УДК 004.942 **DOI** 10.15622/ia.24.3.7

В.А. Богатырев, В. Фунг

ПРИБЛИЖЕННАЯ ОЦЕНКА ЗАДЕРЖЕК В КОМПЬЮТЕРНОЙ СИСТЕМЕ С КОНТЕЙНЕРНОЙ ВИРТУАЛИЗАЦИЕЙ

Богатырев В.А., Фунг В. **Приближенная оценка задержек в компьютерной системе с контейнерной виртуализацией.**

Аннотация. Ключевую роль в достижении высокой надежности, безопасности, отказоустойчивости и малых задержек обслуживания запросов в распределенных системах (в том числе облачных вычислений) играет консолидация ресурсов обработки и хранения данных в кластерах, эффективность которых повышается при использовании технологий виртуальных машин и контейнерной виртуализации. Сложность построения моделей массового обслуживания систем контейнерной виртуализации вызвана тем, что интенсивность выполнения запросов в каждом контейнере связана с динамическим разделением общих ресурсов между активными (выполняющими функциональные задачи) контейнерами и издержками на поддержку всех развернутых в виртуальной машине контейнеров, в том числе неактивных, ожидающих направления в них запросов для обслуживания. Снижение интенсивности обслуживания в каждом контейнере из-за совместного использования общих ресурсов зависит от многих трудно исследуемых факторов. Для кластеров с контейнерной виртуализацией в данной статье предлагается приближенная граничная оценка среднего времени ожидания запросов и вероятности их своевременного обслуживания. При построении аналитической модели каждый контейнер представляется как отдельная одноканальная система массового обслуживания с бесконечной очередью и простейшим входным потоком. Основное отличие предлагаемой модели виртуального кластера заключается в граничной верхней, нижней и усредненной оценке возможного снижения интенсивности обслуживания в контейнерах из-за разделения между ними общих ограниченных вычислительных ресурсов узла кластера в зависимости от количества развернутых в нем контейнеров и изменяющегося числа активных контейнеров, зависящего от интенсивности входного потока. Показано существование оптимального числа развернутых в узлах контейнеров, при котором среднее время пребывания запросов в системе минимально, либо вероятность выполнения запросов за заданное время максимальна. Предлагаемые модели могут быть применены при структурно-параметрической оптимизации кластеров с конвейерной виртуализацией, в том числе в случае масштабирования и реконфигурации, адаптивной к и изменениям трафика, путем отключения или подключения части развернутых контейнеров в зависимости от изменений нагрузки в системе.

Ключевые слова: контейнер, контейнерная виртуализация, кластер, разделение ресурсов, задержка.

1. Введение. В настоящее время для распределенных компьютерных систем, в том числе облачных вычислений, все более критичными становятся требования их безопасности, надежности, отказоустойчивости [1 – 4] и производительности [5]. Ключевую роль в достижении этих требований играет консолидация ресурсов обработки и хранения данных в кластерах, эффективность которых

повышается при использовании технологий виртуальных машин и контейнеров.

Виртуальные машины (ВМ) [6 – 8] эмулируют полноценные компьютеры в виртуальной среде, при этом каждая ВМ включает в себя собственные выделяемые аппаратные средства, такие как процессор, память, сетевые интерфейсы и дисковое пространство. ВМ могут применяться для запуска различных операционных систем и приложений на одном физическом сервере. Это позволяет эффективно использовать вычислительные ресурсы и обеспечивает логическую изоляцию ВМ, что способствует повышению информационной безопасности компьютерной системы [9].

Применение виртуальных машин в облачных системах обеспечивает максимальную эффективность использования ресурсов. Платформы, такие как VMWare, OpenStack, Proxmox и другие, позволяют поставщикам облачных услуг предоставлять клиентам независимые виртуальные машины, адаптируемые под их конкретные потребности и требуемые ресурсы. Вместе с тем клиенты могут разворачивать свои приложения в контейнерах, работающих на физических серверах или виртуальных машинах, предоставляемых облачными поставщиками.

Технология виртуальных контейнеров стимулировала развитие микросервисной архитектуры, при которой приложения разделяются на небольшие, независимые сервисы, взаимодействующие через хорошо определённые программные интерфейсы приложений (API – Application Programming Interface). Такой подход потенциально позволяет повысить устойчивость функционирования к отказам системы, так как отказ (сбой) одного или нескольких микросервисов может не приводить к выходу из строя всего приложения, хотя в ряде случаев может вызвать частичную потерю функциональности. Использование микросервисной архитектуры способствует простоте реконфигурации, масштабирования и модернизации системы.

