

Оценка эффективности управления вычислениями диспетчером научных рабочих процессов

Михаил Воскобойников, Александр Феоктистов

Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия

Аннотация

Рассматривается проблема повышения эффективности управления вычислениями при выполнении научных приложений в гетерогенной распределенной вычислительной среде, схемы решения задач в которых реализуются научными рабочими процессами. Одним из ключевых факторов, влияющих на время выполнения рабочих процессов, является время, затрачиваемое на передачу данных между их модулями. В известных системах управления рабочими процессами передача данных между модулями зачастую осуществляется через центральный узел, что увеличивает время выполнения этих процессов. В работе предложены средства управления вычислениями с учетом структуры рабочего процесса и передачи данных напрямую между узлами. Разработан эвристический алгоритм назначения узлов для запуска модулей рабочего процесса на основе оценки времени их выполнения и размера данных, передаваемых между ними. Результаты экспериментов показывают существенное сокращение времени выполнения ряда известных научных рабочих процессов при использовании данного алгоритма относительно применения алгоритмов подобного назначения в системе HTCCondor.

Ключевые слова

Научные приложения, рабочие процессы, назначение узлов, передача данных

1. Введение

Управление вычислениями (планирование вычислений и назначение ресурсов) является чрезвычайно актуальной проблемой при проведении крупномасштабных экспериментов в гетерогенной распределенной вычислительной среде (ГРВС). Как правило, в современных научных приложениях план решения задачи реализуется научным рабочим процессом (НРП) – информационно-вычислительной структурой, определяющей бизнес-логику предметной области, расчетные данные, набор используемых прикладных модулей и вычислительные ресурсы, требуемые для проведения экспериментов. НРП может быть абстрактным или конкретным. Абстрактный НРП описывает схему решения задачи без привязки ее операций к конкретному программному обеспечению (ПО) и вычислительным ресурсам. В конкретном НРП операции схемы решения задачи связываются с определенным ПО, которое выполняется на заданных ресурсах.

Как правило, при построении НРП используют процедурную или непроцедурную постановку задачи. В рамках процедурной постановки задачи формируется последовательность операций НРП, а затем осуществляется информационное планирование входных и выходных параметров. При непроцедурной постановке задачи задаются исходные (входные) и целевые (выходные) параметры НРП. Далее на вычислительной модели предметной области (описании ее параметров, вычислительных операций, операций обработки данных, отношений между параметрами и операциями) осуществляется автоматическое планирование последовательности

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia

EMAIL: mikey1988@mail.ru (A. 1); agf@icc.ru (A. 2)

ORCID: 0000-0003-3034-4907 (A. 1); 0000-0002-9127-6162 (A. 2)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings

DOI: 10.47350/ICCS-DE.2025.09

операций НРП, необходимых для решения поставленной задачи. В системах управления НРП (СУ НРП) автоматическое планирование вычислений осуществляется одним из ее компонентов (например, планировщиком вычислений).

СУ НРП – это программное средство, позволяющее частично автоматизировать процессы создания и выполнения НРП, а также управления им в соответствии с его информационно-вычислительной структурой. В СУ НРП рабочий процесс зачастую задается в виде ориентированного ациклического графа (англ. Directed Acyclic Graph, DAG). При этом вершины DAG соответствуют операциям НРП, реализуемым в дальнейшем прикладными модулями или сервисами, а ребра – потокам данных между ними. На практике хорошо известны, поддерживаются, развиваются и широко используются следующие СУ НРП: UNICORE [1], Galaxy [2], Pegasus [3], DAGMan [4], Workflow-as-a-Service Cloud Platform [5], Apache Airflow [6], HyperFlow [7] и Orlando Tools [8].

