

Модель проблемно-ориентированной облачной вычислительной среды¹

Г.И. Радченко <gleb.radchenko@susu.ru>
Южно-Уральский государственный университет,
454080, Россия, г. Челябинск, проспект Ленина, д. 76.

Аннотация. Для эффективного использования ресурсов высокопроизводительных вычислительных систем при реализации методов численного исследования физических, биологических, социальных и др. явлений могут быть использованы подходы предоставления распределенных проблемно-ориентированных вычислительных сред. Они обеспечивают пользователям прозрачный доступ к решению конкретных классов прикладных задач на базе доступных вычислительных ресурсов. Для повышения эффективности таких сред необходимо применение проблемно-ориентированных методов планирования вычислительных задач, использующих информацию о предметной области для прогнозирования вычислительных характеристик задач при планировании и распределении заданий. В статье представлены модели предметной области и проблемно-ориентированной облачной вычислительной среды, ориентированные на поддержку разработки новых проблемно-ориентированных алгоритмов планирования при выполнении расчетов в конкретных предметных областях на базе облачных вычислительных систем.

Ключевые слова: облачные вычисления; модель вычислительной среды; распределенные вычислительные системы; проблемно-ориентированная вычислительная среда

Ключевые слова: модели данных, объектная модель, ODMG 3.0, SQL, разыменованние объектных ссылок, размещение объектов.

DOI: 10.15514/ISPRAS-2015-27(6)-17

Для цитирования: Радченко Г.И. Модель проблемно-ориентированной облачной вычислительной среды. Труды ИСП РАН, том 27, вып. 6, 2015 г., стр. 275-284. DOI: 10.15514/ISPRAS-2015-27(6)-17.

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант № 15-29-07959 офи-м.

1. Введение

Использование методов суперкомпьютерного моделирования и интеллектуального анализа данных обеспечивает получение качественно новых результатов во всех отраслях знаний, позволяя проводить численные исследования физических, биологических, социальных и др. явлений, предоставляя реальную альтернативу дорогостоящим (или невозможным) экспериментам [1].

Этот тренд привел к появлению особой научной дисциплины, названной «Computational Science» («Вычислительная наука»). Вычислительная наука – это быстро развивающаяся мульти-дисциплинарная научная дисциплина, использующая передовые вычислительные методы для решения сложных задач, объединяющая в себе методы, алгоритмы и программное обеспечение для компьютерного моделирования, разработанные для решения сложных научных и инженерных задач; а также аспекты фундаментальной информатики и информационных технологий, обеспечивающие развитие аппаратных, программных, сетевых компонентов и СУБД, необходимых для решения таких вычислительно-сложных задач [2]. Примерами разделов вычислительной науки являются такие дисциплины как биоинформатика, вычислительная химия, вычислительная гидродинамика, вычислительная инженерия и др.

Процесс решения задач такого типа с использованием суперкомпьютерных ресурсов для рядового пользователя может быть сопряжен с определенными трудностями в связи с необходимостью специфических знаний, умений и навыков в области высокопроизводительных вычислений. Для эффективного использования ресурсов высокопроизводительных вычислительных систем и организации прозрачного доступа к распределенным вычислительным ресурсам могут быть использованы подходы предоставления распределенных проблемно-ориентированных вычислительных сред (ПВС). В таком случае, вместо прямого доступа к аппаратным интерфейсам удаленных вычислительных устройств, пользователю предоставляется прозрачный веб-интерфейс распределенной проблемно-ориентированной вычислительной среды, которая берет на себя задачи по решению конкретных классов прикладных задач на базе доступных вычислительных ресурсов, включая декомпозицию задания в иерархию вычислительных задач; поиск и выделение необходимых вычислительных ресурсов; мониторинг хода решения задач; передачу результатов решения задач пользователю [3–5].

