



УДК 004.45

## МОДЕЛЬ ДИНАМИЧЕСКОЙ МИГРАЦИИ ВИРТУАЛЬНЫХ МАШИН С ГИБРИДНЫМ ПОДХОДОМ

С.М. Алексанков<sup>a,b</sup><sup>a</sup> Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация<sup>b</sup> АО «НИИ «Масштаб», Санкт-Петербург, 194100, Российская ФедерацияАдрес для переписки: [aleksankov.sergey@gmail.com](mailto:aleksankov.sergey@gmail.com)

### Информация о статье

Поступила в редакцию 27.04.17, принята к печати 15.05.17

doi: 10.17586/2226-1494-2017-17-4-725-732

Язык статьи – русский

**Ссылка для цитирования:** Алексанков С.М. Модель динамической миграции виртуальных машин с гибридным подходом // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 4. С. 725–732. doi: 10.17586/2226-1494-2017-17-4-725-732

### Аннотация

**Предмет исследования.** Исследован процесс переноса виртуальных машин без прерывания сервиса («динамическая миграция», англ. «Live Migration»). Предложена модель, позволяющая оценить время переноса виртуальных машин и время недоступности сервисов при динамической миграции с «гибридным подходом» («Hybrid approach»). Миграция виртуальных машин позволяет приложениям прозрачно перемещаться вместе с их средами выполнения между физическими машинами и применяется для балансировки загрузки и оптимизации размещения виртуальных машин на узлах кластера в центрах обработки данных. **Метод.** Предложен гибридный алгоритм динамической миграции с позиции исследования времени переноса и времени недоступности сервисов при переносе виртуальных машин. **Основные результаты.** Предложена аналитическая модель, позволяющая оценивать время переноса виртуальных машин и недоступности сервисов при динамической миграции с гибридным подходом. Выполнен анализ механизмов миграции при использовании гибридного подхода в сравнении с другими механизмами. Показано, что использование при переносе виртуальных сервисов с одинаково высокими скоростями записи и чтения гибридного подхода позволяет одновременно минимизировать время миграции и время недоступности сервисов по сравнению с механизмами с копированием до остановки и после остановки. **Практическая значимость.** Результаты работы представляют интерес при проведении анализа и оптимизации вычислительных процессов в центрах обработки данных, в которых виртуальные машины могут перемещаться между физическими серверами при профилактических работах и для обеспечения интеллектуальной балансировки нагрузки. Также предложенные модели могут применяться при определении возможности и целесообразности включения механизмов миграции в программные продукты управления виртуализированной инфраструктурой различного применения.

### Ключевые слова

живая миграция, динамическая миграция, виртуализация, виртуальные машины, высокая доступность, кластеры, моделирование

## LIVE MIGRATION MODEL OF VIRTUAL MACHINES WITH HYBRID APPROACH

S.M. Aleksankov<sup>a,b</sup><sup>a</sup> ITMO University, Saint Petersburg, 197101, Russian Federation<sup>b</sup> R&D Institute "Mashtab", Saint Petersburg, 194100, Russian FederationCorresponding author: [aleksankov.sergey@gmail.com](mailto:aleksankov.sergey@gmail.com)

### Article info

Received 27.04.17, accepted 15.05.17

doi: 10.17586/2226-1494-2017-17-4-725-732

Article in Russian

**For citation:** Aleksankov S.M. Live migration model of virtual machines with hybrid approach. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2017, vol. 17, no. 4, pp. 725–732 (in Russian). doi: 10.17586/2226-1494-2017-17-4-725-732

### Abstract

**Subject of Research.** The paper deals with the process of live migration for virtual machines without service interruption. We propose a model for time estimation of virtual machine migration and service down time at live migration with "hybrid approach". Migration of virtual machines enables the applications to navigate transparently together with their execution environments between physical hosts. The migration of virtual machines without service interruption (Live Migration) is

used to balance the load efficiently and optimize the deployment of virtual machines on physical nodes in data centers. **Method.** We have proposed a hybrid approach for live migration from the position of migration time and service down time during the migration of virtual machines. **Main Results.** An analytical model is proposed that allows estimating the time of virtual machines migration and the inaccessibility of services during live migration with a hybrid approach. The migration process is analyzed with the use of a hybrid approach and compared to the other mechanisms. It is shown that the usage of hybrid approach virtual services with the same high read-write speed gives the possibility to minimize simultaneously migration time and service down time as compared to the mechanisms with copying process before stop and after it. **Practical Relevance.** Predicting the migration time and the inaccessibility of services is of interest in analyzing and optimizing computational processes in data centers where virtual machines can move between physical servers during preventive and emergency operations to provide intelligent load balancing. Also, the proposed models can be used to identify the possibility and feasibility of migration mechanisms incorporating into the management software products of virtualized infrastructure for various applications.

**Keywords**

live migration, network migration, virtualization, virtual machines, high availability, clusters, modeling

**Введение**

Высокая надежность, отказоустойчивость и безопасность систем критического применения реализуются на основе технологий виртуализации [1–14] при резервировании основных подсистем [1–7, 9], в том числе средств хранения [15], обработки и передачи данных [16] с консолидацией ресурсов, при кластеризации [17–20] и динамическом распределении запросов [21–23].

