

УДК 004.942

## МОДЕЛИ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ВЫПОЛНЕНИЯ СХЕМ РЕШЕНИЯ ЗАДАЧ В ГЕТЕРОГЕННОЙ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

<sup>1</sup>Феоктистов А.Г., <sup>2</sup>Башарина О.Ю.<sup>1</sup>ФГБУН «Институт динамики систем и теории управления им. В.М. Матросова» СО РАН,  
Иркутск, e-mail: agf65@yandex.ru;<sup>2</sup>ФГБОУ ВО «Иркутский государственный университет», Иркутск, e-mail: basharinaolga@mail.ru

В настоящее время распределенные пакеты прикладных программ играют существенную роль в рамках поддержки проведения крупномасштабных научных экспериментов в гетерогенной вычислительной среде. Гетерогенность подразумевает использование в рамках среды узлов (персональных компьютеров, серверов, кластеров, облачных платформ) с разными архитектурами и вычислительными характеристиками. Схемы решения задач в таких пакетах строятся на вычислительных моделях, представляющих собой частный случай семантической сети. Для построения схем решения задач используются методы планирования вычислений и информационного планирования. В статье предложены новые модели прогнозирования времени выполнения схем решения задач в гетерогенной среде. Задача прогнозирования времени выполнения программ в гетерогенной среде является чрезвычайно актуальной. Ее решение позволяет эффективно распределять и использовать ресурсы среды. Предложенные модели позволяют, в отличие от известных, учитывать различные оценки времени выполнения модулей, входящих в состав схем решения задач. В статье приведены результаты сравнительного анализа точности прогнозирования времени выполнения схем с использованием различных оценок времени выполнения модулей. Наибольшая точность достигается при использовании оценок времени выполнения модулей, полученных на основе динамического анализа программ.

**Ключевые слова:** распределенные вычисления, гетерогенные ресурсы, вычислительная модель, время выполнения программ, прогнозирование

## MODELS FOR PREDICTING THE EXECUTION TIME OF PROBLEM-SOLVING SCHEMES IN A HETEROGENEOUS DISTRIBUTED COMPUTING ENVIRONMENT

<sup>1</sup>Feoktistov A.G., <sup>2</sup>Basharina O.Yu.<sup>1</sup>Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, e-mail: agf65@yandex.ru;<sup>2</sup>Irkutsk State University, Irkutsk, e-mail: basharinaolga@mail.ru

Nowadays, distributed applied software packages play a significant role in supporting large-scale scientific experiments in a heterogeneous computing environment. Within the framework of the environment, heterogeneity implies the use of nodes (personal computers, servers, clusters, cloud platforms) with different architectures and computational characteristics. In such packages, problem-solving schemes are created on computational models, which are a special case of the semantic network. To construct the problem-solving schemes, methods of computations planning and information planning are used. We propose new models for predicting the execution time of problem-solving schemes in a heterogeneous environment. The problem of predicting the program execution time in a heterogeneous environment is extremely relevant. Its solution allows us to effectively allocate and use environment resources. In contrast to the known models, the proposed ones allow us to take into account various evaluations for the runtime of scheme modules. In the paper, the results of a comparative analysis of the accuracy in predicting the execution time of problem-solving schemes using various evaluations of the module runtime are shown. The greatest accuracy is achieved by using estimates of the module runtime obtained on the basis of dynamic analysis of programs.

**Keywords:** distributed computing, heterogeneous resources, computational model, program runtime, predicting

В настоящее время научные приложения, ориентированные на проведение крупномасштабных научных экспериментов в гетерогенной распределенной вычислительной среде (ГРВС), играют существенную роль в процессе решения важных практических задач на основе математического моделирования исследуемых систем [1]. Как правило, такие приложения реализуются в виде распределенных пакетов прикладных программ (РППП). Гетерогенность среды подразумевает использование в ней узлов (персональных компьютеров, серверов, кластеров, облачных платформ) с разными программно-аппаратными архитектурами и вычислительными характеристиками.

