

На правах рукописи



ШАМАКИНА Анастасия Валерьевна

**МЕТОДЫ УПРАВЛЕНИЯ РЕСУРСАМИ
В ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ
РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ**

05.13.11 – математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Челябинск – 2014

Работа выполнена на кафедре системного программирования
ФГБОУ ВПО «Южно-Уральский государственный университет»
(национальный исследовательский университет)

Научный руководитель: СОКОЛИНСКИЙ Леонид Борисович
доктор физ.-мат. наук, профессор,
проректор по информатизации, ФГБОУ ВПО «Южно-
Уральский государственный университет»
(национальный исследовательский университет)

Официальные оппоненты: ВОЕВОДИН Владимир Валентинович
доктор физ.-мат. наук, член-корреспондент РАН,
заместитель директора НИВЦ, ФГБОУ ВПО
«Московский государственный университет имени
М.В. Ломоносова»

СОЗЫКИН Андрей Владимирович
кандидат техн. наук,
заведующий сектором суперкомпьютерных технологий,
ФГБУН Институт математики и механики имени
Н.Н. Красовского УрО РАН

Ведущая организация: ФГАОУ ВПО Санкт-Петербургский национальный
исследовательский университет информационных техно-
логий, механики и оптики

Защита состоится 24 декабря 2014 г. в 14:00 часов на заседании диссертационного совета
Д 212.298.18 при ФГБОУ ВПО «Южно-Уральский государственный университет» (нацио-
нальный исследовательский университет) по адресу: 454080, г. Челябинск, пр. Ленина, 76,
ауд. 1001.

С диссертацией можно ознакомиться в библиотеке Южно-Уральского государственного
университета и на сайте:
<http://susu.ac.ru/ru/dissertation/d-21229818/shamakina-anastasiya-valerevna>.

Автореферат разослан “ _____ ” _____ 2014 г.

Ученый секретарь
диссертационного совета



М.Л. Цымблер

Общая характеристика работы

Актуальность темы. Развитие технологий распределенных вычислений в конце 1990-х годов позволило объединить географически-распределенные по всему миру гетерогенные ресурсы. Появились технические возможности для решения масштабных задач в области науки, техники и коммерции на территориально-распределенных ресурсах, принадлежащих разным владельцам. Исследования данной тематики привело к возникновению концепции грид вычислений, и затем к новой концепции облачных вычислений. Для раскрытия всех потенциальных возможностей использования распределенных вычислительных ресурсов принципиально важно наличие результативных и эффективных алгоритмов планирования, используемых менеджерами ресурсов.

В настоящее время перспективным является направление, связанное с применением распределенных вычислительных технологий для решения ресурсоемких научных задач в разных предметных областях: медицине, инженерном проектировании, нанотехнологиях, прогнозировании климата и др. Вычислительное задание в подобных предметных областях во многих случаях имеет *потокковую структуру* и может быть описано с помощью модели потока работ, в соответствии с которой задание представляется в виде ориентированного ациклического графа, узлами которого являются взаимосвязанные вычислительные задачи, а дуги соответствуют потокам данных, передаваемых между отдельными задачами. При этом набор задач, из которых строятся задания, является конечным и предопределенным. Проблемно-ориентированная специфика потоков работ в подобных сложных приложениях выражается в том, что в подавляющем большинстве случаев, еще до выполнения задания, для каждой задачи могут быть получены оценки таких качественных характеристик, как время выполнения задачи на одном процессорном ядре, пределы масштабируемости и объем генерируемых данных. Использование подобных знаний о специфике задач в конкретной проблемно-ориентированной области может существенно улучшить эффективность методов управления вычислительными ресурсами.

В соответствие с этим актуальной является задача разработки методов и алгоритмов управления ресурсами в проблемно-ориентированных распределенных вычислительных средах, учитывающих специфику предметной области, масштабируемость отдельных задач в задании и использующих возможность параллельного выполнения независимых задач.

Цель и задачи исследования. *Цель* данной работы состояла в разработке методов и алгоритмов распределения ресурсов и планирования заданий, учитывающих специфику проблемно-ориентированных распределенных вычислительных сред, а также в разработке на их основе брокера ресурсов, который может быть использован в распределенных вычислительных системах. Для достижения этой цели необходимо было решить следующие задачи:

1. Разработать формальные методы представления проблемно-ориентированных распределенных вычислительных сред.
2. Разработать проблемно-ориентированный алгоритм планирования ресурсов для заданий, представляемых в виде потока работ.
3. Разработать архитектуру и принципы структурной организации брокера ресурсов для проблемно-ориентированных распределенных вычислительных сред.
4. Реализовать брокер ресурсов и провести вычислительные эксперименты для исследования эффективности предложенных подходов.

Методы исследования. В исследованиях, проводимых в диссертационной работе, использован математический аппарат, в основе которого лежит теория множеств и теория графов. Для представления заданий использована модель потока работ. При разработке брокера ресурсов применялись методы объектно-ориентированного проектирования и язык UML.

Научная новизна работы заключается в следующем:

1. Разработана оригинальная математическая модель проблемно-ориентированной распределенной вычислительной среды.
2. Предложен новый алгоритм планирования ресурсов, учитывающий специфику проблемно-ориентированной распределенной вычислительной среды.
3. Разработан брокер ресурсов, ориентированный на работу в проблемно-ориентированных распределенных вычислительных средах.

Теоретическая ценность работы состоит в том, что в ней дано формальное описание методов и алгоритмов управления ресурсами в проблемно-ориентированных распределенных вычислительных средах, включающее в себя математические модели распределенной вычислительной системы и заданий с потоковой структурой.

