# О задаче делегирования нагрузки высокопроизводительных вычислительных комплексов в распределенные сети

Г.И. Евтушенко, К.В. Иванов, А.Б. Новиков

ФГУП «Всероссийский научно-исследовательский институт автоматики им. Н.Л.Духова»

Предлагается стратегия объединения высокопроизводительных вычислительных комплексов и гетерогенных неотчуждаемых ресурсов компьютерного парка предприятия, которая позволяет сократить временные издержки при ожидании вычислительных задач в очереди и, таким образом, повысить качество сервиса. Предложена формализация планирования параллельных заданий в контексте делегирования нагрузки высокопроизводительных вычислительных комплексов в грид-системы с неотчуждаемыми ресурсами. Разработана архитектура системы управления заданиями на основе виртуализации.

*Ключевые слова:* распределенные вычисления, вычислительный грид, неотчуждаемые ресурсы, параллельные вычисления, алгоритмы планирования.

## 1. Введение

В различных сферах деятельности промышленных предприятий, занимающихся научно-исследовательскими и опытно-конструкторскими работами (НИОКР), присутствует множество ресурсоемких задач, требующих интенсивных вычислений с использованием специализированных параллельных и распределенных прикладных приложений. Необходимость в сложных расчетах обуславливает потребность в большом количестве вычислительных ресурсов. Российские федеральные ядерные центры (РФЯЦ) для проведения подобных расчетов оборудуются высокопроизводительными вычислительными комплексами. При этом парк персональных компьютеров не используется в выполнении научных и технических расчетов, отличаясь значительным временем простоя ресурсов.

Для указанных предприятий подходящей областью, позволяющей утилизировать вычислительные ресурсы, являются вычисления в слабосвязанных гетерогенных неотчуждаемых ресурсах. Сети из подобных ресурсов принято называть грид-системой. Идея эффективного использования компьютерного парка промышленных предприятий состоит в задействовании большой суммарной производительности узлов с малой средней загрузкой. Это позволит снизить нагрузку на суперкомпьютер и повысить качество сервиса, сократив время постановки задачи на расчет. Таким образом, прогнозируя загрузку всего высокопроизводительного вычислительного комплекса как единого целого, можно сделать вывод о том, когда следует делегировать нагрузку из суперкомпьютера в грид. Подобное делегирование нагрузки необходимо в те моменты, когда высокая нагрузка смогла бы парализовать очередь заданий.

Одной из проблем в использовании ресурсов парка персональных компьютеров предприятия заключается в том, что компьютеры могут присоединяться и отсоединяться от сети в произвольные моменты времени. Помимо этого парк персональных компьютеров существенно неоднороден [1] как по типам ресурсов, так и по программному обеспечению.

Задача эффективного выполнения параллельных программ на грид-системах с неотчуждаемыми ресурсами не имеет универсального решения. Область взаимодействия грид-системы и суперкомпьютера рассматривается либо со стороны использования существующих суперкомпьютеров для выполнения задач грид-систем [2], либо со стороны объединения нескольких суперкомпьютеров в грид-систему. К подобному состоянию вопроса приводят

задержки во взаимодействии удаленных узлов, предъявляющих требования к максимальному разбиению задачи на независимые части с высокой вычислительной мощностью [1]. Мы предполагаем, что широкое использование аппарата прогнозирования позволит решать параллельные задачи в распределенных средах состоящих из ресурсов компьютерного парка предприятия.

Под гетерогенной средой будем понимать ресурсы суперкомпьютера и компьютерного парка предприятия. Эффективность использования ресурсов напрямую зависит от успешности решения задачи планирования в гетерогенной среде. Требования сокращения временных издержек на решение прикладных задач, упрощения процедуры сопровождения распределенных систем обработки данных в существующих условиях обосновывают актуальность разработки новых методов планирования задач в распределенных системах обработки данных [4].