Современным задачам вычислительной науки характерны высокие требования к предоставляемым вычислительным ресурсам, а также сложная вычислительная структура заданий, которую можно описать в виде потока работ [6]. Также, для задач такого рода характерны многовариантные расчеты, когда вычислительное задание запускается многократно с различными вариациями входных параметров [7]. Приложения такого рода составляют большой процент загрузки современных суперкомпьютерных и

распределенных вычислительных систем, что влечет необходимость в создании методов и алгоритмов эффективного распределения ресурсов таких систем для оптимизации решения таких задач. Одним из подходов, ориентированных на повышение эффективности таких сред, является применение проблемно-ориентированных методов планирования вычислительных задач [8].

В рамках данной статьи будет представлена модель проблемно-ориентированной облачной вычислительной среды, которая бы обеспечила поддержку разработки новых алгоритмов для планирования потоковых приложений при выполнении расчетов в конкретных предметных областях на базе облачных вычислительных систем.

Статья организована следующим образом. В разделе 1 даются основные определения и вводится понятие предметной области. Раздел 2 посвящен обзору модели проблемно-ориентированной облачной вычислительной системы. В заключении изложены выводы и направления дальнейшего развития работы.

2. Модель предметной области

Особенностью распределенных проблемно-ориентированных вычислительных сред является то, что они обеспечивают решение заданий в рамках конкретной предметной области. Такие вычислительные задания часто могут быть представлены в виде ориентированного ациклического графа, узлами которого являются взаимосвязанные вычислительные задачи, а дуги соответствуют потокам данных, передаваемых между отдельными задачами. При этом, в рамках предметной области, набор задач, из которых строятся задания, является предопределенным. Задачи могут быть сгруппированы в конечное множество классов. Класс задач представляет собой множество задач, имеющих одну и ту же семантику, а также одинаковые наборы входных параметров и выходных данных.

С одной стороны, это накладывает ограничения на классы задач, которые могут быть решены в рамках ПВС. С другой стороны, такое ограничение позволяет использовать информацию о предметной области для прогнозирования вычислительных характеристик задач при планировании и распределении заданий, увеличивая эффективность использования доступных вычислительных ресурсов.

Дадим определение предметной области посредством на основе понятия информационного объекта.

Пусть *множество базовых информационных объектов* \mathcal{B} – это счетное множество объектов, которые в рамках соответствующей предметной области рассматриваются как атомарные (т.е. не имеющие составных частей), и разбиваются на k множеств *непересекающихся базовых классов* B_i :

$$\mathcal{B} = \bigcup_{i=1}^k B_i, (\forall i \neq j) B_i \cap B_j = \emptyset.$$

Будем считать, что мы можем определить размер любого базового информационного объекта множества \mathcal{B} (в байтах). Обозначим размер базового информационного объекта b как $|b|$:

$$|b|: b \rightarrow \mathbb{N}.$$

Определим *класс информационных объектов* \mathcal{C} следующим образом:

- 1) базовый класс информационных объектов будет являться классом информационных объектов:

$$\mathcal{C} = (B), \forall B \in \mathcal{B};$$

- 2) упорядоченный набор базовых классов информационных объектов будет являться классом информационных объектов:

$$\mathcal{C} = (B_1, \dots, B_m), (\forall l \in \{1, \dots, m\}) B_l \in \mathcal{B};$$

- 3) упорядоченный набор ранее определенных классов информационных объектов так же будет являться классом информационных объектов:

$$\mathcal{C} = (C_1, \dots, C_u).$$

Таким образом, информационный объект I класса $\mathcal{C} = (C_1, \dots, C_u)$ представляет собой упорядоченный набор конечного числа информационных объектов

$$I = (I_1, \dots, I_u),$$

где I_i – это:

- 1) (b) , где $b \in B$, если $C_i = (B)$;
- 2) (b_1, \dots, b_m) , где $(\forall l \in \{1, \dots, m\}) b_l \in B_l$, если $C_i = (B_1, \dots, B_m)$;
- 3) $(I_{i_1}, \dots, I_{i_y})$, где $(\forall l \in \{1, \dots, y\}) I_l$ – это информационный объект класса C_{i_l} , если $C_i = (C_{i_1}, \dots, C_{i_y})$.