Для обеспечения непрерывности доступа клиентов к приложениям и сервисам используются кластеры высокой доступности, которые представляются для пользователя как логическое объединение нескольких физических серверов. Высокая доступность ресурсов кластерной системы при проведении профилактических работ обеспечивается за счет перемещения служб или приложений между физическими серверами [16–20]. неполадки в кластере могут выразиться для пользователя в кратковременном снижении производительности или недоступности какого-либо ресурса на время от нескольких секунд до нескольких минут в зависимости от типа неполадок и организации системы [1, 2].

Миграция виртуальных машин (ВМ) является неотъемлемой частью технологии виртуализации, которая позволяет приложениям прозрачно перемещаться вместе с их средами выполнения между физическими серверами [3].

С помощью динамической миграции («живая миграция», англ. «Live Migration») [1–6, 9–14] ВМ с одного физического сервера могут быть переданы на другой практически без прерывания работы приложений, что позволяет существенно повысить доступность системы при обеспечении непрерывности вычислительного процесса в случаях балансировки нагрузки, проведения аварийных и профилактических работ.

Динамическая миграция может использоваться для переноса ВМ как между физическими серверами в рамках локального/территориального кластера, так и между географически распределенными серверами или центрами обработки данных. При переносе ВМ в рамках локального кластера, как правило, используется общее хранилище с виртуальными дисками ВМ, что может значительно ускорить миграцию за счет переноса только оперативной памяти, регистров виртуальных процессоров и состояний виртуальных устройств ВМ. Когда использование виртуального хранилища невозможно, в том числе при перемещении ВМ в географически распределенном кластере, переносу подлежат, в том числе, виртуальные диски ВМ, размер которых может значительно превосходить размер оперативной памяти и регистров, что увеличивает как время переноса, так и сложность прогнозирования этого времени. На первый взгляд, технология динамической миграции выглядит предпочтительней переноса с остановкой выполняемого сервиса, особенно при использовании общей системы хранения данных в кластере, но нужно отметить, что динамическая миграция характеризуется высокой сложностью прогнозирования времени и поведения переноса ВМ, что особенно важно, если требуется закончить миграцию всех сервисов кластера к заданному моменту времени (например, к моменту отключения электроэнергии на объекте).

В [7, 8] предложены модели для расчета времени переноса и времени недоступности сервисов для подходов с копированием данных до остановки ВМ (pre-copy approach) [3, 7] и копированием данных после остановки ВМ (post-copy approach) [3, 8]. Показано, что эти подходы позволяют сократить время миграции и прерывания выполнения сервиса только при малых значениях скоростей записи чтения соответственно. При одинаково высоких средних скоростях записи и чтения эффективность этих подходов низка. Для решения проблемы может использоваться основанный на их комбинации «гибридный подход» (англ. «Hybrid approach») [3], сочетающий в себе достоинства исследованных подходов и частично зависящий от скорости чтения данных сервисом, так и от скорости изменения данных сервисом. Указанный подход более сложен, и прогнозирование времени миграции и времени недоступности с использованием гибридного подхода представляет интерес с точки зрения его использования в продуктах виртуализации и минимизации рисков.

### Задачи исследования

В работе решается задача оценки времени переноса виртуальных машин и времени недоступности сервисов при динамической миграции с гибридным подходом, который должен сочетать в себе преимущества подходов с копированием данных до остановки VM (pre-cory) и с копированием данных после остановки VM (post-cory). Для этого необходима разработка аналитической модели оценки времени миграции и времени недоступности.

Динамическая миграция позволяет переносить виртуальные машины практически без прерывания выполнения приложений. Процесс динамической миграции можно разделить на два этапа:

1. передача данных (регистры *vsu*, память, диск(и)) на узел назначения;
2. передача выполнения VM на узел назначения.

Два подхода к реализации динамической миграции [7, 8] основаны на различной последовательности этих шагов. При копировании данных до остановки VM (pre-cory) сначала происходит передача данных, затем – передача управления, а при копировании данных после остановки VM (post-cory) сначала происходит передача управления узлу назначения, а затем – передача данных.

При гибридном подходе гипервизор или программное обеспечение виртуализации на этапе подготовки динамической миграции VM должны определить количество данных, передаваемых в режиме итеративной передачи (как при копировании данных до остановки VM) и в фоновом режиме (как при копировании данных после остановки VM).

Процесс гибридного подхода можно условно разделить на четыре стадии (рис. 1).