Практическая ценность работы заключается в том, что на базе предложенных методов и алгоритмов разработан брокер ресурсов, позволяющий организовать эффективное использование ресурсов в проблемно-ориентированных распределенных вычислительных средах.

Апробация работы. Основные положения диссертационной работы, разработанные модели, методы, алгоритмы и результаты вычислительных экспериментов докладывались автором на следующих международных и всероссийских научных конференциях:

- на Международной научной конференции ISC'2014 (22–26 июня 2014 г., Лейпциг, Германия);
- на Международной научной конференции ISC'2013 (16–20 июня 2013 г., Лейпциг, Германия);
- на Международной научной конференции «Параллельные вычислительные технологии 2014» (1–3 апреля 2014 г., Ростов-на-Дону);
- на Международной научной конференции «Параллельные вычислительные технологии 2010» (29 марта – 2 апреля 2010 г., Уфа);
- на Всероссийской научной конференции «Научный сервис в сети Интернет 2009: масштабируемость, параллельность, эффективность» (21–26 сентября 2009 г., Новороссийск);
- на Всероссийской научной конференции «Научный сервис в сети Интернет 2008: решение больших задач» (22–27 сентября 2008 г., Новороссийск).

Публикации. По теме диссертации опубликовано 15 печатных работ. Работы [1–5] опубликованы в журналах, включенных ВАК в перечень журналов, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук. Работа [6] проиндексирована в библиографической базе данных Scopus. В статье [1] А.В. Шамакиной принадлежит раздел 4, (стр. 100–102). В статье [3] А.В. Шамакиной принадлежит раздел «Расширение состава прикладных сервисов НРС-NASIS», (стр. 83). В статье [4] А.В. Шамакиной принадлежит раздел 3, (стр. 72–74). В работе [10] А.В. Шамакиной принадлежит раздел 3, (стр. 73). В работе [12] А.В. Шамакиной принадлежит раздел 2, (стр. 439). В работе [9] Л.Б. Соколинскому принадлежит постановка задачи, А.В. Шамакиной принадлежат все полученные результаты.

В рамках выполнения диссертационной работы получено 8 свидетельств Роспатента об официальной регистрации программ для ЭВМ и баз данных.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения и библиографии. Объем диссертации составляет 106 страниц, объем библиографии — 107 наименований.

Содержание работы

Во введении приводится обоснование актуальности темы, формулируются цели работы, ее новизна и практическая значимость; приводится обзор работ по тематике исследования и кратко излагается содержание диссертации.

В первой главе, «Технологии распределенных вычислений», дается описание методов управления ресурсами в проблемно-ориентированных распределенных вычислительных средах. Рассматриваются современные подходы к планированию ресурсов. Дается общая классификация алгоритмов планирования. На рис. 1 представлена классификация алгоритмов планирования, основанная на наличии/отсутствии связей между задачами.

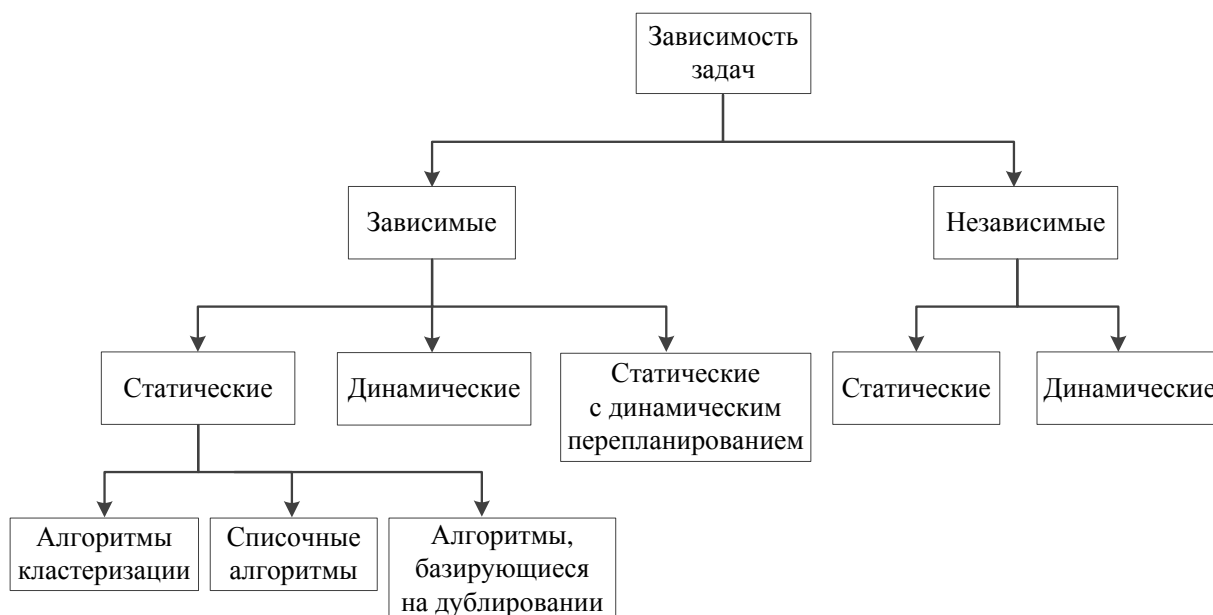


Рис. 1. Классификация алгоритмов планирования ресурсами.