Будем говорить, что базовый информационный объект b *входит в информационный объект* I ($b \in I$), если выполняется одно из следующих условий:

- 1) $I = (b)$;
- 2) $I = (b_1, \dots, b_m)$ и $b \in (b_1, \dots, b_2)$;
- 3) $I = (I_1, \dots, I_n)$ и $\exists I_i | b \in I_i$.

Определим *множество классов информационных объектов* \mathcal{C} как конечное множество всех определенных в рамках предметной области классов информационных объектов:

$$\mathcal{C} = \bigcup_{i=1}^{|\mathcal{C}|} C_i.$$

Определим *набор функций* \mathcal{F} над множеством \mathcal{C} :

$$\mathcal{F} = \bigcup_{i=1}^{|\mathcal{F}|} f_i, \text{ где } (\forall i \in \{1, \dots, |\mathcal{F}|\}) f_i: \mathcal{C}_i^{in} \rightarrow \mathcal{C}_i^{out}$$

Каждая функция $f \in \mathcal{F}$ на вход получает n информационных объектов $\mathcal{J}^{in} = (J_1^{in}, \dots, J_n^{in})$ соответствующих классов $\mathcal{C}^{in} = (C_1, \dots, C_n)$. Результатом работы функции являются m новых информационных объектов $\mathcal{J}^{out} = (J_1^{out}, \dots, J_m^{out})$ соответствующих классов $\mathcal{C}^{out} = (C'_1, \dots, C'_m)$.

Таким образом, определим предметную область \mathfrak{F} как упорядоченную тройку:

$$\mathfrak{F} = \langle \mathcal{B}, \mathcal{C}, \mathcal{F} \rangle,$$

где \mathcal{B} - множество базовых информационных объектов, \mathcal{C} - множество классов информационных объектов, \mathcal{F} - множество функций, определенных над \mathcal{C} .

3. Модель проблемно-ориентированной облачной вычислительной среды

Определим проблемно-ориентированную облачную вычислительную среду как четверку:

$$\mathfrak{C}(\mathfrak{N}, \mathfrak{E}, \mathfrak{M}, \mathfrak{F}),$$

где \mathfrak{N} - множество узлов вычислительной системы; \mathfrak{E} - множество сетевых соединений, связывающих вычислительные узлы; \mathfrak{M} - множество образов виртуальных вычислительных машин, доступных для развертывания на узлы из множества \mathfrak{N} , \mathfrak{F} - базовая предметная область.

Вычислительным узлом $n \in \mathfrak{N}$ назовем вычислительную систему с общей памятью, представленную тройкой:

$$(P_n, m_n, \Pi_n),$$

где P_n - это упорядоченное множество вычислительных ядер узла; m_n - это объем оперативной памяти, доступной на вычислительном узле; Π_n - это вектор характеристик производительности вычислительного узла.

Образом виртуальной машины $m \in \mathfrak{M}$ назовем тройку:

$$(P_m, m_m, \Pi_m),$$

где P_m - это упорядоченное множество вычислительных ядер, выделенных виртуальной машине; m_m - это объем оперативной памяти, выделенный виртуальной машине; Π_m - это вектор характеристик производительности виртуальной машины.

Определим характеристику производительности вычислительной машины как отображение:

$$\pi: m \rightarrow \mathbb{Z}_{>0},$$

где m - образ это виртуальной машины (либо вычислительный узел) существующий в вычислительной системе \mathfrak{C} .

Примерами характеристик производительности могут служить числовые характеристики машины, результаты синтетических тестов (Dhrystone [9], SuperPI [10], LINPACK [11], LAPACK [12] и др.) или результаты тестового выполнения конкретных классов функций с заранее определенными наборами входных данных.