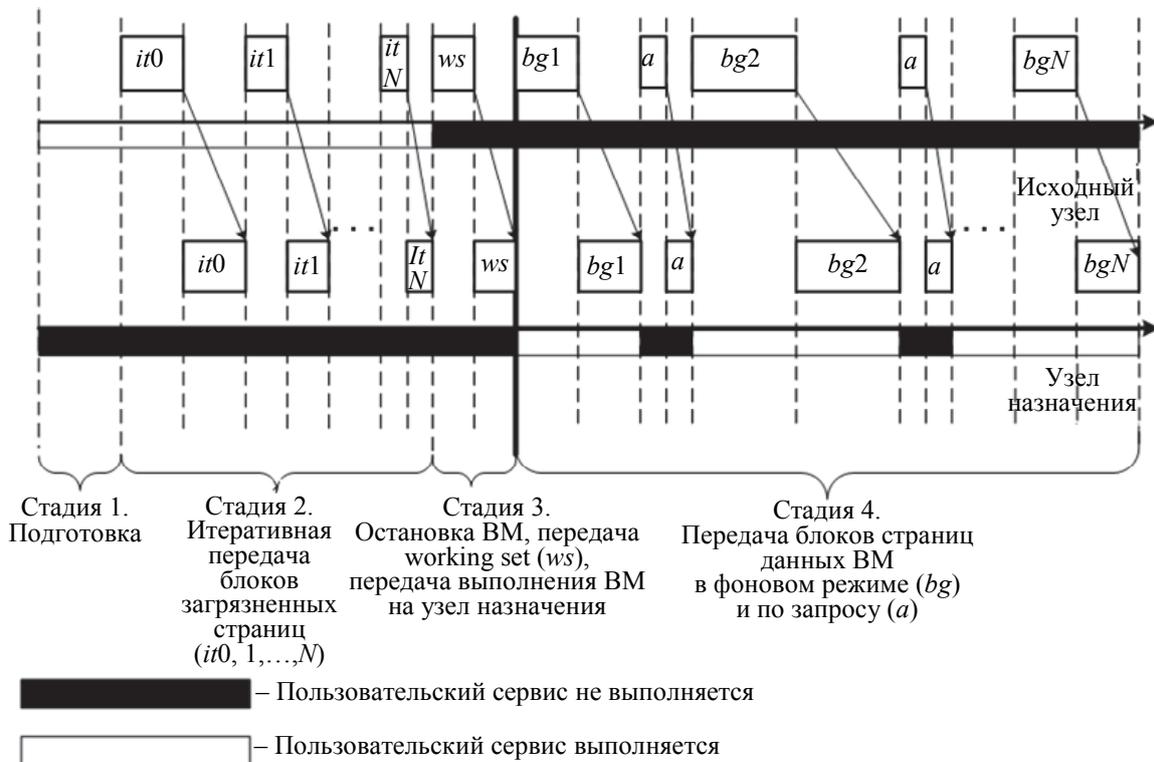


Рис. 1. Временная диаграмма процесса динамической миграции виртуальной машины с гибридным подходом

На исходном узле должна выполняться подготовка к миграции: измерение на некотором временном отрезке средних скоростей чтения и записи перемещаемого сервиса, определение на основании этих сведений объема данных, которые будут переданы в итеративном и фоновом режиме, определение, по возможности, наиболее используемых сервисом страниц для чтения, которые должны быть переданы в первую очередь в итеративном режиме для обхода сетевых ошибок доступа в дальнейшем.

Происходит передача первой части образа диска и памяти в итеративном режиме с исходного узла на узел назначения. Это означает, что в несколько шагов на узел назначения передаются страницы, которые были изменены (загрязнены, «dirty» [3]) за время каждой предыдущей передачи загрязненных страниц. Каждый такой шаг передачи страниц, которые были загрязнены за время предыдущей передачи загрязненных страниц, называется итерацией.

На исходном узле осуществляется остановка VM. На узел назначения передаются регистры виртуальных процессоров, состояния внешних устройств («working set»), после чего на узле назначения возобновляется работа VM. Происходит передача в фоновом режиме второй части образа диска и памяти VM.

Если ВМ пытается выполнить операцию чтения еще не переданных страниц, должна произойти приостановка выполнения пользовательского сервиса, и необходимые страницы должны быть переданы с исходного узла на узел назначения.

При построении моделей предполагается, что ВМ работают на кластере из двух физических серверов без общего хранилища.

### Время динамической миграции при гибридном подходе

Будем считать известными скорость чтения данных ВМ –  $\lambda_r$ , скорость записи данных ВМ –  $\lambda_w$ , скорость передачи данных –  $\mu$ , размер образа виртуальной машины (диск, память) –  $V$ . Время перемещения ВМ  $T_m$  будем рассчитывать как сумму времен подготовки и передачи первой части образа диска, памяти ВМ в итеративном режиме –  $T_1$ , остатка первой части образа диска и памяти ВМ с остановкой выполнения ВМ –  $T_2$ , второй части образа диска и памяти ВМ в фоновом режиме –  $T_3$ :

$$T_m = T_1 + T_2 + T_3. \quad (1)$$

$T_1$  равно сумме времен передачи данных в каждой итерации:

$$T_1 = \sum_{i=0}^n t_i,$$

где время передачи в текущей итерации  $t_i = v_{i-1} / \mu$  вычисляется как объем данных, измененных в предыдущей итерации ( $v_{i-1}$ ), переданных узлу назначения со скоростью передачи данных  $\mu$ .