Метапланировщик DAGMan осуществляет построение плана решения задачи на основе описания НРП в форме DAG. В качестве системы управления прохождением заданий (СУПЗ) на вычислительном кластере выступает HTCondor. Узлы кластера HTCondor подразделяются по следующим ролям: узел Submit, с которого происходит отправка заданий и исходных данных, узел Execute, предназначенный для выполнения заданий, и узел Master, координирующий работу всех узлов кластера. Перед запуском каждого прикладного модуля происходит перемещение его входных данных с узла Submit на узел Execute. После завершения выполнения прикладного модуля его выходные данные перемещаются обратно на узел Submit. Как правило, НРП имеют сложную структуру, состоящую из большого числа операций, экземпляры которых могут выполняться последовательно и параллельно. В связи с этим число пересылок данных в худшем случае равняется $n * 2m$, где n – число операций НРП, m – число вычислительных узлов. При больших объемах пересылаемых данных это может приводить к существенному увеличению времени выполнения НРП и вычислительного эксперимента в целом.

В работе предложены средства управления вычислениями с учетом структуры рабочего процесса и передачи данных напрямую между узлами, обеспечивающие сокращение время проведения экспериментов в сравнении с DAGMan и HTCondor.

2. Управление вычислениями диспетчером НРП

В состав инструментального комплекса FDE-SWFs [9] входит диспетчер НРП, который осуществляет управление вычислительными процессами и средой их выполнения. В качестве СУПЗ в диспетчере НРП используется HTCondor. Схема управления вычислениями представлена на рис. 1. В HPCondor используется централизованный механизм передачи данных. При запуске НРП на вычислительных узлах создаются временные папки, в которые перемещаются данные и модули. Задание выполняется от имени пользователя nobody. Данный пользователь не имеет разрешений на выполнение модулей вне созданной временной папки и команд на перемещение данных между вычислительными узлами. Организация прямого перемещения данных между вычислительными узлами и выполнения на них модулей включает следующие этапы:

- 1) задание пользователя для каждого слота HTCondor (слот представляет собой единицу представления ресурсов вычислительного узла) с разрешением на чтение/запись в папки вне временной папки, создаваемой в операционной системе (ОС) при выполнении задания;
- 2) планирование очередности выполнения модулей для обеспечения минимизации времени выполнения НРП с учетом перемещения данных между узлами;
- 3) генерация скриптов-оболочек, в которых прописываются команды для выполнения модулей и передачи данных между вычислительными узлами в соответствии с НРП;
- 4) формирование паспортов заданий, в которых в качестве исполняемых модулей указываются скрипты-оболочки и адреса вычислительных узлов.

На рис. 2 показана схема работы механизма передачи данных через центральный узел. Можно заметить, что при запуске и завершении выполнения каждого прикладного модуля НРП происходит передача входных и выходных данных между вычислительными узлами и

центральным узлом. На рис. 2 изображена схема работы механизма перемещения данных через центральный узел.