## 2. Планирование параллельных и распределенных задач в грид с неотчуждаемыми ресурсами

На данный момент существует направление по расширению вычислительных ресурсов кластера за счет внешних ресурсов. Одним из решения является добавление вычислительных ресурсов из публичного облака в кластер на нужное время. При этом, не требуется изменение метода планирования распределения заданий, рабочего окружения, интерфейса работы с суперкомпьютером и т.д. В масштабе промышленных предприятий существует возможность использования уже приобретенных вычислительных ресурсов, избегая дополнительных затрат на облачные технологии. Также стоит отметить прочие ограничения на проведение расчетов в сторонних коммерческих организациях, связанных со спецификой режимных предприятий. Таким образом, одним из решений задачи делегирования нагрузки суперкомпьютера является использование незадействованных ресурсов персональных компьютеров пользователей. Существующие работы по управлению заданиями в неотчуждаемом гриде делятся на два типа. Первые рассматривают грид с неотчуждаемыми некластеризованными ресурсами, но при этом модель не учитывает возможность обработки параллельных приложений [3]. Вторые отличаются методами планирования которые гарантируют только факт запуска параллельного расчета [5]. Применение методов, полученных в работе [3], затруднено ввиду отсутствия возможности запуска параллельных расчетов с суперкомпьютера на грид-системе. В свою очередь, интеграция работы [5] не эффективна по критерию времени расчета, так как отключение персонального компьютера, на котором проводился расчет параллельного приложения, приведет к завершению работы всего параллельного приложения. Необходима разработка нового метода управления заданиями, направленного на повышение вероятности выполнения параллельного расчета в грид-системе с неотчуждаемыми гетерогенными ресурсами.

## 3. Математическая модель предметной области

Функцию назначения задач $T$ на узлы обработки $N$ в соответствии с методом планирования $f$ можно представить в виде черного ящика:

$$f(T, N) = W$$

где $t \in T$ задачи из очереди, $n \in N$ узлы обработки, $w \in W$ элемент матрицы назначения, являющейся результатом планирования заданий. Элементы матрицы назначения:

$$W = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1c_n} \\ w_{21} & w_{22} & \dots & w_{2c_n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{c_t1} & w_{c_t2} & \dots & w_{c_tc_n} \end{pmatrix}$$

где $c_t$ - количество задач в очереди $c_n$ - количество доступных узлов,описываются как:

$$w_{ij} = \begin{cases} 0 & \text{если задание } i \text{ не назначено на узел } j \\ a_{opt} & \text{если задание } i \text{ назначено на узел } j \end{cases}$$

где $a_{opt}$ является конфигурация задачи приводящей к минимуму целевой функции. Подобное описание позволяет пользователю указать несколько вариантов конфигурации запуска его приложения.

Вычислительный узел описывается как:

$$n = (f_{r_1}, f_{r_2}, \dots, f_{r_n})$$

где $f_{r_n}$ - функция распределения вероятности доступности ресурса, выражающаяся как:

$$f_{r_i} : \{G, L\} \to \bigcup_{t}^{t_h} \{(\mathbb{N}, \mathbb{Q})_1^t \dots (\mathbb{N}, \mathbb{Q})_m^t\}$$

где $m$ - максимальное количество сервис-слотов для данного ресурса данного узла, $t_h$ - горизонт прогнозирования. Под сервис-слотом здесь понимается атомарное количество ресурса. Для CPU под сервис-слотом может пониматься ядро, для RAM - 128 Мб памяти. Таким образом $f_{r_i}$ представляет собой отображение глобального состояния грид-системы и локального состояния узла на набор вероятностей доступности определенного количества сервис-слотов на $t_h$ интервалов прогноза. Тут $G$ - вектор глобального состояния грид-среды, $L$ - вектор локального состояния узла-обработчика. Данная функция приобретает смысл только в момент планирования, так как существенно зависит от задачи. Система планирования, работая над созданием матрицы назначения $W$, «примеряет» задачи на доступные узлы. Так в векторе локального состояния узла появляются такие элементы как $T_{sub}$ и $T_{wall}$, соответствующие прогнозируемому времени постановки задачи на расчет и времени отведенному пользователем на расчет данной задачи. В качестве примера рассмотрим ограничение выделенного на расчет времени в системе с отчуждаемыми ресурсами:

$$f_{r_t}(G, L) = (T_{sub} + T_{wall} - t, 1.0)$$

где $T_{sub} \in L, T_{wall} \in L, t \in G$, а единица означает что узел-обработчик не будет выключен до завершения расчета. При подобной постановке поведение задачи на даном узле может дополнительно характеризоваться разрывами этой линейной функции. Обработка заданий предполагается пакетной (централизованной), при этом распределению подвергаются сами функции распределения вероятности доступности ресурса. Обобщая функцию $f_{r_t}$ до характеристики доступных сервис-слотов времени на расчет задач в системах с неотчуждаемыми ресурсами, можно дополнить ее аппаратом прогнозирования завершения работы узла. Таким образом, используя в планировании функцию

$$f_{r_t} = \bigcup_{t}^{t_h} \{(1, f_{pr_t}(1))^t, \dots, (m, f_{pr_t}(m))^t\}$$