Объем данных, измененных в текущей итерации, определяется как объем данных, которые были изменены за время передачи данных, измененных в предыдущей итерации. Таким образом,  $v_i$  равно объему данных, измененных в предыдущей итерации  $v_{i-1}$ , умноженному на интенсивность записи  $\lambda_w$  и деленному на интенсивность передачи  $\mu$ . Другими словами, вычисляется размер первой части образа и памяти ВМ  $V$ , умноженный на  $\lambda_w$  в степени, равной номеру итерации  $i$ , деленный на  $\mu$  в степени  $i$ :

$$v_i = v_{i-1} \cdot \frac{\lambda_w}{\mu} = V_1 \cdot \frac{\lambda_w^i}{\mu^i}. \quad (2)$$

Объем первой части образа диска и памяти ВМ определяется как произведение общего объема образа памяти и диска ВМ  $V$ , умноженного на отношение скорости записи ВМ  $\lambda_w$  к сумме скоростей записи  $\lambda_w$  и чтения  $\lambda_r$  ВМ:

$$V_1 = V \cdot \frac{\lambda_w}{\lambda_r + \lambda_w}. \quad (3)$$

Подставим выражения (3) и (2) в (1) и получим формулу, по которой можно рассчитать время передачи данных в текущей итерации  $t_i$ :

$$t_i = V \cdot \frac{\lambda_w}{\lambda_r + \lambda_w} \cdot \frac{\lambda_w^i}{\mu^{i+1}}.$$

Время передачи первой части образа диска и памяти ВМ в итеративном режиме вычисляется как

$$T_1 = \sum_{i=0}^n V \cdot \frac{\lambda_w}{\lambda_r + \lambda_w} \cdot \frac{\lambda_w^i}{\mu^{i+1}}.$$

Количество итераций  $n$  можно оценить из следующего неравенства. Объем данных, которые загрозились во время последней итерации, должен быть меньше или равен  $l$ :  $v_n \leq l$  или  $V_1 \cdot \lambda_w^n \leq l$ . Тогда количество итераций  $n$  можно выразить через логарифм  $n = \log_{\lambda_w} \left( \frac{l}{V_1} \right)$ .  $T_2$  определяется как время передачи со скоростью  $\mu$  объема данных  $l$ , при котором ВМ останавливает свое выполнение:

$$T_2 = \frac{V_2}{\mu} = \frac{l}{\mu}.$$

$T_3$  равно сумме времен передачи данных по запросу  $T_a$  и в фоновом режиме  $T_b$ :  $T_3 = T_a + T_b$ . При этом время передачи по запросу  $T_a$  определяется как объем данных  $V_a$ , переданных за это время со скоростью передачи данных  $\mu$ , а время передачи в фоновом режиме  $T_b$  – как объем данных  $V_b$ , переданных за это время со скоростью  $\mu$ , умноженной на коэффициент накладных расходов на другие задачи  $\alpha$ :

$$T_3 = \frac{V_a}{\mu} + \frac{V_b}{\mu \cdot \alpha}, \quad (4)$$

Объем данных, переданных по запросу  $V_a$ , определяется как произведение времени передачи данных  $T_3$  и скорости чтения данных  $\lambda_r$ :

$$V_a = \lambda_r \cdot T_3, \quad (5)$$

т.е. как тот объем данных, к которому ВМ обращалась для чтения. Объем данных, переданных в фоновом режиме, определяется как объем данных, которые осталось передать помимо данных, переданных по запросу, т.е. как разность объема второй части образа памяти и диска ВМ  $V_3$  и  $V_a$ :

$$V_b = V_3 - V_a = V_3 - \lambda_r \cdot T_3. \quad (6)$$

Объем второй части образа диска и памяти ВМ  $V_3$  определяется как разница между общим объемом образа диска и памяти ВМ и первой частью этого объема:

$$V_3 = V - V_1 = V \cdot \left(1 - \frac{\lambda_w}{\lambda_r + \lambda_w}\right).$$

Подставив (5) и (6) в (4), получаем следующее выражение:

$$T_3 = \frac{\lambda_r \cdot T_3}{\mu} + \frac{V_3 - \lambda_r \cdot T_3}{\mu \cdot \alpha} = \frac{V_3}{\mu \cdot \alpha + \lambda_r \cdot (1 - \alpha)},$$

откуда получаем  $T_3$ :

$$T_3 = \frac{V_3}{\mu \cdot \alpha + \lambda_r \cdot (1 - \alpha)} = V \cdot \left(1 - \frac{\lambda_w}{\lambda_r + \lambda_w}\right) / (\mu \cdot \alpha + \lambda_r \cdot (1 - \alpha)).$$

Время недоступности сервиса при миграции ВМ с гибридным подходом  $T_d$  равно сумме времени передачи остатка первой части образа диска и памяти ВМ  $T_2$  с остановкой выполнения ВМ и времени передачи по запросу  $T_a$ :

$$T_d = T_2 + T_a = \frac{l}{\mu} + \frac{\lambda_r \cdot T_3}{\mu} = \frac{l}{\mu} + \lambda_r \cdot V \cdot \left(1 - \frac{\lambda_w}{\lambda_r + \lambda_w}\right) / (\mu \cdot (\mu \cdot \alpha + \lambda_r \cdot (1 - \alpha))).$$

### Результаты расчетов времен переноса и недоступности сервисов

Ниже приводятся расчеты для случая переноса ВМ с типовым сервисом (например, корпоративным порталом небольшого предприятия), с размером диска 10 Гб и оперативной памятью – 1 Гб, запущенном на кластере из двух серверов общего назначения, связанных локальной сетью Fast Ethernet 100 Мб/с. В качестве устройств хранения данных в серверах предполагаются жесткие диски (HDD), соответственно считаем, что средние скорости записи и чтения внутри ВМ при конкурентном доступе должны быть в пределах 20 Мб/с. Тогда исходные данные будут следующими: размер образа ВМ  $V=11264$  Мб/с, пропускная способность  $\mu=12,5$  Мб/с, предел, при котором останавливается ВМ при итеративной передаче, примем  $l=1000$  Мб, параметр накладных расходов  $\alpha=0,3$ . Расчеты проведены с использованием пакета компьютерной математики Mathcad 15.