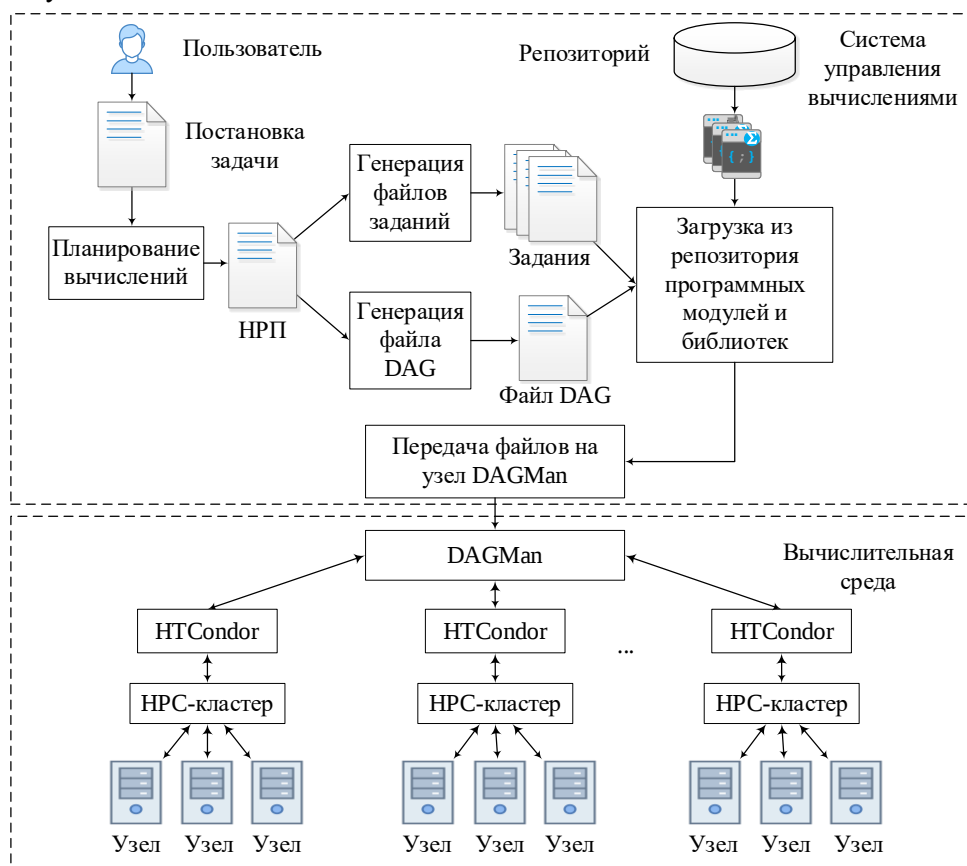


Рисунок 1: Схема взаимодействия диспетчера НРП с DAGMan и HTCondor

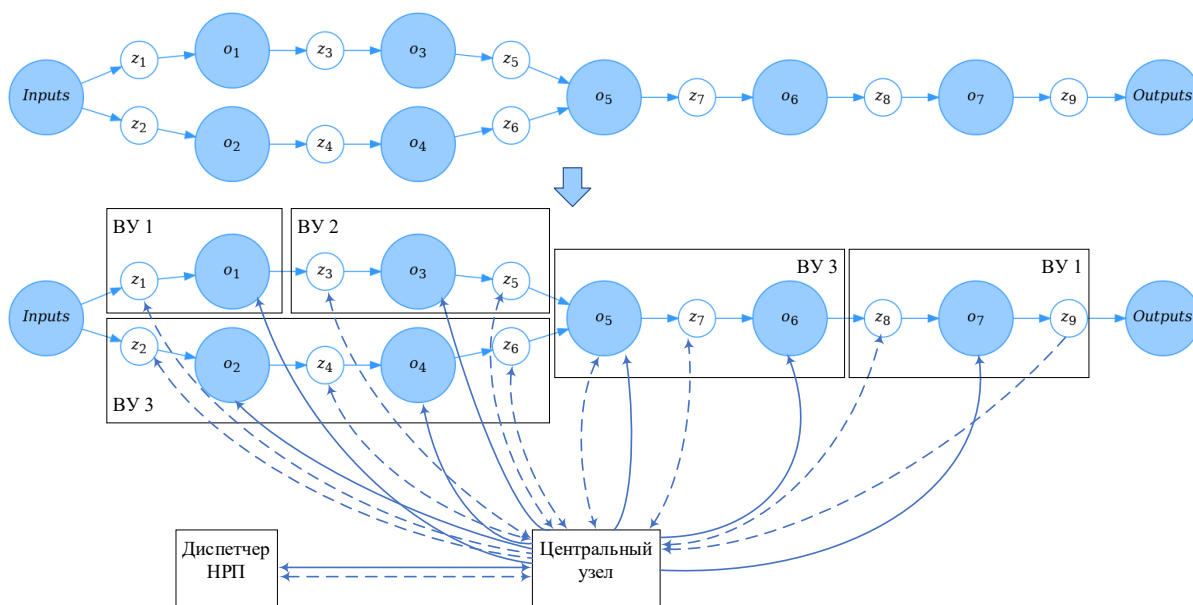


Рисунок 2: Механизм передачи данных через центральный узел

На рис. 3 приведена схема работы механизма передачи данных между вычислительными узлами на основе информации о структуре НРП.