Схемы решения задач в РППП представляют собой спецификацию процесса вычислений в ГРВС. Они строятся на вычислительных моделях ГРВС, представляющих собой частный случай семантической сети. Для построения схем решения задач используются методы планирования вычислений и информационного планирования.

Для эффективного решения задач в ГРВС и рационального использования ее ресурсов необходима возможность оценки времени выполнения схем решения этих задач. В статье предложены новые модели прогнозирования времени выполнения таких схем в ГРВС. Эти модели позволяют, в отличие от известных моделей аналогич-

ного назначения [2–4], учитывать различные оценки времени выполнения модулей, входящих в состав схем. В их числе оценки, получаемые на основе методики динамического анализа времени выполнения программ, предложенной авторами статьи [5] и используемой в процессе непрерывной интеграции прикладного программного обеспечения разрабатываемых пакетов [6].

*Модели прогнозирования времени выполнения схем решения задач*

Вычислительная модель ГРВС определяется структурой

$$\mathbb{M} = \langle ASP, Z, F, M, PR, S, J, R, A, Q, PLCS, MH, O \rangle,$$

$$R, A, Q, PLCS, MH, O \rangle,$$

где  $ASP, Z, F, M, S, J, R, A, Q, O$  – это соответственно множества пакетов прикладных

Параметры подмножеств  $Z_i^{in}$  и  $Z_i^{out}$  отражают назначение и семантику формальных параметров модуля  $m_j$ , реализующего операцию  $f_i$ . Параметры передаются между модулями в виде файлов.

Процессы решения задач в пакетах представлены схемами из  $S$ . Схема  $s \in S$  выполнения операций из  $F$  является аналогом строгой параллельной формы графа алгоритма, а ее построение сходно с построением данной формы [8]. В рамках рассматриваемой вычислительной модели схема является связным подграфом ориентированного ациклического двудольного графа, представляющего схемные знания об алгоритмах исследования предметной области. На рис. 1 представлен пример такого графа. Операции и параметры изображены на нем закрашенными и незакрашенными кружками.

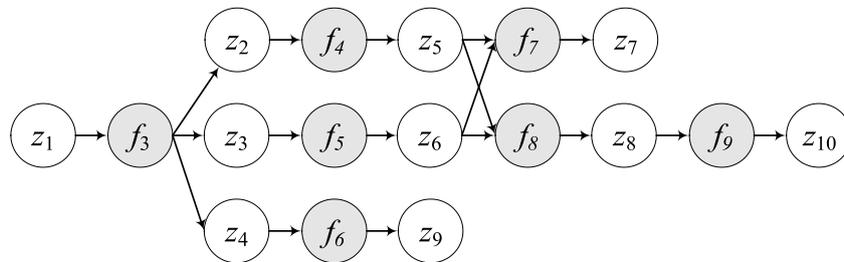


Рис. 1. Двудольный ориентированный граф схемы решения задачи

программ, параметров, операций, программных модулей, продуктов, схем решения задач, заданий, ресурсов, агентов, ограничений на выполнение заданий и использование ресурсов, накладываемых множеством  $PLCS$  административных политик, а также отношений между перечисленными объектами. Структура данных  $MH$  отражает вычислительную историю работы модулей из  $M$ . Детальное описание компонентов модели можно найти в [5, 7].

Операции из  $F$  определяют отношения вычислимости на множестве параметров  $Z$ . Параметры могут быть представлены скалярами, векторами, матрицами различных базовых типов данных или произвольными структурами данных. Каждая операция  $f_i \in F$  реализуется модулем  $m_j \in M$ , где  $i \in 1, n_f$ ,  $j \in 1, n_m$ ,  $n_f$  – число операций,  $n_m$  – число модулей. Один модуль может реализовать несколько операций. С каждой операцией  $f_i$  связано два подмножества параметров  $Z_i^{in}, Z_i^{out} \subset Z$ .  $Z_i^{in}$  включает параметры, значения которых необходимо задать, чтобы получить значения параметров из  $Z_i^{out}$ .