На рис. 2, а, представлен график зависимостей времени миграции ВМ от средней скорости записи виртуального сервиса при средней скорости чтения  $\lambda_r = 3$  Мб/с. Все графики приведены для расчетов времени миграции гибридного подхода, а также рассматриваемых в предыдущих работах подходов с копированием данных до остановки выполнения ВМ и копированием данных после остановки выполнения ВМ [8, 9]. Как видно из графиков, время миграции увеличивается и стремится к бесконечности для гибридного подхода и для подхода с копированием до остановки с увеличением скорости записи, при этом для гибридного подхода время миграции увеличивается медленнее, так как оно только частично зависит от скорости записи. Для подхода с копированием после остановки время миграции не меняется, так как не зависит от скорости записи.

На рис. 2, б, представлен график зависимостей времени миграции ВМ от средней скорости чтения виртуального сервиса при средней скорости записи  $\lambda_w = 3$  Мб/с. Здесь время миграции уменьшается для гибридного подхода и для подхода с копированием после остановки с увеличением скорости чтения. Это связано с тем, что в обоих подходах с увеличением скорости чтения данных увеличивается количество более быстрых передач страниц данных по запросу, которые, как мы увидим далее, влекут за собой приостановку сервиса. При этом для гибридного подхода время миграции уменьшается медленнее, так как оно зависит частично от скорости чтения. Для подхода с копированием до остановки время миграции не меняется, так как не зависит от скорости чтения виртуального сервиса.

На рис. 3, а, представлен график зависимостей времени недоступности виртуального сервиса при динамической миграции от средней скорости записи виртуального сервиса при средней скорости чтения  $\lambda_r = 3$  Мб/с. Как видно, время недоступности сервиса постоянно для всех подходов. Это связано с тем, что гибридный подход и подход копирования до остановки влечет при увеличении скорости записи только увеличение времени миграции за счет увеличения числа итераций передачи страниц, а подход с копированием после остановки не зависит от скорости записи виртуального сервиса.

На рис. 3, б, представлен график зависимостей времени недоступности виртуального сервиса при динамической миграции от средней скорости чтения виртуального сервиса при средней скорости записи  $\lambda_w = 3$  Мб/с. Время недоступности сервиса растет и стабилизируется на некотором уровне для гибридного

подхода и для подхода с копированием после остановки с увеличением скорости чтения, так как в этих подходах соответственно увеличивается количество передач страниц по запросу, влекущих приостановку сервиса. После превышения некоторого значения средней скорости чтения виртуального сервиса все данные ВМ начинают передаваться по запросу, и время недоступности сервиса становится равным времени миграции. При этом для гибридного подхода время недоступности увеличивается медленнее, так как оно зависит частично от скорости чтения. Для подхода с копированием до остановки время недоступности сервиса не меняется, так как не зависит от скорости чтения виртуального сервиса.

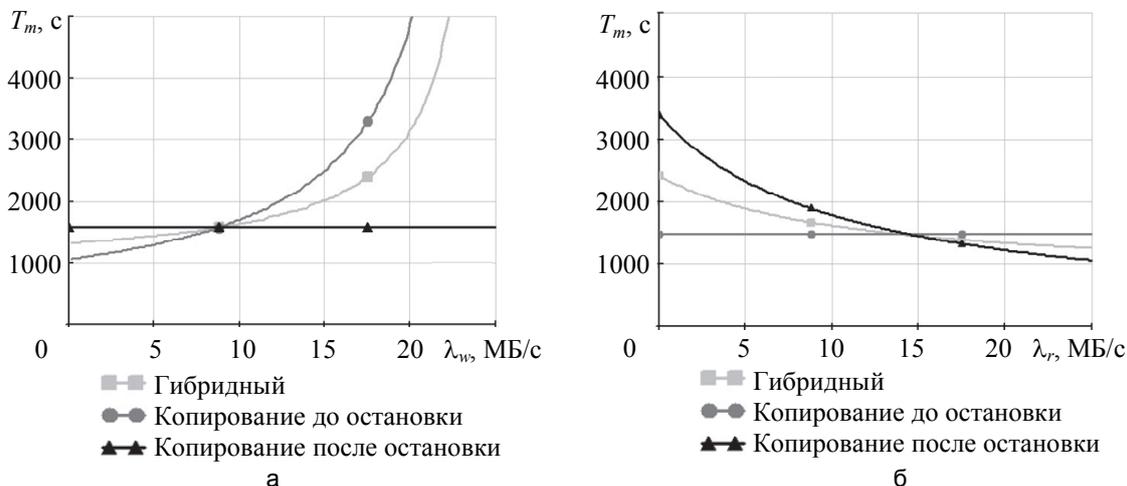


Рис. 2. Зависимости времени миграции  $T_m$  виртуальной машины от скорости передачи записи (а) и скорости чтения (б) виртуального сервиса