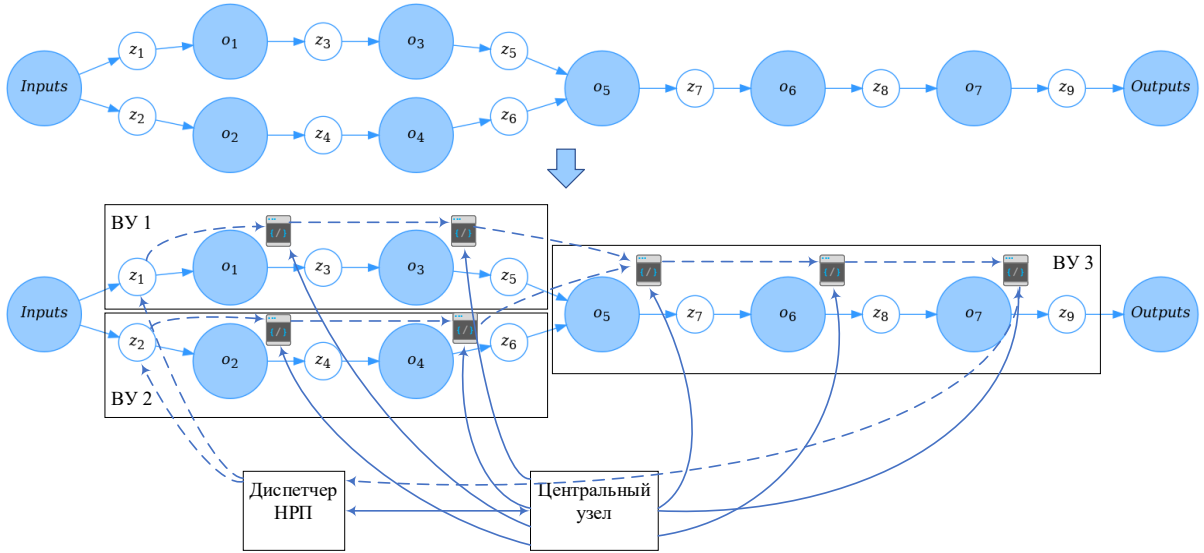


Рисунок 3: Механизм прямой передачи данных между вычислительными узлами

4. Постановка задачи

Пусть имеются m узлов ГРВС (включая центральный узел), на которых должны быть выполнены n операций НРП, где b – индекс основного узла. Введем следующие обозначения:

- t_{il} – время выполнения i -й операции на l -м узле, $i \in \overline{1, n}$, $l \in \overline{1, m}$;
- P – булева матрица размерности $n \times n$, элемент матрицы $p_{ij} = 1$ означает, что i -я операция зависит по входу от j -й операции (j -я операция предшествует i -й операции), $i, j \in \overline{1, n}$;
- D – матрица размерности $m \times m$, отражающая оценки размеров передаваемых данных между узлами, элемент матрицы $d_{lr} > 0$ показывает размер данных, передаваемых между l -м и r -м узлами, $r, l \in \overline{1, m}$;
- X – булева матрица размерности $m \times n$, элемент которой x_{li} показывает, что i -я операция выполняется на l -м узле;
- матрица W размерности $m \times m$ предоставляет информацию о пропускной способности интерконнекта между узлами, элемент матрицы $w_{lr} \geq 0$ демонстрирует пропускную способность интерконнекта между l -м и r -м узлами;
- $0 < \omega_{lr}(t) \leq 1$ – коэффициент снижения пропускной способности интерконнекта в момент времени t между l -м и r -м узлами;
- где τ_{ir} (τ_{jl}) – оценка времени, прошедшего с начала обработки схемы до завершения операции o_i (o_j) на r -м (l -м) узле;
- q_{ir} – оценка времени нахождения модуля, реализующего операцию o_i (в то время, когда все данные, необходимые для его выполнения, готовы) в очереди r -го узла.
- N_l – таблица операций, которые назначены на l -й узел.

Матрица X должна удовлетворять следующим условиям (1) и (2):

$$\bigvee_{i=1}^n \bigvee_{l=1}^{m-1} \bigvee_{k=l+1}^m (x_{li} \wedge x_{ki}) = 0, \quad (1)$$

$$\bigvee_{i=1}^n \bigwedge_{l=1}^m \bar{x}_{li}. \quad (2)$$

В случае выполнения НРП средствами СУПЗ HTCondor, где выбор и назначение узлов для запуска операций осуществляется главным менеджером этой системы, а обмен данными между узлами осуществляется через центральный узел, оценка $C(X)$ времени выполнения НРП в асинхронном режиме по готовности данных определяется следующим образом:

$$C(X) = \max_{i=1, n} \tau_{il}, \quad (3)$$

$$\tau_{il} = q_{il} + t_{il} + \max_{\forall j \in \overline{1, n}: p_{ij}=1, i \neq j, r \in \overline{1, m}} \left(\tau_{jr} + \frac{d_{rl}}{\omega_{rb}(t)w_{rb}} + \frac{d_{rl}}{\omega_{bl}(t)w_{bl}} \right). \quad (4)$$

