_x	M_x	S_x	V_x
Время выбора вычислительного ресурса	14,92	2,10	0,14	—	—	—
Время работы адаптеров платформы управления	0,90	0,06	0,07	0,32	0,08	0,26
Время передачи расчетных данных в хранилище платформы управления	3,62	0,13	0,03	3,13	0,66	0,21
Собственные накладные расходы распределенной среды	10,09	1,90	0,19	186,36	123,97	0,67
Накладные расходы на управление исполнением сервисов в платформе управления	9,16	2,68	0,29	9,70	0,92	0,09
Накладные расходы на управление исполнением WF в платформе управления	6,97	0,31	0,04	0,58	0,20	0,35

Проведенные экспериментальные исследования демонстрируют, что накладные расходы платформы управления сопоставимы с накладными расходами инфраструктуры распределенных вычислений в режиме метакомпьютинга и на порядок меньше накладных расходов в среде Грид, что подтверждает возможность практического использования разработанной платформы управления композитными приложениями в распределенных вычислительных средах, без оказания при этом существенного влияния на общую производительность вычислительной инфраструктуры.

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации, при частичной поддержке государственного контракта № 16.647.12.2025 „Создание функционирующего в режиме удаленного доступа интерактивного учебно-методического комплекса для выполнения работ в области моделирования наноразмерных атомно-молекулярных структур, наноматериалов, процессов и устройств на их основе, в распределенной вычислительной среде“ и ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

СПИСОК ЛИТЕРАТУРЫ

1. *Марьин С. В.* Интеллектуальная платформа управления композитными приложениями в распределенных вычислительных средах. Дис… канд. техн. наук. СПб: СПбГУ ИТМО, 2010.

2. *Deelman E.* et al. Pegasus: Mapping scientific workflows onto the grid // Europ. Across Grids Conf. 2004. P. 11—20.

3. *Бухановский А. В., Ковальчук С. В., Марьин С. В.* Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.

4. *Yu J.* et al. Workflow Scheduling Algorithms for Grid Computing, Metaheuristics for Scheduling in Distributed Computing Environments / Ed. by *F. Xhafa and A. Abraham.* Berlin: Springer, 2008.

5. Свидетельство о регистрации ПС ЭВМ №2009615970. Ядро высокопроизводительного программного комплекса для квантово-механических расчетов и моделирования атомно-молекулярных наноразмерных структур и комплексов / *А. В. Бухановский* и др. 26.10.2009.

6. Грид Национальной нанотехнологической сети (ГридННС) [Электронный ресурс]: <http://www.ngrid.ru/trac/>.

Сведения об авторах

Сергей Владимирович Марьин — канд. техн. наук; НИИ Наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: sm.niinkt@gmail.com

Сергей Валерьевич Ковальчук — канд. техн. наук; НИИ Наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник; E-mail: kovalchuk@mail.ifmo.ru

Рекомендована НИИ НКТ

Поступила в редакцию
15.05.11 г.