Главная стратегия планирования независимых задач заключается в назначении независимых задач на ресурсы в зависимости от их загрузки с целью обеспечения высокой пропускной способности вычислительной системы. Примерами статических алгоритмов с оценкой производительности являются: алгоритм минимального времени выполнения (MET — Minimum Execution Time), алгоритм минимального времени завершения (MCT — Minimum Completion Time), эвристики Min-Min и Max-Min, Suffrage и XSuffrage. Данные алгоритмы обычно используются для планирования заданий, которые состоят из множества независимых задач с большими модулями и интенсивными вычислениями.

В случае планирования задач, имеющих зависимости, задание обычно представляется в виде ориентированного ациклического графа, в котором каждая вершина представляет собой задачу, ориентированное ребро обозначает порядок приоритета между двумя вершинами. В некоторых случаях к вершинам и ребрам могут быть добавлены веса, показывающие вычислительную стоимость и коммуникационную стоимость соответственно. Важнейшей проблемой при планировании заданий с потоковой структурой является

нахождение компромисса между использованием максимального параллелизма задач в задании и минимизации коммуникационных задержек. Для решения данной проблемы были предложены три вида эвристических алгоритмов: эвристики списка; алгоритмы, базирующиеся на дублировании; алгоритмы кластеризации.

Планирование списком — это класс эвристик планирования, в котором задачам присваиваются приоритеты, задачи помещаются в список, упорядоченный по мере уменьшения величины приоритета. Решение о выборе задачи из списка для ее выполнения осуществляется на основе приоритета. Классическими примерами эвристик списка являются HEFT (Heterogeneous Earliest-Finish-Time) и FCP (Fast Critical Path).

Алгоритмы, базирующиеся на дублировании, отличаются стратегиями выбора задач для дублирования. Первоначально, алгоритмы этой группы применялись для неограниченного числа идентичных процессоров, таких как многопроцессорные системы с распределенной памятью. Также они имеют более высокую сложность, чем алгоритмы, обсуждавшиеся выше. Алгоритмы, основанные на дублировании, в распределенных вычислительных средах имеют дело только с независимыми заданиями.

Основная идея *алгоритмов кластеризации* заключается в объединении взаимосвязанных задач в маркированные группы для дальнейшего их отображения на одну группу ресурсов. Примерами алгоритмов кластеризации являются алгоритмы KB/L, алгоритм Саркара и алгоритм DSC.

Во второй главе, «Планирование в проблемно-ориентированных средах», описывается новый проблемно-ориентированный алгоритм планирования ресурсов *POS* для заданий с потоковой структурой, ориентированный на распределенные вычислительные среды, формируемые на базе вычислительных кластеров с многоядерными ускорителями.

Граф задания определяется как размеченный взвешенный ориентированный ациклический граф $G = \langle V, E, init, fin, \delta, \gamma \rangle$, где V — множество вершин, соответствующих задачам, E — множество дуг, соответствующих потокам данных. Вес $\delta(e)$ дуги e определяет объем данных, который необходимо передать по дуге e от задачи, ассоциированной с вершиной $init(e)$, к задаче, ассоциированной с вершиной $fin(e)$. Метка

$$\gamma(v) = (m_v, t_v) \quad (1)$$

определяет максимальное количество процессорных ядер m_v , на которых задача v имеет ускорение, близкое к *линейному*, и время t_v выполнения задачи v на одном ядре. Данная модель предполагает, что *вычислительная стоимость* $\chi(v, j_v)$ задачи v на j_v процессорных ядрах определяется следующей формулой:

$$\chi(v, j_v) = \begin{cases} t_v/j_v, & \text{если } 1 \leq j_v \leq m_v; \\ t_v/m_v, & \text{если } m_v < j_v. \end{cases} \quad (2)$$

Вычислительный узел P — это упорядоченное множество процессорных ядер $\{c_0, \dots, c_{d-1}\}$. *Вычислительная система* — это упорядоченное множество вычислительных узлов $\mathfrak{P} = \{P_0, \dots, P_{k-1}\}$. *Кластеризацией* называется однозначное отображение $\omega: V \rightarrow \mathfrak{P}$ множества вершин V графа задания G на множество вычислительных узлов \mathfrak{P} . Пусть задана вычислительная система $\mathfrak{P} = \{P_0, \dots, P_{k-1}\}$, состоящая из k узлов. *Кластер* W_i — это подмножество всех вершин, отображаемых на вычислительный узел $P_i \in \mathfrak{P}$:

$$W_i = \{v \in V \mid \omega(v) = P_i \in \mathfrak{P}\}. \quad (3)$$

Пусть задан граф задания $G = \langle V, E, init, fin, \delta, \gamma \rangle$, для которого определена функция кластеризации $\omega(v)$. Будем называть такой граф *кластеризованным* и обозначать как $G = \langle V, E, init, fin, \delta, \gamma, \omega \rangle$. Определим функцию $\sigma: E \rightarrow \mathbb{Z}_{\geq 0}$, вычисляющую *коммуникационную стоимость* (время) передачи данных по дуге $e \in E$, следующим образом:

$$\sigma(e) = \begin{cases} 0, & \text{если } \omega(init(e)) = \omega(fin(e)); \\ \delta(e), & \text{если } \omega(init(e)) \neq \omega(fin(e)). \end{cases} \quad (4)$$