С целью поддержки общности планирования вычислений при построении схем решения задач с использованием их постановок в вычислительной модели выделяются две специальные операции  $f_1$  и  $f_2$  [9], моделирующие в дальнейшем условия задачи, подмножество параметров, значения которых заданы, и подмножество параметров, значения которых нужно вычислить. Операция  $f_1$  определяет подмножества  $Z_1^{in} = \emptyset$  и  $Z_1^{out} \neq \emptyset$ , а операция  $f_2$  обуславливает подмножества  $Z_2^{in} \neq \emptyset$  и  $Z_2^{out} = \emptyset$ . В приведенном выше примере  $Z_1^{in} = \emptyset$  и  $Z_1^{out} = \{z_1\}$ , а  $Z_2^{in} = \{z_7, z_9, z_{10}\}$  и  $Z_2^{out} = \emptyset$ .

Прогнозирование времени выполнения схем решения задач здесь и ниже базируется на подходе к оценке производительности вычислительных устройств, предложенном в [7]. Пусть матрица  $\mathbf{P}$  размерности  $k \times k$  представляет сведения о предшествовании  $k$  операций схемы решения задачи. Элемент  $p_{ij} = 1$  ( $p_{ij} = 0$ ) матрицы  $\mathbf{P}$  означает, что выполнение операции  $f_i$  в схеме решения задачи предшествует (не предшествует) выполнению операции  $f_j$  и  $Z_i^{in} \cap Z_j^{out} \neq \emptyset$ . Матрица  $\mathbf{D}$  размерности  $k \times k$  отражает

оценки объемов передаваемых данных между операциями. Элемент матрицы  $\hat{d}_{ij} \geq 0$  показывает объем данных, передаваемых операцией  $f_i$  операции  $f_j$ . Матрица  $\mathbf{W}$  размерности  $k \times k$  предоставляет информацию о пропускной способности интерконнекта между узлами, в которых запускаются модули, реализующие операции схемы. Элемент матрицы  $w_{ij} \geq 0$  демонстрирует пропускную способность интерконнекта между узлами, в которых работают модули, реализующие операции  $f_i$  и  $f_j$ . Тогда оценка  $\hat{T}_s$  времени выполнения схемы в асинхронном режиме по готовности данных определяется следующим образом:

$$\hat{T}_s = \max_{i=1,k} \hat{\tau}_i,$$

$$\hat{\tau}_i = \hat{q}_i + \hat{t}_i + \max_{\forall j \in \{1,k: p_{ij}=1, i \neq j\}} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right),$$

где  $\hat{\tau}_i$  ( $\hat{\tau}_j$ ) – оценка времени, прошедшего с начала обработки схемы до завершения операции  $f_i$  ( $f_j$ ),  $\hat{q}_i$  – оценка времени нахождения в очереди модуля, реализующего операцию  $f_i$  (в то время, когда все данные, необходимые для его выполнения, готовы),  $\hat{t}_i$  – прогнозная оценка времени выполне-

ния этого модуля,  $k$  – число операций схемы,  $0 < \omega_{ji}(t) \leq 1$  – коэффициент снижения пропускной способности интерконнекта в момент времени  $t$  между узлами, в которых функционируют модули, реализующие операции  $f_i$  и  $f_j$ .

Пусть матрица  $\mathbf{S}$  размерности  $m \times k$  представляет собой ярусно-параллельную форму, описывающую обработку схемы в режиме fork/join,  $m$  – число ярусов схемы. Элемент  $s_{li} = 1$  означает, что операция  $f_i$  должна быть выполнена на  $l$ -м ярусе. Переход к операциям  $(l+1)$ -го яруса возможен при условии завершения всех операций на  $l$ -м ярусе. Тогда оценка  $\hat{T}'_s$  времени выполнения схемы в режиме fork/join находится следующим образом:

$$\hat{T}'_s = \sum_{l=1}^m \hat{\tau}_l,$$

$$\hat{\tau}_l = \max_{\forall i \in \{1,k: s_{li}=1\}} \left( \hat{q}_i + \hat{t}_i + \sum_{\forall j \in \{1,k: p_{ij}=1, i \neq j\}} \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}_{vs}$  времени выполнения схемы в среде с виртуализированными ресурсами в асинхронном режиме определяется следующим образом:

$$\hat{T}'_{vs} = \max_{i=1,k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i^{vm\_launch} + \hat{t}_i + \hat{t}_i^{vm\_remove} + \max_{\forall j \in \{1,k: p_{ij}=1, i \neq j\}} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right),$$