где $f_{pr_t}$ соответствующая функция прогнозирования доступности количества сервис-слотов типа $t$ (например [8]), $m$ - максимально возможное для данного узла количество сервис-слотов этого типа. Т.о. можно запрашивать у узла зависимость распределения доступных ресурсов от момента времени. Важным является то, что данные о максимальном количестве сервис-слотов определенных типов инкапсулированы непосредственно на узле-обработчике.

Задачу было решено представить в виде набора функций требований к ресурсам:

$$t = \{f_{req_0}, f_{req_1}, \ldots\}$$

Таким образом, была заложена возможность учета динамики требований к ресурсам. Обобщенное описание требования к ресурсам включает стандартные константные описания, т.е. они выражаются константными функциями.

Задача планирования представлена в виде многокритериальной оптимизации

$$min(-f_p, f_c)$$

т.е. максимизации функции вероятности завершения задачи на узлах $f_p$ и минимизации функции ресурсных затрат $f_c$. Функции $f_p$ и $f_c$ являются интегральными для всего плана (матрицы $W$) и участвуют непосредственно в функции соответствия (Fitness Function). Функция ресурсных затрат оперирует матрицей ресурсных затрат

$$C = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1c_n} \\ c_{21} & c_{22} & \ldots & c_{2c_n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{c_t1} & c_{c_t2} & \ldots & c_{c_tc_n} \end{pmatrix}$$

элементы которой $c \in \mathbb{N}$ показывают суммарное количество простаивающих сервис-слотов при определенном распределении задач по узлам. Функция $f_p$ в свою очередь оперирует матрицей вероятностей элементы которой $p_{ij}$ равны вероятности, которую возвращает ресурсная функция узла $j$ для всех типов запрашиваемых задачей ресурсов. При этом, в случае, если запрошенное количество сервис-слотов больше теоретически доступного на узле - возвращается 0.

Таким образом достигается повышение вероятности завершения параллельно приложения на распределенных вычислительных сетях с неотчуждаемыми ресурсами. Метод планирования максимизирует вероятность успешного завершения расчета на всех выделенных для него узлах в условиях ресурсной динамики, динамики требований и позволяет децентрализовать аппарат прогноза, поручив каждому узлу прогнозировать свою ресурсную динамику. Касаемо построения архитектуры системы на базе прогнозирования, стоит отметить, что в некоторых работах [8,9] отмечается, возможная эффективность систем мониторинга вычислительных систем, основанных на нейросетевой базе.

## 4. Архитектура системы делегирования нагрузки ВВК в грид-системы

Базовыми требованиями к архитектуре системы являются: гибкость; простота масштабирования; отказоустойчивость. Архитектура должна отвечать нормам информационной безопасности.

Система управления заданиями разделена на 3 части (рисунок 1). PyGrid-Server – сервер, занимающийся планированием выполнения задания в гетерогенной вычислительной

сети с неотчуждаемыми ресурсами. По результатам планирования сервер передает или снимает задания с узлов-обработчиков. PyGrid-Client – ПО, с помощью которого происходит взаимодействие с системой PyGrid. Обеспечивает функционал добавления, удаление и мониторинга статуса задачи. В предлагаемом прототипе системы, данная подпрограмма также служит для получения результата расчета. PyGrid-Agent – ПО, работающее на устройстве-обработчике. Отвечает за получение, выполнение и перенаправление вывода выполняющихся задач.

В предлагаемой архитектуре клиент не обязательно является узлом обработчиком. Данное решение позволяет в будущем обеспечивать межгрупповые расчеты, организуя иерархии вычислительных сетей и избегая при этом простоя их вычислительных ресурсов.