Пусть задан кластеризованный граф $G = \langle V, E, init, fin, \delta, \gamma, \omega \rangle$. *Расписание* — это отображение $\xi: V \rightarrow \mathbb{Z}_{\geq 0} \times \mathbb{N}$, которое произвольной вершине $v \in V$ сопоставляет двойку чисел

$$\xi(v) = (\tau_v, j_v), \quad (5)$$

где τ_v определяет время запуска задачи v , j_v — количество процессорных ядер, выделяемых задаче v . Обозначим через s_v время останова задачи v . Имеем

$$s_v = \tau_v + \chi(v, j_v), \quad (6)$$

где χ — функция временной сложности, определенная с помощью формулы (2). Расписание называется *корректным*, если оно удовлетворяет следующим условиям:

$$\forall e \in E \left(\tau_{fin(e)} \geq \tau_{init(e)} + \chi(init(e), j_{init(e)}) + \sigma(e) \right); \quad (7)$$

$$\forall v \in V (j_v \leq m_v); \quad (8)$$

$$\forall t \in \mathbb{N} \left(\forall i \in [0, \dots, k-1] \left(\sum_{\substack{v \in W_i \wedge \\ \tau_v < t \leq s_v}} j_v \leq |P_i| \right) \right). \quad (9)$$

Ярусно-параллельной формой (ЯПФ) называется разбиение множества вершин V ориентированного ациклического графа $G = \langle V, E, init, fin \rangle$ на перенумерованные подмножества (*ярусы*) L_i ($i = 1, \dots, r$), удовлетворяющие следующим свойствам:

$$\left. \begin{aligned} &V = \bigcup_{i=1}^r L_i; \\ &\forall i \neq j \in \{1, \dots, r\} (L_i \cap L_j = \emptyset); \\ &\forall (v_1, v_2) \in E (\forall i \neq j \in \{1, \dots, r\} (v_1 \in L_i \wedge v_2 \in L_j \Rightarrow i < j)). \end{aligned} \right\} \quad (10)$$

Пусть в распланированном графе $G = \langle V, E, init, fin, \delta, \gamma, \omega, \xi \rangle$ задан путь $y = (e_1, e_2, \dots, e_n)$. *Стоимостью* пути $u(y)$ называется величина

$$u(y) = \chi(fin(e_n), j_{fin(e_n)}) + \sum_{i=1}^n \left(\chi(init(e_i), j_{init(e_i)}) + \max(\sigma(e_i), \tau_{fin(e_i)} - s_{init(e_i)}) \right), \quad (11)$$

где χ — функция, определяемая формулой (2), значением которой является вычислительная стоимость вершины; σ — функция, определяемая формулой (4), значением которой является коммуникационная стоимость дуги; j_v и τ_v определяется по формуле (5); s_v определяется по формуле (6).

Пусть Y — множество всех путей в распланированном графе $G = \langle V, E, init, fin, \delta, \gamma, \omega, \xi \rangle$. Путь $\bar{y} \in Y$ называется *критическим*, если

$$u(\bar{y}) = \max_{y \in Y} u(y). \quad (12)$$

Разработанный в диссертационной работе новый *алгоритм POS (Problem-Oriented Scheduling)* планирования ресурсов для распределенных проблемно-ориентированных сред представляется в виде трехуровневой структуры процедур. Процедура первого уровня является головной. Шаг процедуры первого уровня может представлять собой вызов процедуры второго уровня. Такой шаг выделяется полужирным шрифтом. Аналогичный подход может быть применен в описании шагов процедур второго уровня. Пусть задана вычислительная система в виде упорядоченного множества вычислительных узлов: $\mathfrak{P} = \{P_0, \dots, P_{k-1}\}$. Пусть имеется граф задания $G = \langle V, E, init, fin, \delta, \gamma \rangle$. Пусть выполняются следующие условия:

$$|V| \leq |\mathfrak{P}|; \quad (13)$$

$$\forall v \in V \left(\forall P \in \mathfrak{P}(m_v \leq |P|) \right), \quad (14)$$

где m_v — порог линейной масштабируемости, задаваемый функцией разметки γ . Зададим для графа G разбиение в каноническую ЯПФ с ярусами L_i ($i = 1, \dots, r$). Пронумеруем вершины $V = \{v_1, \dots, v_q\}$ графа G таким образом, чтобы выполнялось следующее свойство:

$$\forall i, j \in \{1, \dots, q\} \left((v_i \in L_a \wedge v_j \in L_b \wedge a < b) \Rightarrow i < j \right), \quad (15)$$

то есть на нижних ярусах располагаются вершины с большими номерами.

Головная процедура:

- Шаг 1. Построить начальную конфигурацию G_0 ;
- Шаг 2. $i := 0$;
- Шаг 3. Построить конфигурацию G_{i+1} ;
- Шаг 4. Если остались нерассмотренные дуги, $i := i + 1$ и перейти на шаг 3;
- Шаг 5. Уплотнить конфигурацию G_{i+1} ;
- Шаг 6. Стоп.

Процедура построения начальной конфигурации G_0 :

Шаг 1.1. Зададим функцию начальной кластеризации ω_0 следующим образом:

$$\forall i \in \{1, \dots, q\} \left(\omega_0(v_i) = P_{i-1} \right). \quad (16)$$

Шаг 1.2. Зададим начальное расписание $\xi_0(v) = (\tau_v, j_v)$: определим время запуска τ_v итерационно по уровням ЯПФ:

$$\left. \begin{array}{l} \forall v \in L_1 \left(\tau_v := 0 \right); \\ \forall v \in L_{i>1} \left(\tau_v := \max_{\substack{v' \in L_i; \\ v' \in L_{j \leq i}}} (\lambda(v', v'')) \right). \end{array} \right\} \quad (17)$$

Здесь

$$\lambda(v', v'') = \begin{cases} s_{v'}, & \text{если } (v', v'') \notin E; \\ s_{v'} + \sigma((v'', v')), & \text{если } (v', v'') \in E; \end{cases} \quad (18)$$

где $s_{v'}$ вычисляется по формуле (6). Определим количество ядер j_v , выделяемых вершине v , следующим образом:

$$\forall v \in V \left(j_v = m_v \right). \quad (19)$$

Шаг 1.3. $G_0 := \langle V, E, init, fin, \delta, \gamma, \omega_0, \xi_0 \rangle$;

Шаг 1.4. Конец процедуры.

