

вершенствованием правовой базы потребует решения вопросов как производительности и масштабируемости, так обеспечения информационной безопасности. В качестве варианта решения может быть предложено использование доверенных, сертифицированных аппаратных блокчейн-клиентов и создание, по образцу операторов фискальных данных, сети независимых доверенных блокчейн-центров.

1. *Авдеев, М.* Блокчейн-технологии в госуправлении. Мировой опыт [Электронный ресурс] / М. Авдеев. – Режим доступа: <http://www.forbes.ru/tehnologii/343203-blokcheyn-tehnologii-v-gosupravlenii-mirovoy-opyt> (дата обращения: 29.04.2018).
2. Децентрализованная сеть обмена и хранения информации «Мастерчейн» Версия 1.1 Whitepaper [Электронный ресурс] // АФТ. – Москва, 2017. – Режим доступа: http://fintechru.org/documents/Masterchain_whitepaper_11_08.pdf, свободный (дата обращения: 29.04.2018).
3. *Архипов, А.* Что умеет Мастерчейн? [Электронный ресурс] / А. Архипов // Инвест-Форсайт. – Июль 5, 2017. – Режим доступа: <https://www.if24.ru/chto-umeet-mastercheyn> (дата обращения: 29.04.2018).
4. Ассоциация «Национальный координационный центр обработки транзакций с правами и объектами интеллектуальной собственности» [Электронный ресурс]. – Режим доступа: <https://ipchain.ru/possibilities/assoc> (дата обращения: 29.04.2018).
5. ThreeFold token [Электронный ресурс]. – Режим доступа: <https://threefoldtoken.com> (дата обращения: 29.04.2018).

***Я.С. Коровин, С.Г. Капустян, М.В. Хисамутдинов,
А.И. Каляев, Д.Я. Иванов***

АЛГОРИТМ ФУНКЦИОНИРОВАНИЯ АГЕНТА ОБЛАЧНОГО СЕРВИСА СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ*

*НИИ многопроцессорных вычислительных систем ЮФУ, г. Таганрог,
donat.ivanov@gmail.com*

При создании систем поддержки принятия решений (СППР), применяемых на крупных мехатронных комплексах и объектах, необходимо обеспечивать обработку больших объемов поступающей информации и своевременный доступ к необходимой информации множеству пользователей, зачастую разнесенных территориально. Одним из актуальных подходов к построению таких СППР является применение облачных технологий [1, 2]. В данной работе рассматривается работа облачного сервиса [3, 4] на основе мультиагентного диспетчера [5–7].

* Работа выполнена в рамках федеральной целевой программы Минобрнауки РФ (соглашение № 14.575.21.0152 от 26.09.2017).

Постановка задачи диспетчирования облачной вычислительной среды

Пусть облачная вычислительная среда (ОВС) должна решать некоторое множество (поток) различных крупномасштабных задач $Z = \langle Z_1, Z_2, \dots, Z_M \rangle$, которые могут поступать от пользователей в случайные моменты времени. При этом под крупномасштабной задачей будем понимать задачу, состоящую из некоторого множества информационно зависимых (взаимосвязанных) подзадач, каждая из которых имеет значительную вычислительную трудоемкость. Формально каждая такая крупномасштабная задача $Z_l \in Z$ может быть представлена в виде некоторого графа $G_l(Q_l, X_l)$, приведенного на рис. 1.