Очевидно, что для качественного прогнозирования параметров выполнения задач на заданных машинах, нам необходимо учитывать максимально возможное количество характеристик производительности, включая такие характеристики как: количество доступных процессоров; частота процессора; скорость обмена данными с жестким диском; характеристика машины по LINPACK и др. Таким образом, определим вектор характеристик производительности виртуальных машин, развернутых в облачной вычислительной системе \mathfrak{C} :

$$\Pi = [\pi_1, \pi_2 \dots \pi_r].$$

Каждой машине $m \in \mathfrak{M}$ облачной вычислительной системы \mathfrak{C} сопоставим вектор характеристик производительности, отражающий значения производительности вычислительной машины:

$$\Pi: \mathfrak{M} \rightarrow \mathbb{Z}_{>0}^r.$$

В дальнейшем будем считать, что в рамках предоставления вычислительных ресурсов, каждой задаче выделяется одна либо несколько виртуальных машин. Прямого доступа к узлам вычислительной системы не обеспечивается. Особенностью проблемно-ориентированной облачной вычислительной среды является то, что она использует информацию об особенностях классов решаемых задач при планировании и распределении вычислительных ресурсов. Будем требовать, чтобы в рамках проблемно-ориентированной вычислительной среды, для каждого класса задач были определены следующие функции для прогноза процесса выполнения задачи в зависимости от значений входных параметров:

- 1) функция оценки объема выходных данных при определенных входных параметрах;
- 2) функция оценки времени выполнения задачи при определенных входных параметрах на машине с указанным вектором характеристик производительности.

Таким образом, для каждой функции $f \in \mathcal{F}$ из предметной области \mathfrak{F} , выполняющейся в проблемно-ориентированной среде \mathfrak{C} , определим следующий набор операторов:

- 1) оператор ожидаемого выхода $v(f, \mathcal{J}^{in})$ - это оператор, возвращающий ожидаемый общий размер в байтах всех выходных информационных объектов \mathcal{J}^{out} ;

$$v(f, \mathcal{J}^{in}) = |\mathcal{J}^{out}| = \sum_{\forall I \in \mathcal{J}^{out}} \sum_{\forall b \in I} |b|.$$

- 2) оператор ожидаемого времени выполнения функции $\tau(f, \mathcal{J}^{in}, \Pi)$, возвращающий оценочное время выполнения (в секундах) функции f при заданном множестве входных информационных объектов \mathcal{J}^{in} на машине, с вектором характеристик производительности Π :

$$\tau: (f, \mathcal{J}^{in}, \Pi) \rightarrow \mathbb{N}.$$

Время выполнения функции $f: \mathcal{C}^{in} \rightarrow \mathcal{C}^{out}$ на конкретной машине с вектором характеристик производительности Π можно представить в виде оператора, зависящего от вектора входных информационных объектов \mathcal{J}^{in} . К сожалению, невозможно оценить время выполнения функции с идеальной точностью, т.к. вычислительная работа подготовки набора выходных информационных объектов \mathcal{J}^{out} может косвенно зависеть от множества факторов, которые наша модель учесть не может (возможные фоновые процессы, качество предсказания ветвления конкретной версии процессора, объем занятого кэша и др.). Для компенсации данной ошибки, оценку времени выполнения функции можно смоделировать в виде случайной величины:

$$\chi(f, \Pi, \mathcal{J}^{in}) = \tau(f, \Pi, \mathcal{J}^{in}) + \alpha,$$

где $\tau(f, \Pi, \mathcal{J}^{in})$ – детерминированная функция, представляющая зависимость времени выполнения функции f на машине с вектором характеристик производительности Π от вектора входных информационных объектов \mathcal{J}^{in} , α – это стохастическая величина с нулевым математическим ожиданием ($M[\alpha] = 0$), представляющая факторы, не входящие в разрабатываемую модель.

Таким образом, для оценки времени выполнения задач необходимо обеспечить сбор и хранение статистики запусков по всем классам задач. После каждого запуска задачи в базе данных сохраняется следующая информация: значения параметров запуска, вектор характеристик производительности вычислительной машины, включая количество выделенных процессорных ядер и объем выделенной оперативной памяти, время выполнения и объем сгенерированных выходных данных.