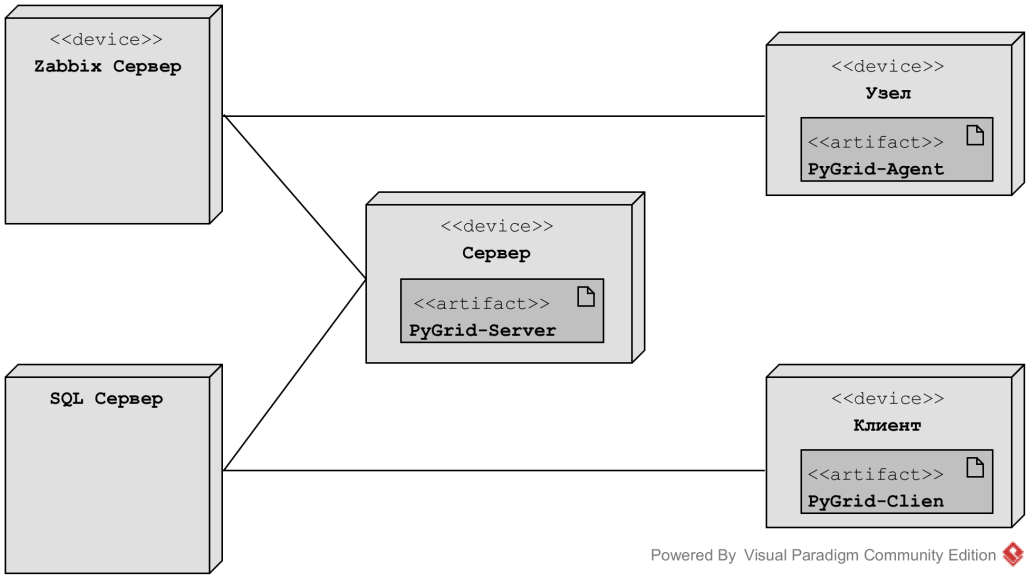


**Рис. 1.** Диаграмма пакетов

Клиентское приложение PyGrid-Client работает непосредственно с базой данных очереди заданий, с которой потом общается сервер. При использовании данного подхода отпадает необходимость в фоновом режиме работы сервера системы – PyGrid-Server. Это позволяет не только снизить количество кода, но и нагрузку на узел выполняющий роль сервера. При этом на SQL сервер перекладывается ответственность за безопасность системы.

Указанная архитектура (рисунок 1) является инфраструктурой для виртуальной среды. Через взаимодействие с Zabbix сервером происходит выбор соответствующего узла-обработчика, образ виртуальной машины (ВМ) и ее конфигурации. В разработанном прототипе системы функционал контроля виртуальной среды вынесен в отдельный сервис. После копирования, разворачивания и запуска виртуальной машины сервер планирования считает виртуальную машину обычным вычислительным узлом. Подобный механизм позволяет не только организовать среду, минимизирующую влияние на машины пользователей, но и позволит сохранить инфраструктуру суперкомпьютера, не перекомпилируя приложения для каждой архитектуры и ОС на которой возможен запуск. Примечательной является работа [7] в которой предложен подход к построению вычислительного грида основанного на комбинации классических виртуальных машин уровня ОС (VMs) и промежуточных механизмов управления в распределенной среде. В работе [7] даются качественные аргументы свидетельствующие о целесообразности применения подобного подхода в плане безопасности, изоляции, настройки, контроля версий и управления ресурсами, а также предоставили количественные результаты исследования производительности.

## 5. Заключение

Предложенная постановка задачи планирования позволяет, с одной стороны, абстрагироваться от конкретных методов моделирования поведения узлов грид-системы, с другой – организовать инфраструктуру, располагающую к исследованию динамики грид-системы методами анализа данных, что достигается за счет абстрагирования от типов ресурсов и оперирования полями вероятностей. Показана возможность построения системы, учитывающей ресурсную динамику, что особенно важно в грид-системах с неотчуждаемыми ресурсами. Управляя ресурсами виртуальной машины или учитывая заряд источника бесперебойного питания, мы можем существенно повысить вероятность выполнения параллельных приложений в грид-системах. Разработан прототип системы управления заданиями, который, в отличие от существующих технологий, позволяет реализовать рассматриваемую стратегию объединения суперкомпьютера и грид-системы с неотчуждаемыми ресурсами. Основные результаты данной работы:

- Задача планирования заданиями представлена как многокритериальная оптимизация;
- Разработана математическая модель планирования параллельных и распределенных заданий в контексте делегирования нагрузки суперкомпьютера в грид-системы, учитывающая динамику вычислительных ресурсов;
- Разработана архитектура системы планирования параллельных и распределенных заданий в грид-системах с неотчуждаемыми некластеризованными ресурсами;
- Разработан и реализован прототип виртуальной среды, системы управления заданиями и системы планирования.