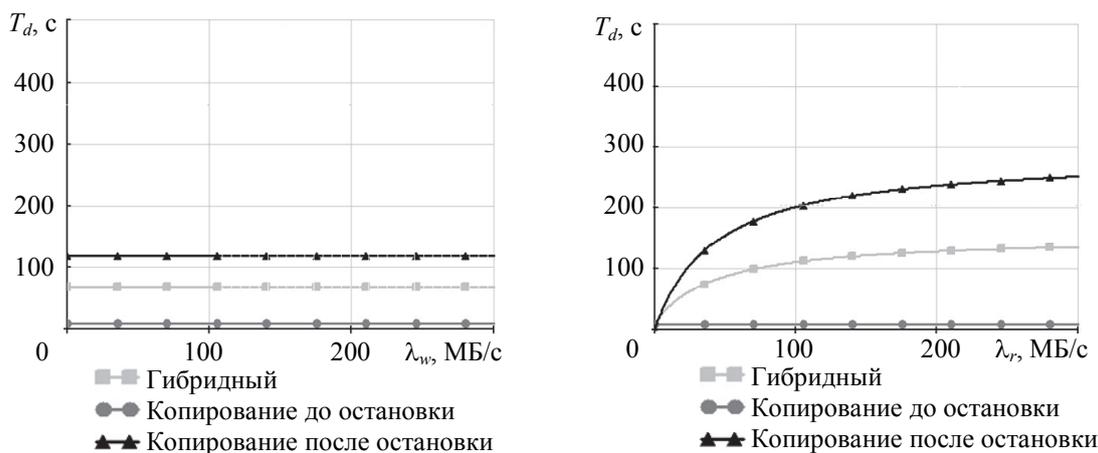


Рис. 3. Зависимости времени недоступности  $T_d$  сервисов от скорости передачи записи (а) и скорости чтения (б) виртуального сервиса

Расчеты подтверждают, что при переносе виртуальных сервисов с одинаково высокими скоростями записи и чтения, гибридный подход позволяет одновременно минимизировать время миграции и время недоступности сервисов по сравнению с механизмами с копированием до остановки и после остановки, которые дают возможность минимизировать либо время недоступности, либо время миграции сервисов соответственно. Но вместе с этим подход с копированием до остановки может из-за увеличившегося количества итераций передачи загрязненных страниц многократно и неконтролируемо увеличить время миграции, что может быть критично при необходимости окончания миграций всех сервисов кластера к заданному времени (например, к моменту отключения электроэнергии на объекте). А подход с копированием после остановки может увеличить время недоступности сервиса из-за роста количества сетевых ошибок страниц при передаче данных в фоновом режиме.

Следует отметить, что гибридный подход эффективен, когда сервис интенсивно обращается к данным и на чтение, и на запись, но при этом он характеризуется дополнительной сложностью реализации подготовительной стадии; сложностью прогнозирования времени миграции, а также тем, что скорости чтения и записи должны быть меньше пропускной способности канала передачи данных.

## Заключение

Предложена модель, позволяющая оценить время переноса виртуальных машин и время недоступности сервисов при динамической миграции с гибридным подходом (Hybrid approach).

Показано, что использование гибридного подхода при переносе виртуальных сервисов с одинаково высокими скоростями записи и чтения позволяет одновременно минимизировать время миграции и время недоступности сервисов по сравнению с механизмами с копированием до остановки и после остановки.

Предлагаемые модели могут использоваться для анализа и оптимизации вычислительных процессов в центрах обработки данных кластерной архитектуры, в которых виртуальные машины могут перемещаться между физическими серверами для проведения профилактических и аварийных работ, а также для обеспечения балансировки нагрузки.

