

## СТРАТЕГИИ ПЛАНИРОВАНИЯ МУЛЬТИДИСЦИПЛИНАРНЫХ ПОТОКОВ РАБОТ И УПРАВЛЕНИЯ РЕСУРСАМИ В ОБЛАЧНЫХ ВЫЧИСЛЕНИЯХ

*В. В. Топорков<sup>1,\*</sup>, Д. М. Емельянов<sup>1</sup>, А. Н. Булхак<sup>1</sup>*

<sup>1</sup> Национальный исследовательский университет «МЭИ», Москва, Россия

Предложены и исследованы многофакторные стратегии управления выполнением и планирования наукоемких приложений, таких как CyberShake, Epigenomics, SIPHT, Montage, LIGO, на облачных платформах «поток работ как сервис» (Workflow as a Service — WaaS). Рассматривается комплексное решение проблемы выполнения приложений в модели WaaS с применением различных критериев и сценариев оптимизации на основе циклической схемы. Предложенная схема позволяет поддерживать совокупность технологий и средств управления выполнением междисциплинарных наукоемких приложений на платформах WaaS.

We propose and investigate multifactor strategies for managing the execution and scheduling of knowledge-intensive applications such as CyberShake, Epigenomics, SIPHT, Montage, LIGO, on cloud platforms “Workflow as a Service” (WaaS). A comprehensive solution to the problem of running applications in the WaaS model is considered using various criteria and optimization scenarios based on a cyclic scheme. The proposed scheme allows supporting a set of technologies and management tools for the execution of interdisciplinary knowledge-intensive applications on WaaS platforms.

PACS: 89.20.Ff

### ВВЕДЕНИЕ

Поток работ является наиболее распространенной моделью выполнения наукоемких приложений в различных областях знаний. В рамках этой модели научные сообщества реализуют такие известные проекты, как Montage (астрономия), LIGO (физика гравитационных волн), CyberShake (сейсмология) и ряд других. Поток работ представляется в форме направленного ациклического графа (Directed Acyclic Graph — DAG), вершины которого соответствуют задачам обработки, а дуги — информационным связям.

При создании платформ WaaS возникает ряд серьезных проблем, от решения которых критично зависит эффективность планирования

---

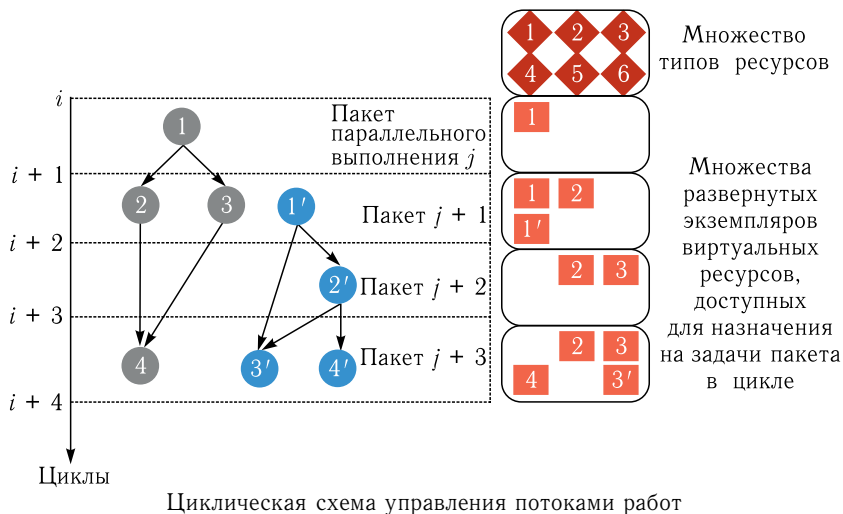
\* E-mail: ToporkovVV@mpei.ru

процессов и использования ресурсов. Это задача отбора провайдеров инфраструктуры IaaS и формирования пула виртуальных машин (ВМ). В известных решениях, как правило, не рассматриваются вопросы одновременного выполнения на одной ВМ множества задач, принадлежащих разным потокам работ. Несмотря на то, что общим в потоках разнотипных работ является представление их в виде DAG, графовые структуры для различных приложений, например Montage, CyberShake и SIPHT, существенно различаются по вычислительной сложности составляющих их задач и видам информационных связей [1]. Еще один вызов — естественное наличие, помимо информационных связей, зависимостей по управлению и циклов в ряде приложений. В некоторых системах управления потоками работ (Workflow Management Systems — WMS) (Pegasus, Apache Airflow, Taverna, Kepler) [2] прибегают к паллиативным приемам для их устранения, что приводит к увеличению времени планирования. Известные к настоящему времени исследования не охватывают всей совокупности указанных выше факторов. Это мотивирует необходимость создания принципиально новых технологий и методов выполнения наукоемких приложений на платформах WaaS.