Утверждение. Расписание ξ_0 , построенное в приведенной процедуре, является корректным.

Определим *субкритический путь*, как путь, имеющий максимальную стоимость среди всех путей, содержащих хотя бы одну нерассмотренную дугу.

Процедура построения конфигурации G_{i+1} :

- Шаг 3.1. Найти в G_i субкритический путь $\tilde{y}_i = (e_1, \dots, e_n)$ (если таких путей несколько, выбрать любой из них);
- Шаг 3.2. Найти первую от начала пути нерассмотренную дугу e_j ($1 \leq j \leq n$) в \tilde{y}_i и пометить ее как рассмотренную;
- Шаг 3.3. Если $i = 0$, то пометить вершину $init(e_j)$ как зафиксированную;

- Шаг 3.4. Если вершины $init(e_j)$ и $fin(e_j)$ зафиксированы, то перейти на шаг 3.14;
- Шаг 3.5. Если вершина $fin(e_j)$ не зафиксирована, то $v'' := fin(e_j), v' := init(e_j)$;
- Шаг 3.6. Если вершина $init(e_j)$ не зафиксирована, то $v'' := init(e_j), v' := fin(e_j)$;
- Шаг 3.7. Построить функцию кластеризации ω_{i+1} , отличающуюся от функции ω_i только в одном значении: $\omega_{i+1}(v'') := \omega_i(v')$;
- Шаг 3.8. **Построить расписание** ξ_{i+1} ;
- Шаг 3.9. $G_{i+1} := \langle V, E, init, fin, \delta, \gamma, \omega_{i+1}, \xi_{i+1} \rangle$
- Шаг 3.10. Найти критический путь \bar{y}_i в G_i (если таких путей несколько выбрать любой из них);
- Шаг 3.11. Найти критический путь \bar{y}_{i+1} в G_{i+1} (если таких путей несколько выбрать любой из них);
- Шаг 3.12. Если $u(\bar{y}_{i+1}) \leq u(\bar{y}_i)$, то перейти на шаг 3.16;
- Шаг 3.13. $G_{i+1} := G_i$
- Шаг 3.14. Если в \tilde{y}_i остались нерассмотренные дуги, перейти на шаг 3.2;
- Шаг 3.15. Если в G_i остались нерассмотренные дуги, перейти на шаг 3.1.
- Шаг 3.16. Конец процедуры.

Введем следующие обозначения: $T(x)$ — номер яруса, которому принадлежит вершина x ; $W_{\omega_i(x)} = \{v \mid v \in V, \omega_i(v) = \omega_i(x)\}$ — кластер, которому принадлежит вершина x .

Процедура построения расписания ξ_{i+1} :

- Шаг 3.8.1. $R := W_{\omega_i(v')} \cap L_{T(v'')} ;$
- Шаг 3.8.2. Если $R = \emptyset$ или $\sum_{v \in R} j_v \leq \left| P_{\omega_i(v')} \right|$ то перейти на шаг 3.8.7;
- Шаг 3.8.3. Для $h = q, \dots, T(fin(e_j)) + 1$ выполнить $L_{h+1} := L_h ;$
- Шаг 3.8.4. $L_{T(v'')+1} := \{v''\} ;$
- Шаг 3.8.5. $L_{T(v')} := L_{T(v')} \setminus \{v''\} ;$
- Шаг 3.8.6. $q := q + 1 ;$
- Шаг 3.8.7. Построить новое расписание ξ_{i+1} путем вычисления времени запуска τ_v всех вершин $v \in V$ с помощью формулы (17);
- Шаг 3.8.8. Пометить вершину v'' как зафиксированную;
- Шаг 3.8.9. Конец процедуры.

В третьей главе, «Реализация брокера ресурсов для проблемно-ориентированных сред», описывается процесс проектирования и реализации программной системы *DiVTB Broker*, представляющей собой брокер ресурсов для проблемно-ориентированных распределенных вычислительных сред. На рис. 2 приведены варианты использования брокера ресурсов. *DiVTB Server* представляет собой программный компонент, взаимодействующий с брокером ресурсов. *DiVTB Broker* отвечает за исполнение заданий и их мониторинг в системе *DiVTB*. Брокер ресурсов допускает вызов с помощью веб-методов и может быть использован веб-сервисами. *DiVTB Collector* представляет собой вспомогательный программный компонент, взаимодействующий с брокером ресурсов. Он осуществляет сбор информации о вычислительных ресурсах в распределенной вычислительной среде.