## Литература

1. Jin H., Li D., Wu S., Shi X., Pan X. Live virtual machine migration with adaptive memory compression // Proc. IEEE Int. Conf. on Cluster Computing (CLUSTER '09). New Orleans, USA, 2009. Art. 5289170. doi: 10.1109/CLUSTER.2009.5289170
2. Bing Wei. A novel energy optimized and workload adaptive modeling for live migration // International Journal of Machine Learning and Computing, 2012. V. 2. N 2. P. 162–167. doi: 10.7763/IJMLC.2012.V2.106
3. Sahni S., Varma V. A hybrid approach to live migration of virtual machines // Proc. IEEE Int. Conf. on Cloud Computing for Emerging Markets (CCEM 2012). Bangalore, India, 2012. P. 12–16. doi: 10.1109/CCEM.2012.6354587
4. Косивченко А. Комплексное решение: виртуализация + отказоустойчивый кластер // Системный администратор, 2009. № 9(82). С. 16–19.
5. Обзор миграции виртуальных машин и хранилища [Электронный ресурс]. 2013. Режим доступа: <https://technet.microsoft.com/ru-ru/library/jj628158.aspx>, свободный. Яз. рус. (дата обращения 30.05.2015).
6. Wu Y., Zhao M. Performance modeling of virtual machine live migration // Proc. IEEE 4<sup>th</sup> Int. Conf. on Cloud Computing (CLOUD 2011). Washington, USA, 2011. P. 492–499. doi: 10.1109/CLOUD.2011.109
7. Алексанков С.М. Модели динамической миграции с итеративным подходом и сетевой миграции виртуальных машин // Научно-технический вестник информационных технологий, механики и оптики. 2015. Т. 15. № 6. С. 1098–1104. doi: 10.17586/2226-1494-2015-15-6-1098-1104
8. Алексанков С.М. Модель процесса динамической миграции с копированием данных после остановки виртуальных машин // Изв. вузов. Приборостроение. 2016. Т. 59. № 5. С. 348–354. doi: 10.17586/0021-3454-2016-59-5-348-354
9. Богатырев В.А., Алексанков С.М., Демидов Д.В., Беззубов В.Ф. Надежность резервированного вычислительного комплекса при ограниченном восстановлении // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 3 (85). С. 67–72.
10. Zhu J., Jiang Z., Xiao Z., Li X. Optimizing the performance of virtual machine synchronization for fault tolerance // IEEE Transactions on Computers. 2011. V. 60. N 12. P. 1718–1729. doi: 10.1109/TC.2010.224
11. Agrawal S. Hardware virtualization towards a proficient computing environment // International Journal of Innovation and Applied Studies. 2013. V. 3. N 2. P. 528–534.
12. Ibrahim K.Z., Hofmeyr S., Iancu C., Roman E. Optimized pre-copy live migration for memory intensive applications // Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis. Seattle, USA, 2011, art. 40. doi: 10.1145/2063384.2063437
13. Ye K., Jiang X., Huang D., Chen J., Wang B. Live migration of multiple virtual machines with resource reservation in cloud computing environments // Proc. 4<sup>th</sup> Int. Conf. on Cloud Computing, CLOUD 2011. Washington, USA, 2011. P. 267–274. doi: 10.1109/CLOUD.2011.69
14. Malik V, Barde C.R. Live migration of virtual machines in cloud environment using prediction of CPU usage // International Journal of Computer Applications. 2015. V. 117.

## References

1. Jin H., Li D., Wu S., Shi X., Pan X. Live virtual machine migration with adaptive memory compression. *Proc. IEEE Int. Conf. on Cluster Computing, CLUSTER'09*. New Orleans, USA, 2009, art. 5289170. doi: 10.1109/CLUSTER.2009.5289170
2. Bing Wei. A novel energy optimized and workload adaptive modeling for live migration. *International Journal of Machine Learning and Computing*, 2012, vol. 2, no. 2, pp. 162–167. doi: 10.7763/IJMLC.2012.V2.106
3. Sahni S., Varma V. A hybrid approach to live migration of virtual machines. *Proc. IEEE Int. Conf. on Cloud Computing for Emerging Markets, CCEM 2012*. Bangalore, India, 2012, pp. 12–16. doi: 10.1109/CCEM.2012.6354587
4. Kosivchenko A. Complete solution: virtualization + failover cluster. *Sistemnyi Administrator*, 2009, no. 9, pp. 16–19. (In Russian)
5. *Obzor Migratsii Virtual'nykh Mashin i Khranilishcha* [Review of virtual machines migration and storage]. 2013. Available at: <https://technet.microsoft.com/ru-ru/library/jj628158.aspx> (accessed 30.05.2015).
6. Wu Y., Zhao M. Performance modeling of virtual machine live migration. *Proc. IEEE 4<sup>th</sup> Int. Conf. on Cloud Computing, CLOUD 2011*. Washington, USA, 2011, pp. 492–499. doi: 10.1109/CLOUD.2011.109
7. Aleksankov S.M. Models of live migration with iterative approach and move of virtual machines. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2015, vol. 15, no. 6, pp. 1098–1104. (In Russian) doi: 10.17586/2226-1494-2015-15-6-1098-1104
8. Aleksankov S.M. Model of live migration process with data copying after virtual machines stopping. *Journal of Instrument Engineering*, 2016, vol. 59, no. 5, pp. 348–354. (In Russian) doi: 10.17586/0021-3454-2016-59-5-348-354
9. Bogatyrev V.A., Aleksankov S.M., Demidov D.V., Bezzubov V.F. Reliability of redundant computing complex with recovery limitations. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2013, no. 3, pp. 67–72. (In Russian)
10. Zhu J., Jiang Z., Xiao Z., Li X. Optimizing the performance of virtual machine synchronization for fault tolerance. *IEEE Transactions on Computers*, 2011, vol. 60, no. 12, pp. 1718–1729. doi: 10.1109/TC.2010.224
11. Agrawal S. Hardware virtualization towards a proficient computing environment. *International Journal of Innovation and Applied Studies*, 2013, vol. 3, no. 2, pp. 528–534.
12. Ibrahim K.Z., Hofmeyr S., Iancu C., Roman E. Optimized pre-copy live migration for memory intensive applications. *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis*. Seattle, USA, 2011, art. 40. doi: 10.1145/2063384.2063437
13. Ye K., Jiang X., Huang D., Chen J., Wang B. Live migration of multiple virtual machines with resource reservation in cloud computing environments. *Proc. 4<sup>th</sup> Int. Conf. on Cloud Computing, CLOUD 2011*. Washington, USA, 2011, pp. 267–274. doi: 10.1109/CLOUD.2011.69
14. Malik V, Barde C.R. Live migration of virtual machines in cloud environment using prediction of CPU usage. *International Journal of Computer Applications*, 2015, vol.