Технологии виртуализации, предоставляют мощные инструменты для обеспечения как надежности, так и информационной безопасности компьютерной системы, при этом в результате логической изоляции приложений и сервисов в виртуальных средах риск воздействия одного компонента на другой может быть минимизирован. Таким образом, виртуализация позволяет ускорить восстановление работоспособности системы в случае отказов, сбоев или злонамеренных воздействий при атаках.

Изоляция у контейнеров менее строгая, чем у виртуальных машин, что требует применения дополнительных мер для обеспечения

их безопасности. Повышению безопасности способствует использование минимальных базовых образов при контейнерной виртуализации, а также регулярное обновление операционной системы.

Реконфигурация отказоустойчивых систем при виртуализации связана с необходимостью репликации и миграции виртуальных машин, а также развертывания реплик контейнеров при необходимости [10, 11]. Построение отказоустойчивых кластеров (включая кластеры с контейнерной виртуализацией), требует обоснования и оптимизации решений обеспечения надежности, производительности кластера и высокоскоростного надежного доступа к нему через коммуникационную среду, которая используется также для миграции виртуальных машин при реконфигурации системы [12 – 14].

Важными составляющими обоснования выбора структуры, функциональной организации и правил эксплуатации кластера является оптимизация дисциплин: диспетчеризации и распределения запросов при балансировке нагрузки; физического и информационного восстановления системы после отказов и сбоев; обеспечения непрерывности вычислений и устойчивости к потере данных. При этом отметим взаимозависимости перечисленных дисциплин, требующих оптимизации. Оптимизация кластера требует построения взаимосвязанных моделей его надежности и обслуживания трафика.

Выбор решений по оптимизации кластера с контейнерной виртуализацией осложняется необходимостью учета при моделировании динамического разделения и совместного использования ограниченных вычислительных ресурсов системы между виртуальными машинами и контейнерами. Для отказоустойчивого функционирования требуется выделение общих ресурсов системы для процессов контроля и реконфигурации, включая формирование реплик и миграцию виртуальных машин и контейнеров при обнаружении отказов.

Адаптивная реконфигурация, предусматривающая создание дополнительных реплик контейнеров или их удаление, может быть выполнена в зависимости от изменения интенсивности трафика. При этом важно учитывать компромисс между задержками обслуживания запросов, производительностью, надежностью и энергопотреблением системы.

Эффективное решение задач реконфигурации должно базироваться на концепции модельно-ориентированного проектирования с использованием моделей надежности и массового

обслуживания. При построении отказоустойчивых кластеров необходимо учитывать взаимное влияние мероприятий и механизмов, направленных на обеспечение надежности и производительности, и находить компромисс при выборе проектных решений.

В кластерах, предусматривающих совместную работу множества виртуальных машин и контейнеров, важно понимать, как отказы отдельных компонентов могут влиять на общую производительность и задержки обслуживания потоков запросов.

На оценку задержек и других вероятностно-временных показателей качества функционирования компьютерных систем ориентирован аппарат теории массового обслуживания [15 – 17]. Модели обслуживания кластерных систем на базе виртуальных машин, размещаемых на физических узлах кластера, в том числе в облачной среде, предложены в работах [18 – 20]. Известные модели кластеров виртуальных машин учитывают балансировку нагрузки узлов (виртуальных машин), проводимую при диспетчеризации запросов. Модели, отражающие работу кластеров в нестационарных режимах обслуживания при периодических изменениях во времени трафика запросов, описаны в статьях [21 – 23]. Периодичность изменения трафика вызвана его повторявшимися колебаниями в течение суток, дней недели, месяца и года.

При проектировании компьютерных систем кластерной архитектуры важен анализ совокупного влияния отказов и дисциплин физического и информационного восстановления на надежность и задержки обслуживания. При обосновании выбора кластерных систем важна разработка критериев, учитывающих обеспечение высоких уровней надежности, производительности и доступности (готовности) системы [24, 25].

Для современных концепций построения отказоустойчивых кластеров на основе виртуальных машин с размещением в них контейнеров важной задачей является анализ задержек обслуживания поступающего потока запросов [26], с учетом особенностей контейнерной виртуализации, заключающейся в автоматическом разделении ограниченных ресурсов виртуальной машины между активными контейнерами. Модели кластеров [18 – 26] не учитывают особенностей контейнерной виртуализации.