### **ЦИКЛИЧЕСКАЯ СХЕМА УПРАВЛЕНИЯ ВЫПОЛНЕНИЕМ ПОТОКОВ РАБОТ**

В работе предлагается комплексное решение проблемы выполнения приложений в модели WaaS на основе следующей схемы (рисунок). На платформу WaaS в любой момент времени поступает множество потоков работ. При этом каждая из задач потока может выполняться на некотором подмножестве типов ВМ, которые предоставляются провайдерами IaaS. Время выполнения отдельной задачи потока складывается из следующих составляющих: время обработки как отношение объема вычислений (например в миллионах инструкций) к производительности процессора (например в миллионах инструкций в секунду) на ВМ соответствующего типа, время на обмен данными между задачами потока (чтение и запись в глобальное хранилище, например Amazon S3), время на развертывание ВМ и контейнера в ВМ соответствующего типа. Монетарная стоимость выполнения задачи потока на ВМ конкретного типа определяется как отношение времени выполнения к периоду биллинга развернутых ВМ, умноженное на стоимость одного периода биллинга.

Планирование потоков работ и предоставление ресурсов в виде ВМ и контейнеров осуществляется циклично. Цикл связан с назначением готовых к выполнению задач различных потоков. При этом делается попытка использовать уже развернутые ВМ и контейнеры соответствующего типа. Если это удастся, то из времени выполнения вычитается время на развертывание ВМ и (или) контейнера в ВМ. В случае отсутствия активных ВМ назначение задач откладывается до следующей итерации.



При поступлении соответствующего потока работ до назначения задач на доступные ВМ выполняется его планирование по методу критических работ (МКР) с заданными критериями эффективности [3]. Суть МКР заключается в следующем. Выделяются так называемые критические работы — последовательности неназначенных задач с наибольшей суммой оценок длительности при использовании наиболее производительных ВМ из отобранного пула. Далее критические работы ранжируются по максиминной длительности. Процедура планирования очередной по рангу критической работы может вызвать конфликты между задачами, конкурирующими за один и тот же ресурс и принадлежащими разным критическим работам и потокам. После разрешения конфликтов по алгоритму назначения на конкретные экземпляры ВМ достигнутый промежуточный план должен сохраняться.

## МНОГОФАКТОРНЫЕ СТРАТЕГИИ НАЗНАЧЕНИЯ ВИРТУАЛЬНЫХ РЕСУРСОВ

Одной из важнейших проблем, возникающих при планировании и выполнении вычислительных задач из потоков работ, является эффективное распределение и управление виртуальными ресурсами (ВМ). Под управлением ВМ подразумевается определение моментов запуска (создания) и завершения (остановки, освобождения) отдельных ВМ и контейнеров, а также назначение и порядок выполнения готовых к запуску вычислительных задач. Возможны различные стратегии управления.

Во-первых, можно создавать новую выделенную ВМ специально для выполнения каждой отдельной задачи и освобождать ее, когда задача будет завершена. Эта «жадная» стратегия позволяет строго соотносить

типы и время жизни ВМ с требованиями задач, особенно в условиях, когда время, необходимое для создания и завершения работы ВМ, значительно меньше среднего времени выполнения задачи. Однако время создания и подготовки ВМ включает время на настройку необходимой программной среды и время на копирование и получение требуемых входных данных.

Во-вторых, можно поддерживать некоторый динамически изменяющийся пул постоянно активных ВМ и распределять между ними готовые к запуску задачи (стратегия контроля). Динамика пула подразумевает уменьшение и увеличение количества активных машин в зависимости от вычислительных потребностей в определенный момент времени.

В-третьих, возможна смешанная стратегия, когда некоторый базовый минимальный пул активных ВМ поддерживается во время выполнения потоков работ, но могут быть созданы дополнительные выделенные машины для учета всех неназначенных и готовых к запуску задач.

Реализация «жадной» стратегии представляется тривиальной, стратегии контроля и смешанная требуют особых алгоритмов управления пулом активных ВМ и назначения им входящих задач.

## ФОРМИРОВАНИЕ ПУЛА ВИРТУАЛЬНЫХ МАШИН

Важным начальным шагом для эффективного назначения ВМ является определение количества одновременно требуемых ВМ. Это число зависит от структуры потоков работ, взаимосвязей между задачами, времени их выполнения, истории назначения ВМ и т. д. Таким образом, число ВМ должно определяться динамически. Каждый пакет в циклической схеме (см. рисунок) должен содержать задачи, которые могут выполняться параллельно, не нарушая зависимости по данным и крайние сроки.