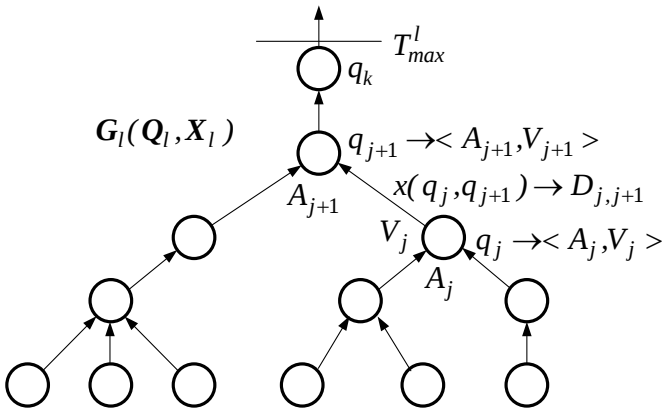


Рис. 1. Граф $G_l(Q_l, X_l)$ задачи

Вершины $q_j \in Q_l$ этого графа соответствуют некоторым подзадачам A_j , принадлежащим множеству подзадач $A = \langle A_1, A_2, \dots, A_c \rangle$, а дуги $x(q_j, q_{j+1}) \in X_l$ определяют информационные взаимосвязи между подзадачами (т.е. если две вершины q_j и q_{j+1} соединены дугой $x(q_j, q_{j+1})$, то это означает, что данные, полученные в результате решения подзадачи A_j , приписанной вершине q_j , необходимы для выполнения подзадачи A_{j+1} , приписанной вершине q_{j+1}). Будем считать, что каждой вершине $q_j \in Q_l$ приписан тип решаемой подзадачи $A_j \in A$, а также ее вычислительная трудоемкость V_j , оцениваемая числом эле-

ментарных вычислительных операций, выполняемых при ее решении, а дуге $x(q_j, q_{j+1}) \in X_l$ приписан объем данных $D_{j,j+1}$, передаваемых от подзадачи A_j , приписанной вершине q_j , подзадаче A_{j+1} , приписанной вершине q_{j+1} (см. рис. 1). Кроме того, положим, что пользователь устанавливает и приписывает конечной вершине q_k графа $G_l(Q_l, X_l)$ момент времени $T_{\max}^l$, к которому он желает получить результат решения своей задачи Z_l , а также определяет количество премиальных баллов (бонусов) O , которое он готов заплатить за это.

Пусть в состав ОВС входит множество вычислительных ресурсов $R = \langle R_1, R_2, \dots, R_N \rangle$. При этом, под вычислительным ресурсом будем понимать некоторое вычислительное устройство (ПК, сервер, многопроцессорная вычислительная система и т.п.), подключенное к облачной инфраструктуре. Будем считать, что каждый ресурс $R_i \in R$ может решать некоторый набор (подмножество) типов подзадач $A_i = \langle A_1^i, A_2^i, \dots, A_L^i \rangle \subseteq A$ ($i = 1, 2, \dots, N$), причем реальная производительность ресурса R_i при решении подзадачи $A_j^i \in A_i$ составляет $S_j^i = F(A_j^i)$ ($j = 1, 2, \dots, L$) операций в секунду. Кроме того, будем считать, что пропускная способность канала связи ресурса $R_i \in R$ с облачной инфраструктурой составляет Y_i байт/сек.

Целью работы диспетчера ОВС является такое распределение подзадач потока задач Z по вычислительным ресурсам $R_1, R_2, \dots, R_N$, при котором все задачи $Z_l \in Z$ будут выполнены к установленным пользователями моментам времени $T_{\max}^l$.

Алгоритм функционирования агента

- 1) агент свободного ресурса R_p опрашивает доску объявлений (ДО);
- 2) при обнаружении на ДО задачи Z_l агент R_p считывает ее дескриптор и анализирует граф задачи $G_l^j(Q_l^j, X_l^j)$. Если $G_l^j(Q_l^j, X_l^j) = \emptyset$, то переход к п.1, иначе
- 3) если $T_{\text{тек}} \geq T_{\max}^l$, где $T_{\text{тек}}$ – текущий момент времени, то переход к п. 22, иначе
- 4) в графе $G_l^j(Q_l^j, X_l^j)$ выделяется подграф $G_l^{jp}(Q_l^{jp}, X_l^{jp})$, вершинам которого приписаны подзадачи множества A_p , выполняемые ресурсом R_p ;