Обмен данными между узлами через центральный узел существенно увеличивает общее время выполнения НРП. Поэтому в диспетчере НРП реализован дополнительный механизм передачи данных напрямую с узла выполнения предшествующей операции на узел выполнения последующей операции. В этом случае, если операции выполняются на одном узле, то размер передаваемых данных между ними считается равным нулю. Тогда оценка $Y(X)$ времени выполнения НРП в асинхронном режиме по готовности данных с помощью диспетчера НРП определяется следующим образом:

$$Y(X) = \max_{i=1, n} \tau_{il}, \quad (5)$$

$$\tau_{il} = q_{il} + t_{il} + \max_{\forall j \in \overline{1, n}: p_{ij}=1, i \neq j} \left(\tau_{jr} + \frac{s}{\omega_{lr}(t)w_{lr}} \right), \quad (6)$$

$$s = \begin{cases} d_{lr}, & \text{если } l \neq r, \\ 0 & \text{и противном случае.} \end{cases} \quad (7)$$

В (5)-(7) $Y(X)$ является целевой функцией, которую требуется минимизировать:

$$Y(X) \downarrow \min_{x_{li}}. \quad (8)$$

В общем случае задача (8) является NP-трудной задачей [10], оптимальное решение которой сложно получить за время, приемлемое для управления вычислениями в реальном времени. Поэтому в статье предлагается эвристический алгоритм решения этой задачи.

5. Алгоритм назначения операций на вычислительные узлы

Суть алгоритма **A** назначения операций на вычислительные узлы состоит в следующем: каждому узлу вычислительной модели назначаются такие операции из очереди операций, чтобы общее время выполнения НРП было минимальным. Для этого выполняется оценка времени выполнения операции на каждом узле с учетом оценок размера передаваемых данных между операциями и скоростью передачи данных между узлами ГРВС. При этом если на очередном шаге работы алгоритма одному и тому же узлу назначаются две и более операции, то выбирается та операция, которая предшествует большему числу операций НРП.

Алгоритм A. Назначение операций на вычислительные узлы

Inputs: $P, T, D, O, n, m, q, W, \omega$

Outputs: *nodes*

```

1   $t = 0$ 
2   $freeNodeList = \text{Array}()$ 
3  for  $k = \overline{1, m}$  step 1 do
4     $freeNodeList[k] = \text{True}$ 
5   $knownParams = \text{get\_known\_params}(O)$ 
6   $readyOperations = \text{get\_ready\_operations}(O)$ 
7   $subOperList = \text{get\_subsequent\_operations}(P)$ 
8   $nodeTime = \text{Array}()$ 
9  while  $\text{length}(readyOperations) > 0$  do
10    $operation = \text{getOperation}(readyOperations)$ 
11    $currNodes = \text{Array}()$ 
12    $times = \text{Array}()$ 
13   for  $l$  in  $freeNodeList$  do
14     for  $r$  in  $freeNodeList$  do
15       for  $i = \overline{1, n}$  step 1 do
16         if  $p[i] == 1$  and  $use[i] == 1$  then
17            $times.add(\text{calcExecTimes}(operation, T, D, W, \omega, nodeTime))$ 
18         end if
```

```

19     end do
20 end do
21 end do
22 currNodes = get_curr_nodes(times)
23 k = getNodeMinTimes(times)
24 if length(currNodes[k]) == 1 then
25     currOperation = currNodes[k]
26     nodeTime[k] = get_operation_time(currOperation, T, D)
27     readyOperation.remove(currOperation)
28     nodes.add(currOperation)
29     freeNodeList[k] = False
30 else if length(currNodes[k]) > 1 then
31     currOperation = get_suitable_operation(currNodes[k], subOperList)
32     nodeTime[k] = set_operation_to_node(currOperation, T, D, W,  $\omega$ )
33     nodes.add(currOperation)
34     readyOperation.remove(currOperation)
35     freeNodeList[k] = False
36 end if
37 do
38     countExecutedOper = get_executed_operation(nodes)
39     if countExecutedOper == n then
40         exit
41     t = getTimeShift(freeNodeList, nodeTime)
42     for k = 1, m step 1 do
43         if nodeTime[k] == t then
44             operation = get_operation_from_node(nodes[k])
45             freeNodeList[k] = True
46             knownParams = knownParams.union(operation)
47         end if
48     end do
49     readyOperations = get_ready_operations(O)
50 while length(readyOperations) == 0
51 end do