- N 23. P. 1–5. doi: 10.5120/20691-3604
15. Bogatyrev V.A. Fault tolerance of clusters configurations with direct connection of storage devices // *Automatic Control and Computer Sciences*. 2011. V. 45. N 6. P. 330–337. doi: 10.3103/S0146411611060046
  16. Bogatyrev V.A. An interval signal method of dynamic interrupt handling with load balancing // *Automatic Control and Computer Sciences*. 2000. V. 34. N 6. P. 51–57.
  17. Богатырев В.А., Богатырев С.В. Резервированное обслуживание в кластерах с уничтожением неактуальных запросов // *Вестник компьютерных и информационных технологий*. 2017. № 1. С. 21–28. doi: 10.14489/vkit.2017.01.pp.021-028
  18. Богатырев В.А., Богатырев А.В. Модель резервированного обслуживания запросов реального времени в компьютерном кластере // *Информационные технологии*. 2016. Т. 22. № 5. С. 348–355.
  19. Богатырев В.А. Богатырев А.В. Надежность функционирования кластерных систем реального времени с фрагментацией и резервированным обслуживанием запросов // *Информационные технологии*. 2016. Т. 22. № 6. С. 409–416.
  20. Богатырев В.А., Богатырев С.В. Надежность мультикластерных систем с перераспределением потоков запросов // *Известия высших учебных заведений. Приборостроение*. 2017. Т. 60. № 2. С. 171–177. doi: 10.17586/0021-3454-2017-60-2-171-177
  21. Bogatyrev V.A. Protocols for dynamic distribution of requests through a bus with variable logic ring for reception authority transfer // *Automatic Control and Computer Sciences*. 1999. V. 33. N 1. P. 57–63.
  22. Богатырев В.А., Богатырев А.В., Богатырев С.В. Перераспределение запросов между вычислительными кластерами при их деградации // *Известия высших учебных заведений. Приборостроение*. 2014. Т. 57. № 9. С. 54–58.
  23. Богатырев В.А., Богатырев А.В., Голубев И.Ю., Богатырев С.В. Оптимизация распределения запросов между кластерами отказоустойчивой вычислительной системы // *Научно-технический вестник информационных технологий, механики и оптики*. 2013. № 3 (85). С. 77–82.
  - 117, no. 23, pp. 1–5. doi: 10.5120/20691-3604
  15. Bogatyrev V.A. Fault tolerance of clusters configurations with direct connection of storage devices. *Automatic Control and Computer Sciences*, 2011, vol. 45, no. 6, pp. 330–337. doi: 10.3103/S0146411611060046
  16. Bogatyrev V.A. An interval signal method of dynamic interrupt handling with load balancing. *Automatic Control and Computer Sciences*, 2000, vol. 34, no. 6, pp. 51–57.
  17. Bogatyrev V.A., Bogatyrev S.V. Redundant service clusters with the destruction of irrelevant queries. *Herald of Computer and Information Technologies*, 2017, no. 1, pp. 21–28. (In Russian) doi: 10.14489/vkit.2017.01.pp.021-028
  18. Bogatyrev V.A., Bogatyrev A.V. The model of redundant service requests real-time in a computer cluster. *Informacionnye Tehnologii*, 2016, vol. 22, no. 5, pp. 348–355. (In Russian)
  19. Bogatyrev V.A., Bogatyrev A.V. The reliability of the cluster real-time systems with fragmentation and redundant service requests. *Informacionnye Tehnologii*, 2016, vol. 22, no. 6, pp. 409–416. (In Russian)
  20. Bogatyrev V.A., Bogatyrev S.V. Reliability of multi-cluster systems with redistribution of the flow of requests. *Journal of Instrument Engineering*, 2017, vol. 60, no. 2, pp. 171–177. (In Russian) doi: 10.17586/0021-3454-2017-60-2-171-177
  21. Bogatyrev V.A. Protocols for dynamic distribution of requests through a bus with variable logic ring for reception authority transfer. *Automatic Control and Computer Sciences*, 1999, vol. 33, no. 1, pp. 57–63.
  22. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. Requests redistribution between computing clusters under degradation. *Journal of Instrument Engineering*, 2014, vol. 57, no. 9, pp. 54–58. (In Russian)
  23. Bogatyrev V.A., Bogatyrev A.V., Golubev I.Yu., Bogatyrev S.V. Queries distribution optimization between clusters of fault-tolerant computing system. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2013, no. 3, pp. 77–82. (In Russian)

#### Авторы

**Алексанков Сергей Михайлович** – аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация; инженер, АО «НИИ «Масштаб», Санкт-Петербург, 194100, Российская Федерация, alexsankov.sergey@gmail.com

#### Authors

**Sergey M. Aleksankov** – postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation; engineer, R&D Institute "Mashtab", Saint Petersburg, 194100, Russian Federation, alexsankov.sergey@gmail.com