## Литература

1. Воеводин В.В. Решение больших задач в распределенных вычислительных средах // Автоматика и Телемеханика. 2007. № 5. С. 32-45.

2. Демичев А.П., Ильин В.А., Крюков А.П. Введение в грид-технологии. Москва: Изд-во НИИЯФ МГУ, 2007. 87 с.

3. Березовский П.С. Управление заданиями в гриде с некластеризованными ресурсами. Москва, 2011, 128 с.

4. Голубев И.А. Планирование задач в распределенных вычислительных системах на основе метаданных. СПб, 2014, 136 с.

5. Коваленко В.Н., Коваленко Е.И., Корягин Д.А., Семячкин Д.А. Управление параллельными заданиями в гриде с неотчуждаемыми ресурсами // Препринты ИПМ № 63, Москва, 2007, 28 с.

6. Youcef Derbal. A Probabilistic Scheduling Heuristic for Computational Grids // Multiagent and Grid Systems 2006 vol: 2 (1) pp: 45-59

7. Figueiredo R, Dinda P.A. (2003). A Case For Grid Computing On Virtual Machines // In Proceedings of the 23rd ICDCS, p. 550.

8. Евтушенко Г.И. Сравнительный анализ применения искусственной нейронной сети прямого распространения и рекуррентной нейронной сети в задаче прогноза загрузки вычислительных ресурсов // ITIPM'2015, vol: 1 pp: 5-9, Ufa, Russia, 2015.

9. Иванов К.В. Система мониторинга с прогнозированием ошибок // Параллельные вычислительные технологии (ПаВТ'2013). 2013. С. 592.

# A high-performance computing systems workload delegation to grids

G.I. Evtushenko, K.V. Ivanov, A.B. Novikov

FSUE All-Russia Research Institute of Automatics

High-performance computing systems and heterogeneous dynamic enterprise computers resources combining strategy is offered. It allows to reduce job waiting time in queue. Parallel tasks scheduling formalization in context of high-performance computing systems workload delegation to the grid-systems with dynamic resources is offered. The job management system architecture based on virtualization is developed.

*Keywords:* distributed computing, grid computing, dynamic resources, parallel computing, scheduling algorithms

## References

1. Voevodin V.V. Reshenie bol'shikh zadach v raspredelyennykh vychislitel'nykh sredakh [Large problems solving in distributed computing environments] // Avtomatika i Telemekhanika [Automation and Remote Control]. 2007, P. 32-45.

2. Demichev A.P., Il'in V.A., Kryukov A.P. Vvedenie v grid-tekhnologii [Introduction to grid technologies]. Izd-vo NIIYaF MGU, 2007. 87 p.

3. Berezovskiy P.S. Upravlenie zadaniyami v gride s neklasterizovannymi resursami [Job control in nonclustered resources grid], 2011. 128 p.

4. Golubev I.A. Planirovanie zadach v raspredelyennykh vychislitel'nykh sistemakh na osnove metadannykh [Job scheduling in metadata based distributed computing systems based]. SPb, 2014. 136 p.

5. Kovalenko V.N., Kovalenko E.I., Koryagin D.A., Semyachkin D.A. Upravlenie parallel'nymi zadaniyami v gride s neotchuzhdaemymi resursami [Parallel jobs management in grid with dynamic resources], 2007. 28 p.

6. Youcef Derbal. A Probabilistic Scheduling Heuristic for Computational Grids // Multiagent and Grid Systems 2006 vol: 2 (1) pp: 45-59

7. Figueiredo R, Dinda P.A. (2003). A Case For Grid Computing On Virtual Machines // In Proceedings of the 23rd ICDCS, p. 550.

8. Evtushenko G.I. Sravnitel'nyy analiz primeneniya iskusstvennoy neyronnoy seti pryamogo rasprostraneniya i rekurrentnoy neyronnoy seti v zadache prognoza zagruzki vychislitel'nykh resursov [Comparative analysis of feedforward artificial neural networks and recurrent neural networks use in computing resources load forecasting problem] // The $3^{rd}$ International Conference on Intelligent Technologies for Information Processing and Management (ITIPM'2015), vol: 1, Ufa, Russia, 2015.

9. Ivanov K.V. Sistema monitoringa s prognozirovaniem oshibok [Monitoring system with the error prediction] // Parallel'nye vychislitel'nye tekhnologii (PaVT'2013)[Parallel Computational Technologies (PCT'2013)], 2013. P. 592.