Модели оценки задержек систем с контейнерной виртуализацией для однородного потока запросов без учета задержек их распределения по узлам и балансировки нагрузки рассматривались в [27 – 29]. В работе [30] предложена модель оценки задержек в кластере виртуальных машин, компонуемых контейнерами с учетом

двух этапов распределения запросов. Модель [30] учитывает задержки в очереди балансировщика нагрузки кластера при распределении запросов между узлами кластера, каждый из которых содержит группу виртуальных машин. Модель [30] учитывает также задержки в очередях для каждой группы виртуальных машин, компонуемых контейнерами. На основе предложенной в [30] аналитической модели проанализировано влияние числа развернутых в виртуальной машине контейнеров на время отклика системы по выполнению запросов, однако влияние соотношения числа развернутых и активных контейнеров при этом не учитывается (не установлено). Результаты статьи [30] могут быть использованы при обосновании выбора конфигурации кластера, включая определение оптимального числа разворачиваемых в узлах контейнеров. Ограниченность модели СМО работы [30], в которой рассматривается кластер, состоящий из нескольких серверов, на каждом из которых развернуты несколько контейнеров, обусловлена предположением независимости каждого канала обслуживания, представляющего отдельный контейнер. Динамическое изменение интенсивности обслуживания в разных каналах в зависимости от доли загруженных (активных) каналов при этом не учитывалось.

Влияние числа разворачиваемых в узлах кластера контейнеров на среднее время ожидания и устойчивость облачных вычислений к DDoS атакам исследовано в [9]. Результаты статьи [9], подтвержденные аналитическим моделированием и экспериментальными исследованиями, могут быть применены при динамической реконфигурации систем в зависимости от интенсивности DDoS атак, направленных на нарушение стационарности обслуживания потока легальных запросов и их потерю.

Ограниченность существующих моделей кластеров с контейнерной виртуализацией обусловлена недостаточностью исследований влияния на задержки обслуживания запросов динамического разделения ограниченных вычислительных ресурсов между простаивающими контейнерами и контейнерами, выполняющими функциональные запросы. Такая особенность совместной работы виртуальных контейнеров вызывает усложнение моделей массового обслуживания, так как интенсивность выполнения запросов в каждом контейнере связана с динамическим разделением общих ресурсов между активными (выполняющими функциональные задачи) контейнерами с учетом издержек на поддержку всех развернутых в виртуальной машине контейнеров (активных и

простаивающих). Снижение интенсивности обслуживания в каждом контейнере из-за совместного использования общих ресурсов зависит от многих трудно исследуемых факторов, в том числе от частоты и длительности обращения контейнеров во время обслуживания запроса к тем или иным ресурсам процессоров и памяти. Разделение общих вычислительных ресурсов зависит от наличия интервалов простоев при таких обращениях, во время которых общие ресурсы потенциально могут использоваться другими активными контейнерами. Такое адаптивное разделение общих ресурсов несомненно способствует эффективности их совместного использования и минимизации простоев в результате предоставления ресурсов для работы других активных контейнеров. Учет перечисленных факторов совместного использования общих ресурсов приводит к трудоемкости построения аналитической модели обслуживания кластерных систем с контейнерной виртуализацией.

При построении аналитической модели массового обслуживания систем с контейнерной виртуализацией в статье [31] предлагается использовать экспериментально установленную зависимость снижения интенсивности обслуживания отдельных контейнеров от числа развернутых в виртуальной машине контейнеров и их активной части, которая изменяется в зависимости от интенсивности потока запросов. Эксперименты, описанные в [31], подтверждают существенное влияние разделения ограниченных вычислительных ресурсов виртуальной машины между активными и развернутыми контейнерами.

При сочетании экспериментальных исследований и аналитического моделирования в статье [31] дана оценка вероятностно-временных показателей качества обслуживания запросов в системах контейнерной виртуализации. В статьях [31, 32] узлы кластера представлены как многоканальные СМО с общей очередью (конечной для [32] и бесконечной для [31]). Показано существование оптимального значения числа разворачиваемых контейнеров в узле кластера, а также сформулирована задача оптимизации этого числа, направленная на снижение задержек обслуживания запросов в таких системах.

Ограничения, предлагаемого в [31, 32] подхода к построению аналитической модели обслуживания кластера с контейнерной виртуализацией, заключается в необходимости проведения достаточно трудоемкого эксперимента в зависимости от изменений среднего времени выполнения запросов для каждого значения числа разворачиваемых в узлах кластера виртуальных контейнеров и их

активного числа, зависящего от интенсивности трафика. Многоканальные СМО [31, 32] ориентированы на системы, в которых к развернутым в каждом узле контейнерам организуется общая очередь. Эти модели не отражают функционирование кластеров с контейнерной виртуализацией, в которых к каждому контейнеру организуется отдельная очередь, а кластер представляется группой одноканальных СМО. Построение таких моделей затруднено из-за того, что разделение ограниченных общих ресурсов кластера между контейнерами приводит к необходимости учета зависимости облуживания очередей разных СМО, представляющих различные контейнеры. Сложность установления зависимости интенсивности обслуживания в разных одноканальных СМО обусловлено разделением общих ресурсов между разными СМО в зависимости от соотношения числа развернутых и активных контейнеров, меняющемся при изменениях интенсивности входного потока.