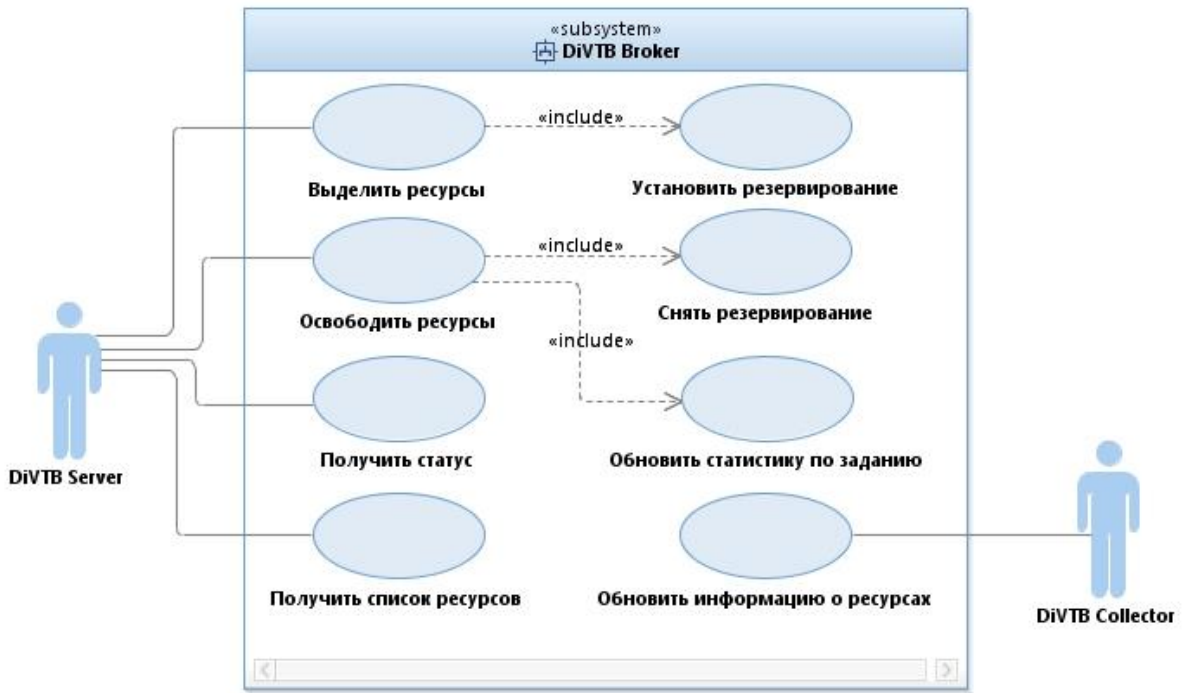


Рис. 2. Диаграмма вариантов использования DiVTB Broker.

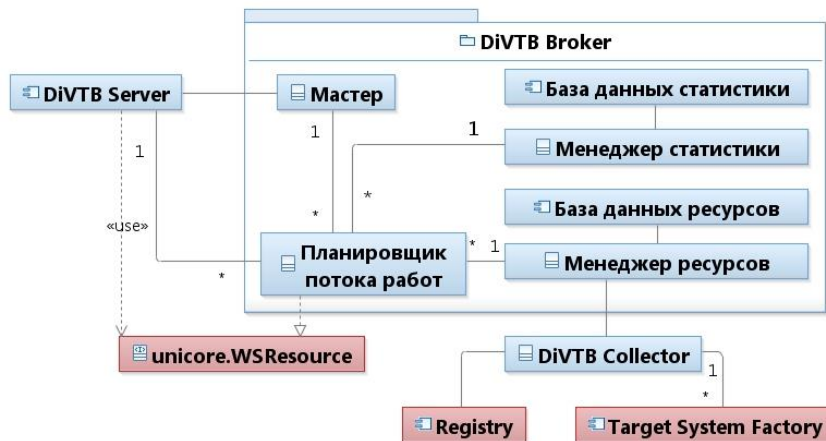


Рис. 3. Архитектура брокера ресурсов DiVTB Broker.

Согласно представленной на рис. 2 диаграмме вариантов использования была разработана архитектура DiVTB Broker, изображенная на рис. 3. DiVTB Broker состоит из следующих компонентов. Мастер принимает запросы от DiVTB Server и создает экземпляр планировщика потока задач, представляющий собой WS-ресурс в терминах UNICORE. Экземпляр планировщика потока задач осуществляет обработку одного запроса, формирует список требуемых для исполнения задания ресурсов и производит их резервирование. Менеджер ресурсов управляет базой данных ресурсов, содержащей информацию о целевых системах и резервировании ресурсов. Менеджер статистики управляет базой данных статистики, содержащей информацию о статистике выполнения задач. Коллектор работает независимо от DiVTB Broker и осуществляет сбор информации для базы данных ресурсов. Для всех основных функций строятся диаграммы последовательности UML, с помощью которых описываются принципы работы брокера ресурсов DiVTB Broker. DiVTB Broker был реализован на языке Java. Брокер ресурсов представляет собой сервис для платформы UNICORE. Исходные тексты брокера ресурсов свободно доступны в сети Интернет по адресу: <https://bitbucket.org/shamakina/pos>.

В четвертой главе, «Вычислительные эксперименты», приводятся результаты вычислительных экспериментов по исследованию алгоритма POS, разработанного в диссертационной работе. Используются следующие два класса заданий. *Класс MKO (многокритериальная оптимизация)* представляет вычислительные задания по многокритериальной оптимизации, составляющие большой процент загрузки современных суперкомпьютерных и распределенных вычислительных систем. *Класс СЗ (случайные задания)* представляет случайные задания с различным числом вершин и дуг. В классе СЗ выделены следующие *три группы заданий* (здесь T — среднее время выполнения задачи на одном ядре, Δ — средний вес дуги).