4. Заключение

В данной статье предложены модели предметной области и проблемно-ориентированной облачной вычислительной среды. Представленные модели поддерживают возможность прогнозирования характеристик выполнения вычислительных задач в распределенных вычислительных средах (таких как время выполнения и объем выходных данных). Данные оценки будут использоваться для более эффективного планирования ресурсов распределенных вычислительных сред.

В рамках развития данного исследования, планируется выявить соответствующие классы и построить для них соответствующие оценочные функции для прогнозирования времени выполнения задач и объема выходных данных при планировании многовариантных расчетов в облачных вычислительных системах.

Список литературы

- [1]. Glotzer S.C. International assessment of research and development in simulation-based engineering and science. Imperial College Press, 2011. 312 p. doi: 10.1142/9781848166981.
- [2]. Reed D. et al. Computational Science: Ensuring America's Competitiveness. United States. President's Information Technology Advisory Committee. National Coordination Office for Information Technology Research & Development, 2005. 104 p.
- [3]. Folino G. et al. A grid portal for solving geoscience problems using distributed knowledge discovery services. Future Generation Computer Systems, 26(1), 2010. P. 87–96. doi: 10.1016/j.future.2009.08.002.
- [4]. Walker D.W. et al. The software architecture of a distributed problem-solving environment. Concurrency: Practice and Experience, 12(15), 2000. P. 1455–1480.
- [5]. Радченко Г.И. Распределенные виртуальные испытательные стенды: использование систем инженерного проектирования и анализа в распределенных вычислительных средах. Вестник ЮУрГУ. Серия “Математическое моделирование и программирование”, том 10, № 37(254), 2011 г. стр. 108–121.
- [6]. Deelman E. et al. Workflows and e-Science: An overview of workflow system features and capabilities. Future Generation Computer Systems, 25(5), 2009. P. 528–540.
- [7]. Bil C. Concurrent Engineering in the 21st Century. Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges, 2015. P. 421–454. doi: 10.1007/978-3-319-13776-6.
- [8]. Шамакина А.В., Соколинский Л.Б. Исследование алгоритма планирования POS для проблемно-ориентированных вычислительных сред. Параллельные вычислительные технологии труды международной научной конференции (31 марта - 2 апреля 2015 г., г. Екатеринбург), 2015. стр. 488–493.
- [9]. Weicker R.P. Dhrystone: a synthetic systems programming benchmark. Communications of the ACM, 27(10), 1984. P. 1013–1030. doi: 10.1145/358274.358283.
- [10]. WPrime Systems. Super PI. 2013. URL: <http://www.superpi.net/> (дата обращения: 14.11.2015).
- [11]. Dongarra J.J., Luszczek P., Petite A. The LINPACK benchmark: Past, present and future. Concurrency and Computation: Practice and Experience. 15(9), 2003. P. 803–820. doi: 10.1002/cpe.728.
- [12]. Demmel J., Dongarra J., Parlett B. Prospectus for the next LAPACK and ScaLAPACK libraries // PARA'06 Proceedings of the 8th international conference on Applied parallel computing: state of the art in scientific computing, 2006. P. 11–23. doi: 10.1007/978-3-540-75755-9.

Model of Problem-Oriented Cloud Computing Environment

G. Radchenko <gleb.radchenko@susu.ru>
South Ural State University,
454080, Russian Federation, Chelyabinsk, Lenina pr-kt, 76

Abstract. For efficient use of high-performance computing resources while implementing Computational Science methods for the study of physical, biological and social problems one can use problem-oriented distributed computing environments approach. They provide users with transparent access to the solution of specific classes of applications based on the available computing resources. To increase the effectiveness of such environments, one must use problem-oriented planning methods, which use the information about the subject area for predicting computing problems performance for optimal tasks planning and allocation. In the article the models of the subject area and problem-oriented cloud computing environment, focused on supporting the development of new problem-oriented scheduling algorithms are presented. Subject area P is defined as an ordered triple, consisting of a set of basic information objects B , the set of information object classes C and a set of functions defined over C . The task-oriented cloud computing environment can be defined as an ordered quadruple consisting of the set of nodes of a computer system N ; a set of network connections E ; a set of virtual machines images M , the basic subject area P . It should be required that within a problem-oriented computing environment the following functions for the prediction of the task execution, depending on the values of the input parameters for each class of problems were identified: the estimation of the amount of output data when a certain set of input parameters is given; the evaluation function of task execution time, given certain input parameters on the machine with the specified performance characteristics vector. Since it is impossible to estimate the time of the task execution with a perfect accuracy, task runtime evaluation should be modeled as a random variable. The provided model allows tasks execution time and output parameters volume evaluation through the collection, storage and analysis of statistics for all problems, executed in the environment.