5) если $G_l^{jp}(Q_l^{jp}, X_l^{jp}) = \emptyset$, то перейти к п. 1, иначе

6) $i = 1$; $G_i(Q_i, X_i) = G_l^{jp}(Q_l^{jp}, X_l^{jp})$;

7) агент R_p выделяет в графе $G_i(Q_i, X_i)$ наиболее длинную нить $H_i = \langle q_1^i, q_2^i, \dots, q_k^i \rangle$, конечной вершине которой приписано требуемое время исполнения T_{k+1}^i (в момент размещения задания Z_i на ДО требуемое время $T_{k+1}^i = T_{\max}^l$ приписано только конечной вершине q_k графа $G_l(Q_l, X_l)$). Если таковой нити в графе $G_i(Q_i, X_i)$ нет, то перейти к п. 1, иначе

8) агент R_p определяет допустимый момент времени T_1^i , когда необходимо начать выполнение нити $H_i = \langle q_1^i, q_2^i, \dots, q_k^i \rangle$, чтобы успеть завершить ее исполнение к моменту T_{k+1}^i , как

$$T_1^i = T_{k+1}^i - \left(\sum_{j=1}^k \frac{V_j^i}{S_j^p} + \frac{D_{k,k+1}^i}{Y_p} \right); \quad (1)$$

9) если $T_1^i \geq T_{\text{тек}}$, где $T_{\text{тек}}$ – текущий момент времени, то переход к п. 13, иначе

10) нить H_i исключается из графа $G_i(Q_i, X_i)$, т.е.

$$G_{i+1}(Q_{i+1}, X_{i+1}) = G_i(Q_i, X_i) / H_i;$$

11) если $G_{i+1}(Q_{i+1}, X_{i+1}) = \emptyset$, то перейти к п. 1, иначе

12) $i = i + 1$, перейти к п. 7;

13) агент R_p принимает на себя исполнение нити H_i для чего:

– считает с ДО последовательность подзадач, приписанных вершинам нити H_i , требуемое время ее исполнения T_{k+1}^i и идентификатор ресурса R_c , которому необходимо передать результаты ее решения;

– модифицирует дескриптор задач Z_l на ДО: в список участников сообщества R_l записывается его идентификатор; из графа $G_l^j(Q_l^j, X_l^j)$ исключаются вершины нити H_i , в результате чего формируется новый граф $G_l^{j+1}(Q_l^{j+1}, X_l^{j+1}) = G_l^j(Q_l^j, X_l^j) / H_i$;

– вершинам q_f^{i+1} модифицированного графа $G_l^{j+1}(Q_l^{j+1}, X_l^{j+1})$, инцидентным вершинам q_b^i нити $H_i = \langle q_1^i, q_2^i, \dots, q_k^i \rangle$ приписываются

идентификаторы ресурсов R_p , которым необходимо передать результаты исполнения подзадачи, соответствующей вершине q_f^{i+1} , а также требуемое время их исполнения, определяемое как

$$T_b^i = T_{k+1}^i - \left(\sum_{m=b}^k \frac{V_m^i}{S_m^p} + \frac{D_{k,k+1}^i}{Y_p} \right); \quad (2)$$

14) агент R_p переходит к исполнению последовательности подзадач, приписанных вершинам нити $\mathbf{H}_i = \langle q_1^i, q_2^i, \dots, q_k^i \rangle$; $d = 1$.

15) если $T_d^i < T_{\text{тек}}$, где $T_d^i = T_{k+1}^i - \left(\sum_{m=d}^k \frac{V_m^i}{S_m^p} + \frac{D_{k,k+1}^i}{Y_p} \right)$ – требуемое время начала выполнения подзадачи A_d , приписанной вершине $q_d^i \in \mathbf{H}_i$, то перейти к п. 17, иначе

16) агент R_p проверяет наличие исходных данных, необходимых для выполнения подзадачи A_d . Если исходные еще не поступили, то перейти к п. 15, иначе перейти к п. 18;