где  $\hat{t}_i^{vm\_launch}$  и  $\hat{t}_i^{vm\_remove}$  – это соответственно оценки времени на запуск и завершение ВМ для выполнения модуля, реализующего операцию  $f_i$ . Оценки  $\hat{t}_i^{vm\_launch}$  и  $\hat{t}_i^{vm\_remove}$  определяются экспериментальным путем для ВМ различной конфигурации.

Оценка  $\hat{T}_{vs}$  времени выполнения схемы в среде с виртуализированными ресурсами соответственно в режиме fork/join определяется следующим образом:

$$\hat{T}'_{vs} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \{1,k: s_{li}=1\}} \left( \hat{q}_i + \hat{t}_i^{vm\_launch} + \hat{t}_i + \hat{t}_i^{vm\_remove} + \sum_{\forall j \in \{1,k: p_{ij}=1, i \neq j\}} \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}_{rs}$  времени выполнения схемы в асинхронном режиме с рестартом модулей определяется следующим образом:

$$\hat{T}'_{rs} = \max_{i=1,k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i^{fault} + \hat{t}_i + \tilde{t}_i + \hat{t}_i^{rest} + \max_{\forall j \in \{1,k: p_{ij}=1, i \neq j\}} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right),$$

где  $\hat{t}_i^{fault}$  и  $\hat{t}_i^{rest}$  – это соответственно оценки времени обнаружения и идентификации отказа, а также повторного запуска (рестарта) модуля, реализующего операцию  $f_i$ , в случае программно-аппаратного отказа. Оценки  $\hat{t}_i^{fault}$  и  $\hat{t}_i^{rest}$  определяются средним временем выполнения таких процессов системой метамониторинга для разных видов программно-аппаратных отказов,  $\tilde{t}_i$  – время выполнения модуля до отказа.

Оценка  $\hat{T}'_{rs}$  времени выполнения схемы в среде с виртуализированными ресурсами соответственно в режиме fork/join определяется следующим образом:

$$\hat{T}'_{rs} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{li}=1} \left( \hat{q}_i + \hat{t}_i^{fault} + \hat{t}_i + \tilde{t}_i + \hat{t}_i^{rest} + \sum_{\forall j \in \overline{1, k}: p_{lj}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}'_{us}$  времени выполнения схемы в асинхронном режиме по готовности данных на основе пользовательских оценок времени выполнения модулей определяется следующим образом:

$$\hat{T}'_{us} = \max_{i=1, k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i + \max_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right),$$

где  $\hat{t}_i$  – пользовательская оценка времени выполнения модуля.

Оценка  $\hat{T}'_{us}$  времени выполнения схемы в режиме fork/join определяется следующим образом:

$$\hat{T}'_{us} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{li}=1} \left( \hat{q}_i + \hat{t}_i + \sum_{\forall j \in \overline{1, k}: p_{lj}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}'_{uvs}$  времени выполнения схемы в среде с виртуализированными ресурсами в асинхронном режиме определяется следующим образом:

$$\hat{T}'_{uvs} = \max_{i=1, k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i^{vm\_launch} + \hat{t}_i + \hat{t}_i^{vm\_remove} + \max_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}'_{uvs}$  времени выполнения схемы в среде с виртуализированными ресурсами соответственно в режиме fork/join определяется следующим образом:

$$\hat{T}'_{uvs} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{li}=1} \left( \hat{q}_i + \hat{t}_i^{vm\_launch} + \hat{t}_i + \hat{t}_i^{vm\_remove} + \sum_{\forall j \in \overline{1, k}: p_{lj}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}'_{urs}$  времени выполнения схемы в асинхронном режиме с рестартом модулей определяется следующим образом:

$$\hat{T}'_{urs} = \max_{i=1, k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i^{fault} + \hat{t}_i + \hat{t}_i^{rest} + \max_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}'_{urs}$  времени выполнения схемы в среде с виртуализированными ресурсами соответственно в режиме fork/join определяется следующим образом:

$$\hat{T}'_{urs} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{li}=1} \left( \hat{q}_i + \hat{t}_i^{fault} + \hat{t}_i + \hat{t}_i^{rest} + \sum_{\forall j \in \overline{1, k}: p_{lj}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right).$$

Оценка  $\hat{T}'_{uss}$  времени выполнения схемы в асинхронном режиме по готовности данных на основе пользовательских оценок времени выполнения модулей определяется следующим образом:

$$\hat{T}'_{uss} = \max_{i=1, k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \beta_i \hat{t}_i'' + \max_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t)w_{ji}} \right),$$

где  $\hat{t}_i''$  – скорректированная пользовательская оценка времени выполнения модуля,  $\beta_i = \psi(i, MH, T_h) > 0$  – это корректирующий коэффициент, вычисляемый на основе вычислительной истории. Функция  $\psi(i, MH, T_h)$  вычисляет значение коэффициента  $\beta_i$ , используя среднее или медианное значения по выборке времени выполнения модуля, реализующего операцию  $f_p$  за определенный период  $T_h$ .

Оценка  $\hat{T}'_{uss}$  времени выполнения схемы в режиме fork/join определяется следующим образом:

$$\hat{T}'_{uss} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{ij}=1} \left( \hat{q}_i + \beta_i \hat{t}_i'' + \sum_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t) w_{ji}} \right).$$

Оценка  $\hat{T}'_{usvs}$  времени выполнения схемы в среде с виртуализированными ресурсами в асинхронном режиме определяется следующим образом:

$$\hat{T}'_{usvs} = \max_{i=1, k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i^{vm\_launch} + \beta_i \hat{t}_i'' + \hat{t}_i^{vm\_remove} + \max_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t) w_{ji}} \right).$$

Оценка  $\hat{T}'_{usrs}$  времени выполнения схемы в среде с виртуализированными ресурсами соответственно в режиме fork/join определяется следующим образом:

$$\hat{T}'_{usrs} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{ij}=1} \left( \hat{q}_i + \hat{t}_i^{vm\_launch} + \beta_i \hat{t}_i'' + \hat{t}_i^{vm\_remove} + \sum_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t) w_{ji}} \right).$$

Оценка  $\hat{T}'_{usrs}$  времени выполнения схемы в асинхронном режиме с рестартом модулей определяется следующим образом:

$$\hat{T}'_{usrs} = \max_{i=1, k} \hat{\tau}_i, \quad \hat{\tau}_i = \hat{q}_i + \hat{t}_i^{fault} + \beta_i \hat{t}_i'' + \tilde{t}_i + \hat{t}_i^{rest} + \max_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \left( \hat{\tau}_j + \frac{\hat{d}_{ji}}{\omega_{ji}(t) w_{ji}} \right).$$

Оценка  $\hat{T}'_{usrs}$  времени выполнения схемы в среде с виртуализированными ресурсами соответственно в режиме fork/join определяется следующим образом:

$$\hat{T}'_{usrs} = \sum_{l=1}^m \hat{\tau}_l, \quad \hat{\tau}_l = \max_{\forall i \in \overline{1, k}: s_{ij}=1} \left( \hat{q}_i + \hat{t}_i^{fault} + \beta_i \hat{t}_i'' + \tilde{t}_i + \hat{t}_i^{rest} + \sum_{\forall j \in \overline{1, k}: p_{ij}=1, i \neq j} \frac{\hat{d}_{ji}}{\omega_{ji}(t) w_{ji}} \right).$$