```

6. Вычислительный эксперимент

Для демонстрации сокращения времени выполнения вычислений в ГРВС с помощью разработанных средств были выбраны два известных НРП – Epigenomics и CyberShake [11], структура которых приведена на рис. 4.

В рамках эксперимента реализованы три схемы выполнения НРП под управлением HTCondor:

- 1) с централизованной передачей данных;
- 2) с помощью механизма прямой передачи данных;
- 3) с учетом структуры и использованием механизма прямой передачи данных.

Вычислительная среда включает четыре узла со следующими характеристиками: 1 CPU, 2 ГБ памяти, жесткий диск размером 30 ГБ, пропускная способности сети – 5 Гбит/с. Расписания времени выполнения модулей Epigenomics и CyberShake приведены соответственно на рис. 5 и 6. Очевидно, что при переходе от схемы 1 к схемам 2 и 3 расписание выполнения НРП становится все более плотным, задержки на передачу данных становятся короче. Диаграмма времени выполнения Epigenomics и CyberShake для трех схем дана на рис. 7.

В случае выполнения Epigenomics схемы 2 и 3 показали преимущество по сокращению времени вычислений в сравнении со схемой 1 более чем на 32% и 40% соответственно. Для

CyberShake это преимущество оказалось еще существеннее. Схемы 2 и 3 позволили сократить время вычислений в сравнении со схемой 1 более чем на 47% и 53% соответственно.