1. $M1$ — сбалансированные графы заданий, у которых $T/\Delta \in (0,8,1,2)$.
2. $M2$ — крупнозернистые графы заданий, у которых $T/\Delta \in (3,10)$.
3. $M3$ — мелкозернистые графы заданий, у которых $T/\Delta \in (0,1,0,3)$.

Эффективность алгоритма POS исследовалась в сравнении с двумя известными алгоритмами планирования: DSC и Min-Min. В экспериментах варьировались высота l и ширина w графа. Результаты представлены в табл. 1–3. Для сравнения алгоритма POS с алгоритмами DSC и Min-Min вычислялась *относительная эффективность* $\left(1 - \frac{T_{pos}}{T}\right) \cdot 100\%$, T_{dsc} — среднее время выполнения задания для алгоритма DSC, T — время выполнения задания алгоритмом DSC или Min-Min соответственно. Во всех случаях алгоритм POS, разработанный в диссертационной работе, демонстрирует существенное ускорение времени выполнения задания по сравнению с алгоритмами DSC и Min-Min. При этом алгоритм POS существенно превосходит DSC по плотности получаемого расписания и показывает результаты, сравнимые по этому показателю с алгоритмом Min-Min.

Табл. 1. Сравнение алгоритмов POS, DSC и Min-Min для группы M1.

№ п/п	l	w	$ V $	$ E $	Количество задействованных вычислительных узлов			Относительная эффективность	
					POS	DSC	MinMin	POS/DSC	POS/MinMin
1	5	10-20	51	126	7	21	4	73,57%	88,72%
2	10	10-20	117	296	4	36	4	72,76%	89,51%
3	10	20-30	211	505	16	68	6	73,45%	88,04%
4	20	5-10	137	252	2	46	2	90,18%	94,36%
Среднее значение								77,49%	90,16%

Табл. 2. Сравнение алгоритмов POS, DSC и Min-Min для группы M2.

№ п/п	l	w	$ V $	$ E $	Количество задействованных вычислительных узлов			Относительная эффективность	
					POS	DSC	MinMin	POS/DSC	POS/MinMin
1	5	10-20	49	113	3	25	4	85,00%	82,35%
2	10	10-20	130	390	12	42	4	79,14%	60,16%
3	10	20-30	206	464	17	73	6	75,32%	71,14%
4	20	5-10	141	290	13	41	2	79,14%	32,17%
Среднее значение								79,65%	61,46%

Табл. 3. Сравнение алгоритмов POS, DSC и Min-Min для группы М3.

№ п/п	l	w	$ V $	$ E $	Количество задействованных вычислительных узлов			Относительная эффективность	
					POS	DSC	MinMin	POS/DSC	POS/MinMin
1	5	10-20	59	190	4	21	4	77,80%	92,40%
2	10	10-20	123	378	3	34	4	79,55%	97,64%
3	10	20-30	201	453	9	61	6	51,36%	90,72%
4	20	5-10	133	287	2	29	2	68,04%	87,31%
Среднее значение								69,19%	92,02%

В заключении суммируются основные результаты диссертационной работы, выносимые на защиту, приводятся данные о публикациях и апробациях автора по теме диссертации, и рассматриваются направления дальнейших исследований в данной области.

Основные результаты диссертационной работы

На защиту выносятся следующие новые научные результаты.

1. Разработаны формальные методы представления проблемно-ориентированных распределенных вычислительных сред.
2. Разработан проблемно-ориентированный алгоритм планирования ресурсов *POS* для заданий с потоковой структурой.
3. Разработан брокер ресурсов *DiVTB* для проблемно-ориентированных распределенных вычислительных сред, реализующий алгоритм планирования *POS*.
4. Проведены вычислительные эксперименты, подтверждающие эффективность предложенных подходов.

Публикации по теме диссертации

Статьи в журналах из перечня ВАК

1. Радченко Г.И., Соколинский Л.Б., Шамакина А.В. Модели и методы профилирования и оценки времени выполнения потоков работ в суперкомпьютерных системах // Вычислительные методы и программирование: Новые вычислительные технологии. 2013. Т. 14, Вып. 4. С. 96–103.
2. Шамакина А.В. CAEBeans Broker: брокер ресурсов системы CAEBeans // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». 2010. № 16(192). С. 107–115.
3. Бухановский А.В., Марьин С.В., Князьков К.В., Сиднев А.А., Жабин С.Н., Баглий А.П., Штейнберг Р.Б., Шамакина А.В., Воеводин Вад.В., Головченко Е.Н., Фалалеев Р.Т., Духанов А.В., Тарасов А.А., Шамардин Л.В., Моисеенко А.И. Результаты реализации проекта «Мобильность молодых ученых» в 2010 году: развитие функциональных элементов технологии IPSE и расширение состава прикладных сервисов // Известия высших учебных заведений. Приборостроение. 2011. Т. 54, № 10. С. 80–86.
4. Московский А.А., Перминов М.П., Соколинский Л.Б., Черепенников В.В., Шамакина А.В. Исследование производительности суперкомпьютеров семейства «СКИФ Аврора» на

индустриальных задачах // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». 2010. № 35(211). С. 66–78.

5. *Шамакина А.В.* Организация брокера ресурсов в системе CAEBeans // Вестник Южно-Уральского государственного университета. Серия «Математическое моделирование и программирование». 2008. № 27(127). Вып. 2. С. 110–116.

Статья, индексируемая в SCOPUS

6. *Shamakina A.* Brokering Service for Supporting Problem-Oriented Grid Environments // UNICORE Summit 2012 Proceedings, Forschungszentrum Julich, 2012. P. 67–75.