В работе [3] предложен алгоритм FTL (Follow The Leader — «следуй за лидером») для определения сбалансированного размера пула активных ВМ. FTL разделяет входящий поток задач на отдельные пакеты, каждый из которых состоит из задач, которые могут выполняться параллельно.

Суть алгоритма состоит в следующем.

1. Выбрать тип ВМ  $Vm_t$  наибольшей производительности (выбрать лидера).

2. Для каждой задачи вычислить раннее время завершения и позднее время старта с учетом прогнозируемого времени выполнения на ВМ-лидере типа  $Vm_t$ .

3. Выбрать задачу с наименьшим ранним временем завершения  $\min t_f^i$  и назначить на лидера. Эта задача помещается в новый пакет для параллельной обработки.

4. Все задачи с поздним временем старта  $\max t_s^i$  меньшим, чем  $\min t_f^i$ , помещаются в пакет параллельной обработки из п. 3. Никакие две задачи из этого пакета не могут быть выполнены последовательно на одной ВМ.

Данный пакет определяет минимальную степень параллелизма очереди задач в течение некоторого промежутка времени.

5. Для оставшихся задач пересчитать раннее время завершения относительно времени  $\min t_f^i$ , начиная с которого лидер может продолжить выполнение задач.

6. Если в очереди остались задачи, не добавленные в пакеты параллельной обработки, то переход к п. 3. Иначе конец алгоритма.

Выбор ВМ наибольшей производительности в качестве лидера (в п. 1) необходим для формирования пакетов параллельной обработки в п. 3, 4 алгоритма. Если лидер с наибольшей производительностью не может выполнить никакие две задачи из пакета последовательно, то никакая другая ВМ также не сможет этого сделать. Невозможность выполнения нескольких задач из одного пакета последовательно следует из того, что даже лидер не успевает сначала выполнить задачу с наименьшим временем завершения  $\min t_f^i$ , а затем любую другую задачу в момент ее позднего времени старта  $\max t_s^i$ .

Для каждого пакета параллельного выполнения далее решается задача назначения на ВМ (НВМ). Сформулировать эту задачу можно следующим образом.

1. Пакет  $B$  содержит  $N_B$  задач для параллельного выполнения на различных ВМ. Заранее известны основные характеристики задач: вычислительный объем, требования к входным данным, крайний срок выполнения на основе МКР.

2. Пул ресурсов содержит  $N_{vm}$  активных ВМ — кандидатов для запуска задач в рассматриваемом цикле выполнения пакета  $B$ . В общем случае  $N_{vm} \neq N_B$ , а кроме того, не все ВМ могут подходить для выполнения задач из  $B$ , в частности, с учетом необходимости соблюдения крайнего срока выполнения.

3. Для выполнения необходимых требований текущего пакета задач  $B$  (или же, напротив, для экономии вычислительных средств) задача предусматривает возможность создания новых или остановку активных ВМ.

Для решения задачи НВМ предлагается использовать венгерский алгоритм Куна–Манкреса для оптимального назначения или поиска совершенного паросочетания в двудольном графе  $G = (T, R, E)$ , где  $T$  представляет множество задач пакета  $B$ ,  $R$  соответствует множеству доступных ресурсов, а  $E$  — множество ребер между  $T$  и  $R$ . Ребро между задачей из  $T$  и ресурсом из  $R$  означает возможность выполнения задания на соответствующей ВМ с соблюдением всех требований. Весом ребра является значение критерия эффективности для данного назначения. В качестве примера такого критерия может служить общее время, необходимое на выполнение задачи с учетом времени копирования данных, или же стоимость такого выполнения. Общая вычислительная

сложность реализации венгерского алгоритма в этом случае составляет  $O(N_B^3)$ . Детальное описание алгоритма НВМ можно найти в работе [1].

## СРАВНЕНИЕ СТРАТЕГИЙ ПЛАНИРОВАНИЯ

В экспериментах по сравнению стратегий планирования («жадной» и контроля на основе алгоритма НВМ) срок выполнения для каждого отдельного потока генерируется случайным образом с использованием ВМ с наивысшей производительностью и ВМ с наименьшей производительностью. Крайний срок определяет базовый план выполнения и, следовательно, влияет на варианты и стратегии распределения ресурсов. Тестирование рабочей нагрузки включает 100 независимых потоков работ. Рабочие процессы построены на основе реальных научных приложений (Montage, Cybershake, Genome, LIGO, SIPHT) и содержат по 50 вершин каждый. В табл. 1 приведены результаты выполнения рабочей нагрузки в зависимости от частоты поступления рабочих процессов (от 0,5 до 100 потоков в минуту). Результаты получены с использованием разработанного программного обеспечения в среде Python 3, процессора Core i5 и 8 ГБ оперативной памяти.