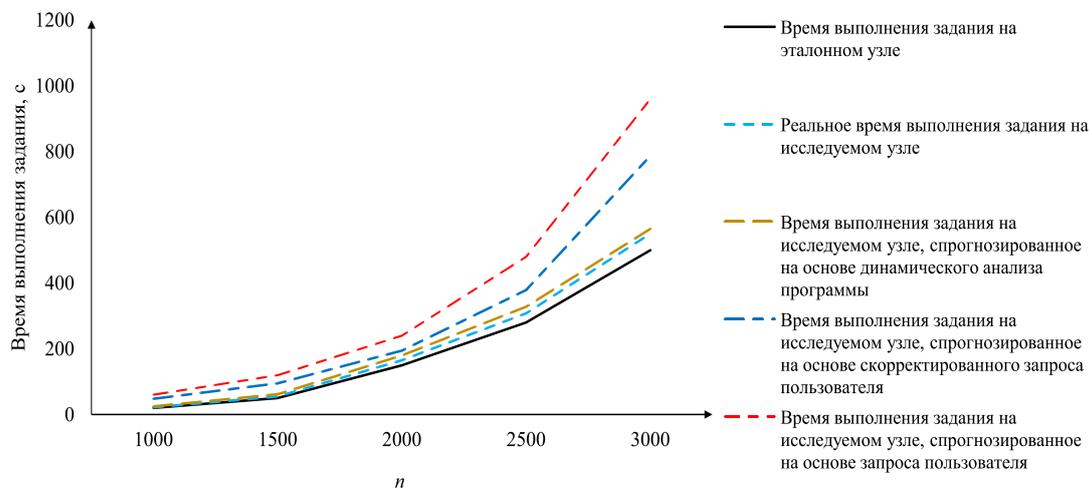


Рис. 2. Результаты реального и прогнозируемого времени выполнения заданий по перемножению матриц

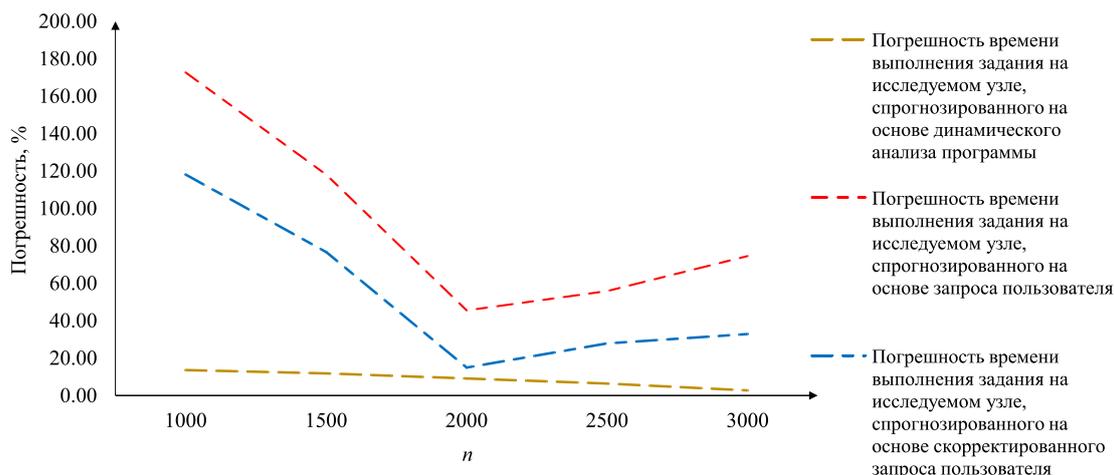


Рис. 3. Погрешность прогнозируемого времени выполнения заданий

В качестве примера проведем сравнение погрешности различных способов прогнозирования времени выполнения заданий для задач линейной алгебры (перемножение матриц размерности  $n \times n$ ). Задачи решались в ГРВС, организованной на базе ресурсов Центра коллективного пользования «Иркутский суперкомпьютерный центр СО РАН» [10], в процессе непрерывной интеграции (модификации, контроля версий, сборки, тестирования, доставки и размещения) модулей, реализующих операции линейной алгебры. На рис. 2 приведены результаты времени выполнения заданий по перемножению матриц для  $n = 1000, 1500, 2000, 2500, 3000$  одной и той же программой на эталонном и исследуемом узлах, а также время выполнения этих заданий, определенное с помощью трех способов его прогнозирования, рассмотренных выше.