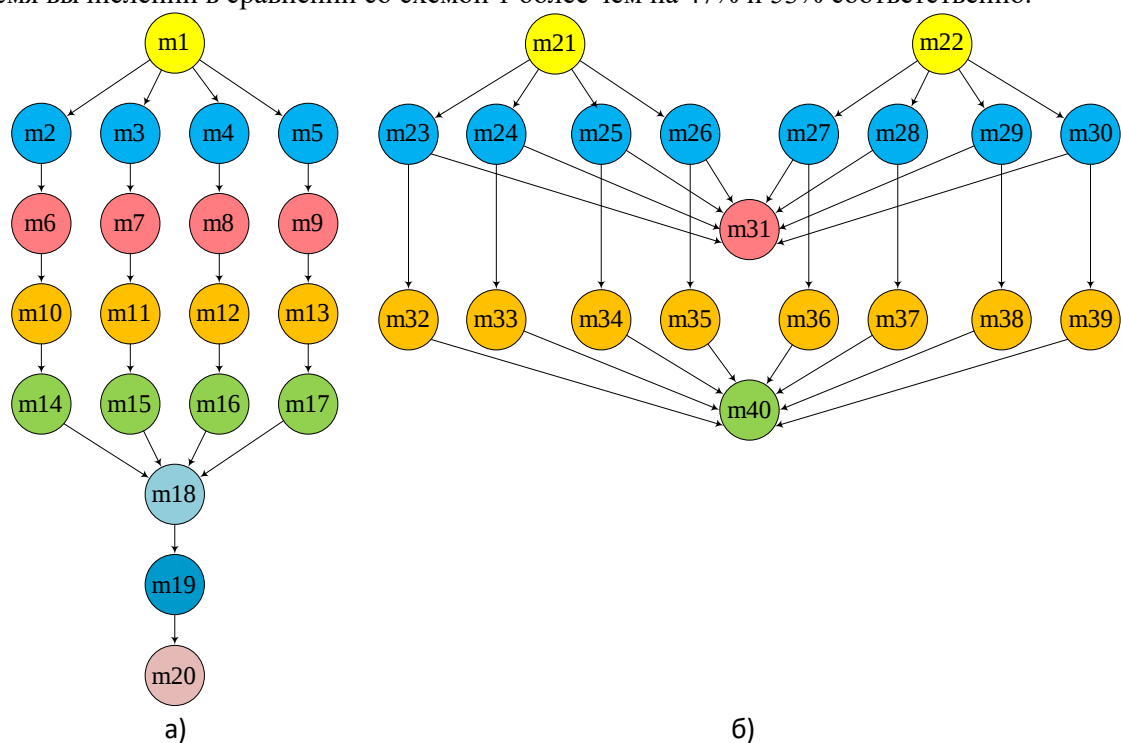


Рисунок 4: НРП: Epigenomics (а) и CyberShake (б)