Таблица 1. Влияние частоты поступления потоков работ

| Частота поступления (в минуту) | Стратегия | Время выполнения задачи, с | Общая стоимость ВМ | Число создаваемых ВМ |
|--------------------------------|-----------|----------------------------|--------------------|----------------------|
| 0,5                            | НВМ       | 409280                     | 13226              | 3912                 |
| 1                              | НВМ       | 409486                     | 13277              | 4116                 |
| 2                              | НВМ       | 409602                     | 13316              | 4334                 |
| 6                              | НВМ       | 409601                     | 13279              | 4503                 |
| 12                             | НВМ       | 409586                     | 13257              | 4574                 |
| 60                             | НВМ       | 409578                     | 13168              | 4619                 |
| 100                            | НВМ       | 409596                     | 13168              | 4633                 |
|                                | «Жадная»  | 409650                     | 13906              | 4955                 |

Планирование по алгоритму НВМ напрямую зависит от плотности поступления входящих задач. Так, исходя из частоты поступления рабочих процессов и состава параллельных пакетов, НВМ выделяет на 5–20% меньше экземпляров ВМ по сравнению с «жадной» стратегией, повторно используя их для выполнения нескольких последовательных задач.

Табл. 2 содержит результаты выполнения рабочей нагрузки в зависимости от времени, необходимого для создания (инициализации) и освобождения (сворачивания) ВМ. Этот параметр влияет на результат приме-

Таблица 2. Влияние времени создания и сворачивания ВМ

| Время<br>инициализации/<br>освобождения<br>ВМ | Стратегия | Время<br>выполнения<br>задачи, с | Общая<br>стоимость<br>ВМ | Число<br>создаваемых<br>ВМ |
|-----------------------------------------------|-----------|----------------------------------|--------------------------|----------------------------|
| 0/0                                           | НВМ       | 391361                           | 10878                    | 4175                       |
| 0/0                                           | «Жадная»  | 391559                           | 10885                    | 4955                       |
| 10/1                                          | НВМ       | 391297                           | 11009                    | 4127                       |
| 10/1                                          | «Жадная»  | 391559                           | 11053                    | 4955                       |
| 100/10                                        | НВМ       | 391449                           | 12144                    | 4125                       |
| 100/10                                        | «Жадная»  | 391559                           | 12557                    | 4955                       |
| 300/30                                        | НВМ       | 391453                           | 14576                    | 4134                       |
| 300/30                                        | «Жадная»  | 391559                           | 15899                    | 4955                       |
| 500/50                                        | НВМ       | 391413                           | 17076                    | 4118                       |
| 500/50                                        | «Жадная»  | 391559                           | 19242                    | 4955                       |

нения как алгоритма НВМ, так и «жадного» алгоритма, поскольку общая стоимость напрямую зависит от общего времени использования ВМ, включая периоды инициализации и освобождения ВМ. На основании полученных результатов можно утверждать, что НВМ выделяет на 16 % меньше экземпляров ВМ для выполнения того же количества задач, что дает преимущество до 10 % над «жадными» алгоритмами по критерию общей стоимости. Преимущество увеличивается с увеличением времени создания и инициализации ВМ.

## ЗАКЛЮЧЕНИЕ

Предложен комплексный подход к планированию междисциплинарных потоков работ и управлению ресурсами на платформах WaaS. Предлагаемые решения позволяют управлять пулом активных ВМ, определяя для каждого экземпляра интервалы создания, подготовки, использования, передачи данных и выключения. В настоящее время основным ограничивающим фактором является относительно высокая (кубическая) вычислительная сложность алгоритма НВМ в зависимости от степени параллелизма входящего потока задач. Таким образом, существует естественное ограничение на размер потоков работ, планирование которых может быть завершено за приемлемое время. Дальнейшая работа будет касаться проблем сложности алгоритмов планирования на масштабируемых платформах WaaS.

Исследование проведено в ходе выполнения гранта Российского научного фонда № 25-21-00656.

## СПИСОК ЛИТЕРАТУРЫ

1. *Toporkov V., Yemelyanov D., Bulkhak A., Pirogova M.* Job Batch Scheduling in Workflow-as-a-Service Platforms // Proc. PCT 2024. CCIS. Cham: Springer, 2024. V. 2241. P. 65–79.
2. *Amstutz P., Mikheev M., Crusoe M.R. et al.* Existing Workflow Systems. <https://s.apache.org/existing-workflow-systems> (accessed 18.08.2024).
3. *Toporkov V., Yemelyanov D., Bulkhak A.* Scheduling in Workflow-as-a-Service Model with Pre-parameterized DAG Using Inaccurate Estimates // Proc. ICCS 2025 Workshops. LNCS. Cham: Springer, 2025. V. 15912. P. 168–182.