17) агент R_p отправляет результаты решения подзадачи, приписанной вершине $q_{d-1}^i \in \mathbf{H}_i$, на ДО и осуществляет модификацию дескриптора задачи Z_l : исключает свой идентификатор из списка сообщества $\mathbf{R}_l$; добавляет неисполненные вершины $q_d^i, q_{d+1}^i, \dots, q_k^i$ нити $\mathbf{H}_i$ в граф задачи; приписывает крайней вершине этой нити момент времени T_{k+1}^i , к которому она должна быть исполнена, а также идентификатор ресурса R_c , которому необходимо передать результаты ее решения. Переход к п. 1;

18) агент R_p выполняет операцию A_d , приписанную вершине $q_d^i \in \mathbf{H}_i$ с помощью «своего» ресурса;

19) если агенту R_p поступило сообщение о прекращении выполнения задачи Z_l , то переход к п. 1, иначе

20) $d = d + 1$, если $d \leq k$, то переход к п. 15, иначе

21) агент R_p сообщает на ДО об успешном завершении решения нити $\mathbf{H}_i$ задачи Z_l . При этом результаты решения передаются ресурсу

R_c , идентификатор которого приписан конечной вершине данной нити, идентификатор агента R_p исключается из списка членов сообщества R_l по решению данной задачи, а ему начисляются премиальные баллы $O_p = O \cdot V^p / V$, где V^p – суммарная вычислительная трудоемкость подзадач, выполненных агентом R_p ; V – суммарная трудоемкость всей задачи Z_l . Переход к п. 1;

22) Задача Z_l не может быть решена к установленному пользователем моменту времени. Дескриптор задачи Z_l удаляется с ДО, пользователю направляется сообщение о невозможности решения его задачи к требуемому моменту времени, а всем агентам, идентификаторы которых записаны в списке участников сообщества R_l по выполнению задачи Z_l , передается сообщение о прекращении ее исполнения. Переход к п. 1.

Исследования предложенного алгоритма, проведенные с помощью программной модели ОВС, показали, что коэффициент гарантированности выполнения пользовательской задачи превышает 90 %, а отношение времени, затраченного ресурсами ОВС на выполнение пользовательских задач, к общему времени их работы в системе превышает 70 %, что позволяет использовать предложенный алгоритм при построении облачных сервисов СППР.

1. *Armbrust, M. A view of cloud computing / M. Armbrust et al. // Commun. ACM. – 2010. – Vol. 53, No 4. – P. 50–58.*
2. *Aazam, M. E-HAMC: Leveraging Fog computing for emergency alert service / M. Aazam, E.N. Huh // 2015 IEEE International Conference on Pervasive Computing and Communication Workshops. – 2015. – P. 518–523.*
3. *Kalyaev, A.I. Method of multiagent scheduling of resources in cloud computing environments / A.I. Kalyaev, I.A. Kalyaev // J. Comput. Syst. Sci. Int. – 2016. – Vol. 55, No 2. – P. 211–221.*
4. *Sharma, R. Cloud SaaS and model driven architecture / R. Sharma, M. Sood // Int. Conf. Adv. Comput. Commun. Technol. – 2011. – P. 978–981.*
5. *Каляев А.И., Каляев И.А. Метод мультиагентного диспетчирования ресурсов в облачных вычислительных средах [Текст] / А.И. Каляев, И.А. Каляев // Известия Российской академии наук. Теория и системы управления. – 2016. – № 2. – С. 51.*
6. *Каляев, И.А. Метод мультиагентного диспетчирования ресурсов в облачной вычислительной среде при решении потока задач [Текст] / И.А. Каляев // Вестник компьютерных и информационных технологий. – 2015. – С. 31–40.*
7. *Иванов, Д.Я. Принципы организации децентрализованных сетевых информационных систем [Текст] / Д.Я. Иванов, Э.В. Мельник // Вестник компьютерных и информационных технологий. – 2013. – № 4. – С. 25–30.*