Очевидно, что наиболее точный прогноз формируется на основе динамического анализа программ. На рис. 3 приведены погрешности различных способов прогнозирования времени выполнения заданий в сравнении с реальным временем их выполнения в исследуемом узле. Результаты, приведенные на рис. 3, показывают, что погрешность времени выполнения задания, спрогнозированного на основе динамического анализа программы, убывает с увеличением размерности матриц. В данном примере она не превышает 13,64%. В то же время изменение погрешности времени выполнения задания, спрогнозированного на основе запроса пользователя, может иметь немонотонный характер. На практике время выполнения задания,

спрогнозированного на основе запроса пользователя, как правило, завышается. Корректировка времени в запросе пользователя на основе вычислительной истории его заданий может уменьшить погрешность прогнозирования.

### Заключение

Рассмотренные в статье модели прогнозирования схем решения задач в ГРВС будут использованы в процессе децентрализованного управления вычислениями [6].

*Работа выполнена при поддержке РФФИ, проект № 19-07-00097-А и Президиума РАН, программа № 7, проект «Методы, алгоритмы и инструментальные средства децентрализованного группового решения задач в вычислительных и управляющих системах».*

### Список литературы

1. Deelman E., Peterka T., Altintas I., Carothers C.D., van Dam K.K., Moreland K., Parashar M., Ramakrishnan L., Tauber M., Vetter J. The future of scientific workflows. *The International Journal of High Performance Computing Applications*. 2018. Vol. 32. № 1. P. 159–175. DOI: 10.1177/1094342017704893.
2. Cao F., Zhu M.M., Ding D. Distributed workflow scheduling under throughput and budget constraints in grid environments. *Lecture Notes in Computer Science*. 2013. Vol. 8429. P. 62–80. DOI: 10.1007/978-3-662-43779-7\_4.
3. Qin J., Fahringer T. Semantic-based scientific workflow composition. *Scientific Workflows*. – Springer, Berlin, Heidelberg, 2012. P. 115–134. DOI: 10.1007/978-3-642-30715-7\_7.
4. Deelman E., Vahia K., Juvea G., Ryngea M., Callaghan S., Maechling P.J., Mayani R., Chen W., da Silva R.F., Livny M., Wenger K. Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*. 2015. Vol. 46. P. 17–35. DOI: 10.1016/j.future.2014.10.008.
5. Феоктистов А.Г., Башарина О.Ю. Методика динамического анализа времени выполнения программ в гетерогенных распределенных вычислительных средах // *Вестник Иркутского*

го государственного технического университета. 2018. Т. 22. № 6. С. 109–119. DOI: 10.21285/1814-3520-2018-6-109-119.

6. Феоктистов А.Г., Горский С.А., Сидоров И.А., Костромин Р.О., Фереферов Е.С., Бычков И.В. Непрерывная интеграция функционального наполнения распределенных пакетов прикладных программ в Orlando Tools // Труды ИСП РАН. 2019. Т. 31. № 2. С. 83–96. DOI: 10.15514/ISPRAS-2019-31(2)-7.

7. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Корсуков А.С. Децентрализованное управление потоками заданий в интегрированной кластерной системе // Вестник Новоси-

бирского гос. ун-та. Серия: Информационные технологии. 2011. Т. 9. № 2. С. 42–54.

8. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ–Петербург, 2002. 608 с.

9. Опарин Г.А., Новопашии А.П. Булево моделирование планирования действий в распределенных вычислительных системах // Известия РАН. Теория и системы управления. 2004. № 5. С. 105–108.

10. ЦКП Иркутский суперкомпьютерный центр СО РАН. [Электронный ресурс]. URL: <http://hpc.icc.ru/> (дата обращения: 23.10.2019).