Статьи в изданиях, индексируемых в РИНЦ

7. *Шамакина А.В.* Обзор технологий распределенных вычислений // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2014. Т. 3, № 3. С. 51–85.
8. *Шамакина А.В.* Брокер ресурсов для поддержки проблемно-ориентированных сред // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2012. № 46(305). Вып. 1. С. 88–98.
9. *Шамакина А.В., Соколинский Л.Б.* Формальная модель задания в распределенных вычислительных средах // Параллельные вычислительные технологии (ПаВТ'2014): труды международной научной конференции (1–3 апреля 2014 г., г. Ростов-на-Дону). Челябинск: Издательский центр ЮУрГУ, 2014. С. 343–354.
10. *Московский А.А., Перминов М.П., Соколинский Л.Б., Черепенников В.В., Шамакина А.В.* Опыт использования суперкомпьютера "СКИФ Аврора" для решения научно-технических задач // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.). Челябинск: Издательский центр ЮУрГУ, 2010. С. 258–267.
11. *Шамакина А.В.* CAEBeans Broker: брокер ресурсов системы CAEBeans // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.). Челябинск: Издательский центр ЮУрГУ, 2010. С. 643–650.

Другие публикации

12. *Шамакина А.В.* Алгоритм планирования ресурсов POS для распределенных проблемно-ориентированных сред // Научный сервис в сети Интернет: многообразие суперкомпьютерных миров: Труды Международной суперкомпьютерной конференции (22–27 сентября 2014 г., Новороссийск). М.: Изд-во МГУ, 2014. С. 124–136.
13. *Шамакина А.В.* Протокол взаимодействия с брокером ресурсов в системе CAEBeans // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской научной конференции (21–26 сентября 2009 г., Новороссийск). М.: Изд-во МГУ, 2009. С. 400–402.
14. *Шамакина А.В.* Организация брокера ресурсов в системе CAEBEANS // Научный сервис в сети Интернет: решение больших задач: Труды Всероссийской научной конференции (22–27 сентября 2008 г., Новороссийск). М.: Изд-во МГУ, 2008. С. 326–327.

Свидетельства о регистрации программ и баз данных

15. *Шамакина А.В.* Свидетельство Роспатента о государственной регистрации программы для ЭВМ «RaVIS Broker» № 2011610338 от 11.01.2011, правообладатели: ИПС им. А.К. Айламазяна РАН, ГОУ ВПО «ЮУрГУ».
16. *Юрков В.В., Дорохов В.А., Радченко Г.И., Насибулина Р.С., Шамакина А.В.* Свидетельство Роспатента о государственной регистрации программы для ЭВМ «CAEBeans Toolbox: программная среда для разработки проблемно-ориентированных оболочек для грид» № 2008614485 от 03.10.2008, правообладатель: ГОУ ВПО «ЮУрГУ».
17. *Юрков В.В., Дорохов В.А., Радченко Г.И., Насибулина Р.С., Шамакина А.В.* Свидетельство Роспатента о государственной регистрации программы для ЭВМ «CAEBeans Sphere: программное средство для поддержки распределенных вычислительных сред на базе платформы Microsoft.NET» № 2008614486 от 03.10.2008, правообладатель: ГОУ ВПО «ЮУрГУ».
18. *Радченко Г.И., Насибулина Р.С., Шамакина А.В., Юрков В.В., Федянин О.Н., Дорохов В.А.* Свидетельство Роспатента о государственной регистрации программы для ЭВМ «Пакет проблемно-ориентированных оболочек CAEBeans для решения типовых инженерных задач» № 2008611898 от 04.05.2008, правообладатель: ГОУ ВПО «ЮУрГУ».
19. *Радченко Г.И., Шамакина А.В., Худякова Е.С., Репина К.В., Захаров Е.А.* Свидетельство Роспатента о государственной регистрации базы данных «Справочно-библиографический ресурс «Проблемно-ориентированные грид-сервисы в распределенных вычислительных средах» № 2013620973 от 22.08.2013, правообладатель: ФГБОУ ВПО «ЮУрГУ» (НИУ).
20. *Радченко Г.И., Шамакина А.В., Худякова Е.С., Репина К.В.* Свидетельство Роспатента о государственной регистрации базы данных «Справочно-библиографический ресурс «Распределенная обработка данных и облачные вычислительные среды» № 2013621019 от 27.08.2013, правообладатель: ФГБОУ ВПО «ЮУрГУ» (НИУ).
21. *Соколинский Л.Б., Радченко Г.И., Шамакина А.В.* Свидетельство Роспатента о государственной регистрации базы данных «Справочно-библиографический ресурс «Грид-технологии» № 2013620033 от 09.01.2013, правообладатель: ФГБОУ ВПО «ЮУрГУ» (НИУ).
22. *Соколинский Л.Б., Радченко Г.И., Шамакина А.В.* Свидетельство Роспатента о государственной регистрации базы данных «Справочно-библиографический ресурс «Суперкомпьютерные технологии» № 2013620032 от 09.01.2013, правообладатель: ФГБОУ ВПО «ЮУрГУ» (НИУ).

Работа выполнялась при финансовой поддержке Минобрнауки РФ в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (гос. контракт № 14.574.21.0035).

Подписано в печать «23» октября 2014 г.
Формат 60x84 1/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 1,0. Уч.-изд. л. 1,2.
Тираж 100 экз.

Типография «Фотохудожник»
454111, г. Челябинск, ул. Свободы, 155/1