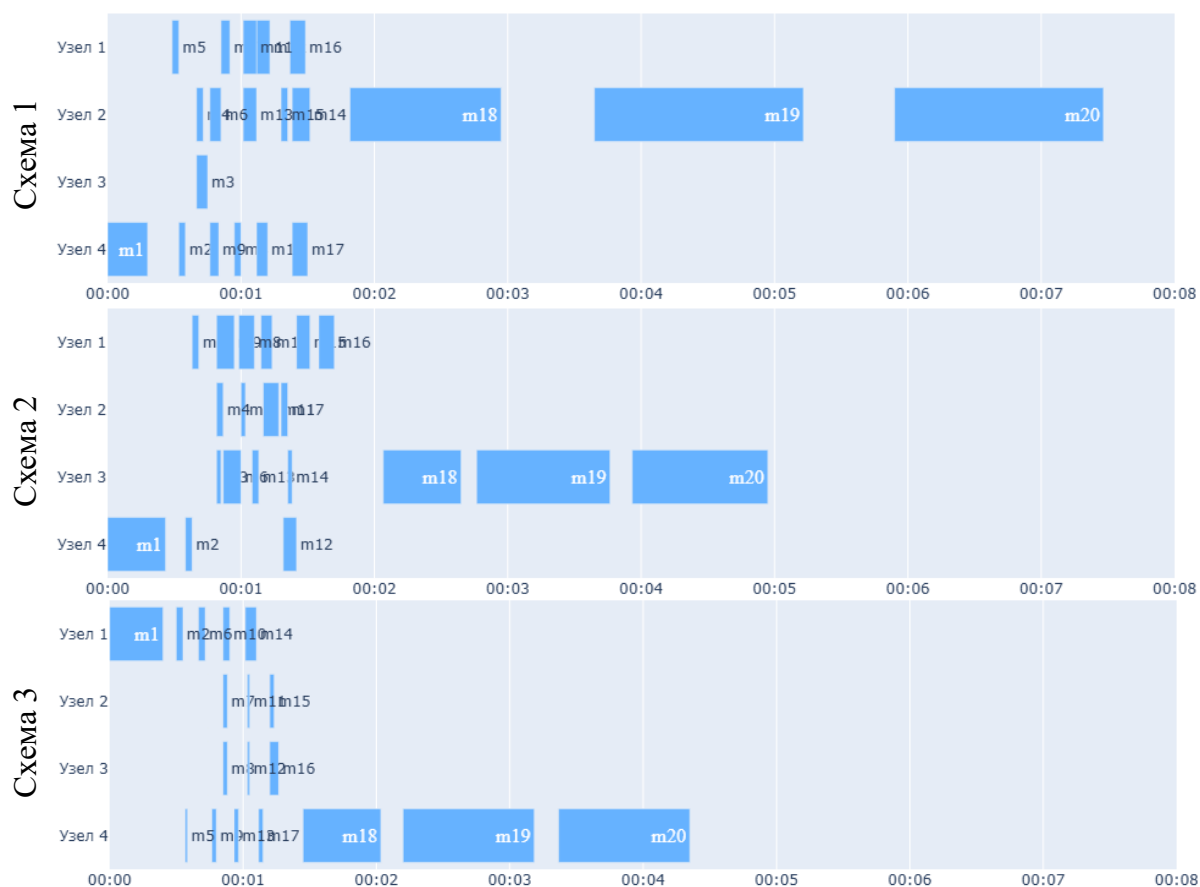


Рисунок 5: Расписания времени выполнения модулей Epigenomics



Рисунок 6: Расписания времени выполнения модулей CyberShake

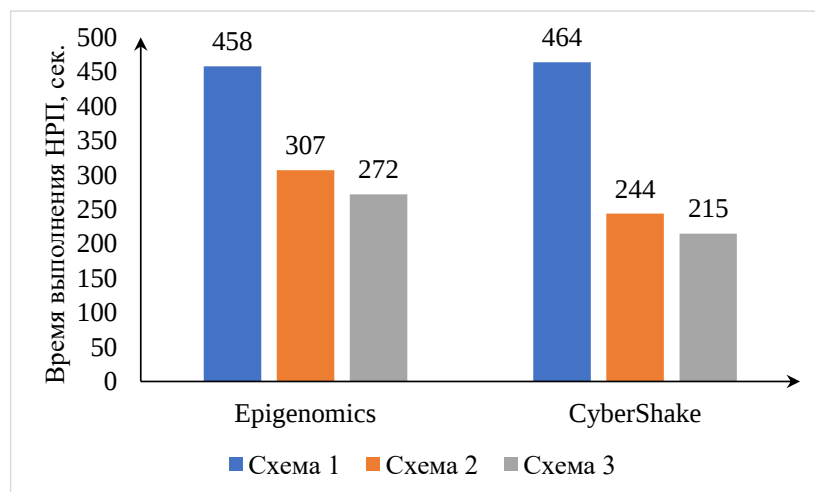


Рисунок 7: Диаграмма времени выполнения НРП

7. Заключение

Предложены новые средства управления для кластера, функционирующего под управлением DAGMan и HTCondor. Экспериментальные исследования показали, что данные средства обеспечивают существенное сокращение времени вычислений при выполнении НРП в сравнении с традиционной схемой управления, используемой в DAGMan и HTCondor. Это сокращение достигается благодаря учету структуры НРП и поддержке механизма прямой передачи данных между узлами кластера.

8. Благодарности

Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации, проект № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9).

9. References

- [1] UNICORE. URL: <https://www.unicore.eu>.
- [2] J. Goecks, A. Nekutenko, J. Taylor et al. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, *Genome biology*, 2010, vol. 11, pp. R86. doi:10.1186/gb-2010-11-8-r86.
- [3] E. Deelman, G. Singh, M.H. Su et al. Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems, *Scientific Programming*, 2005, vol. 13, pp. 219–237. doi:10.1155/2005/128026.
- [4] HTCondor. URL: <https://htcondor.org>.
- [5] M. H. Hilman, M. A. Rodriguez, R. Buyya. Workflow-as-a-service cloud platform and deployment of bioinformatics workflow applications, *Knowledge Management in the Development of Data-Intensive Systems*, Auerbach Publications, 2021, pp. 205–226. doi:10.1201/9781003001188-14.
- [6] P. Singh, *Learn PySpark: Build Python-based Machine Learning and Deep Learning Models*, Apress, 2019, pp. 67–84. doi:10.1007/978-1-4842-4961-1.
- [7] B. Balis. HyperFlow: A model of computation, programming approach and enactment engine for complex distributed workflows, *Future Generation Computer Systems*, 2016, vol. 55, pp. 147–162. doi:10.1016/j.future.2015.08.015.
- [8] A. Feoktistov, S. Gorsky, I. Sidorov et al. Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment, *Communications in Computer and Information Science*, 2020, vol. 1087, pp. 18–32. doi:10.1007/978-3-030-41005-6_2.
- [9] А. Г. Феоктистов, М. Л. Воскобойников, А. Н. Черных, Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs, *Труды Института системного программирования РАН*, 2024, №36(6), с. 195–214. doi:10.15514/ISPRAS-2024-36(6)-11.
- [10] М. Гэри, Д. Джонсон, *Вычислительные машины и труднорешаемые задачи*, М., Мир, 1982.
- [11] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi. Characterizing and profiling scientific workflows, *Future Generation Computer Systems*, 2013, vol. 29, no. 3, pp. 682–692. doi:10.1016/j.future.2012.08.015.