Keywords: cloud computing; computing environment model; distributed computing system; problem-oriented computing environment.

DOI: 10.15514/ISPRAS-2015-27(6)-17

For citation: Radchenko G. Model of Problem-Oriented Cloud Computing Environment. *Trudy ISP RAN/Proc. ISP RAS*, vol. 27, issue 6, 2015, pp. 275-284 (in Russian). DOI: 10.15514/ISPRAS-2015-27(6)-17

References

[1]. Glotzer S.C. International assessment of research and development in simulation-based engineering and science. Imperial College Press, 2011. 312 p. doi: 10.1142/9781848166981.

- [2]. Reed D. et al. Computational Science: Ensuring America's Competitiveness. United States. President's Information Technology Advisory Committee. National Coordination Office for Information Technology Research & Development, 2005. 104 p.
- [3]. Folino G. et al. A grid portal for solving geoscience problems using distributed knowledge discovery services. *Future Generation Computer Systems*, 26(1), 2010. P. 87–96. doi: 10.1016/j.future.2009.08.002.
- [4]. Walker D.W. et al. The software architecture of a distributed problem-solving environment. *Concurrency: Practice and Experience*, 12(15), 2000. P. 1455–1480.
- [5]. Radchenko G.I. Raspredelelynye virtual'nye ispytel'nye stendy: ispol'zovanie sistem inzhenernogo proektirovaniya i analiza v raspredelelynykh vychislitel'nykh sredakh. [Distributed virtual test-beds: usage of CAE systems in distributed computing environments] *Vestnik JuUrGU. Seriya "Matematicheskoe modelirovanie i programmirovaniye"* [SUSU Bulletin: The "Mathematical Modeling and Programming" series], vol 10, No 37(254), 2011. pp. 108–121. (In Russian).
- [6]. Deelman E. et al. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5), 2009. P. 528–540.
- [7]. Bil C. Concurrent Engineering in the 21st Century. *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, 2015. P. 421–454. doi: 10.1007/978-3-319-13776-6.
- [8]. Shamakina A.V., Sokolinsky L.B. Issledovanie algoritma planirovaniya POS dlja problemno-orientirovannykh vychislitel'nykh sred [Study of the the POS scheduling algorithm for problem-oriented computing environments] *Parallel'nye vychislitel'nye tekhnologii trudy mezhdunarodnoj nauchnoj konferencii (31 marta - 2 aprilja 2015 g., g. Ekaterinburg)* [Parallel Computing Technologies: proceedings of the International Scientific Conference (31 March - 2 April 2015 Ekaterinburg)], 2015. pp. 488–493. (In Russian).
- [9]. Weicker R.P. Dhrystone: a synthetic systems programming benchmark. *Communications of the ACM*, 27(10), 1984. P. 1013–1030. doi: 10.1145/358274.358283.
- [10]. WPrime Systems. Super PI. 2013. URL: <http://www.superpi.net/> (дата обращения: 14.11.2015).
- [11]. Dongarra J.J., Luszczek P., Petite A. The LINPACK benchmark: Past, present and future. *Concurrency and Computation: Practice and Experience*. 15(9), 2003. P. 803–820. doi: 10.1002/cpe.728.
- [12]. Demmel J., Dongarra J., Parlett B. Prospectus for the next LAPACK and ScaLAPACK libraries // *PARA'06 Proceedings of the 8th international conference on Applied parallel computing: state of the art in scientific computing*, 2006. P. 11–23. doi: 10.1007/978-3-540-75755-9.