Для упрощения моделей кластеров с контейнерной виртуализацией при организации отдельных очередей к каждому контейнеру в данной статье предлагается приближенная граничная оценка среднего времени ожидания запросов и вероятности их своевременного обслуживания.

Построение предлагаемой приближенной аналитической модели предполагает экспериментальное определение зависимости интенсивности обслуживания в отдельных контейнерах от изменяемого числа разворачиваемых контейнеров в узле кластера и их числа, участвующих в выполнении функциональных запросов (активных контейнеров). Число активных контейнеров зависит от интенсивности трафика запросов. Экспериментально полученные данные о интенсивности обслуживания в контейнере с учетом динамического разделения ресурсов узла на обслуживание запросов в активных контейнерах предполагается использовать для верхней, нижней и усредненной оценки загрузки контейнеров. Загрузка контейнеров определяется в зависимости от интенсивности входного потока запросов, распределяемых балансировщиком в узел кластера. Результаты аналитической оценки загрузки контейнеров используются для оценки математического ожидания числа контейнеров, задействованных в обработке трафика и математического ожидания интенсивности обслуживания запросов в контейнерах. При определении математического ожидания интенсивности обслуживания запросов в контейнерах используются результаты экспериментов, что позволяет учесть влияние динамического разделения общедоступных ресурсов узла между контейнерами, участвующими и не

участвующими в обслуживании запросов. Полученная приближенная оценка интенсивности обслуживания в контейнерах в дальнейшем используется для вычисления вероятности своевременного обслуживания запросов и их средней задержки в узлах кластера в зависимости от интенсивности поступающих запросов.

Таким образом целью статьи является исследование возможностей повышения эффективности кластерных систем конвейерной виртуализации на основе предлагаемых приближенных граничных моделей для оценок среднего времени пребывания запросов в кластере и вероятности их своевременного обслуживания за время, не превышающее предельно допустимое значение ожидания в очереди.

В качестве целевого критерия эффективности для систем, не требующих выдачи результатов в директивные сроки, используется среднее время пребывания (ожидания) запросов в системе. Для систем с ограниченным допустимым временем ожидания (системы реального времени) в качестве целевой функции используется максимизация вероятности выполнения запросов за время меньшее предельно допустимого.

Оптимальные масштабирование и реконфигурация узлов кластера при изменениях входных потоков реализуется путем отключения или подключения части развернутых в виртуальных машинах контейнеров в зависимости от изменений нагрузки. Отключение части развернутых контейнеров позволяет уменьшить не только энергопотребление кластера, но и снизить загрузку общих ресурсов на поддержку функционирования неактивных контейнеров.

2. Рассматриваемые варианты организации системы с контейнерной виртуализацией. Рассмотрим варианты построения компьютерных систем на базе виртуальных машин с контейнерной виртуализацией. Простейшая организация таких систем, представленная на рисунке 1, включает виртуальную машину с развернутыми в ней n контейнерами. В простейшем случае контейнеры считаются функционально и параметрически одинаковыми, при этом к каждому контейнеру формируется отдельная очередь запросов. Виртуальные машины (рисунок 1), при необходимости, могут объединяться в кластеры на базе одного или нескольких физических узлов (серверов) кластера.

Организация кластера, состоящего из N независимых узлов, реализованных в виде виртуальных машин, каждая из которых содержит n контейнеров, в предположении отсутствия образования очередей в балансировщике нагрузки приведена на рисунке 2, а с

учетом возможности формирования очереди при распределении запросов на рисунке 3.

Особенностью моделируемой системы является то, что интенсивность обслуживания в каждом контейнере изменяется в зависимости от количества развернутых контейнеров и меняющегося числа активных контейнеров, что предполагает зависимость обслуживания запросов в разных контейнерах узла.

На рисунках 1-3 рассматриваемых вариантов систем обозначены: интенсивность входного потока запросов λ, поступающего в каждый узел кластера (виртуальную машину), а также интенсивности входного и выходного потока всего кластера (которые при отсутствии потери запросов в системе совпадают). Следует подчеркнуть, что представленные одноканальные СМО, отличаются от классических, в которых интенсивность обслуживания в каждой СМО не меняется. В системах с виртуальный контейнеризацией общие ресурсы узла динамически разделяются между контейнерами, причем как активными, так и пассивными. Для исследования этого распределения между активными и пассивными контейнерами на интенсивность обслуживания в них запросов в следующем разделе описаны проведенные эксперименты.

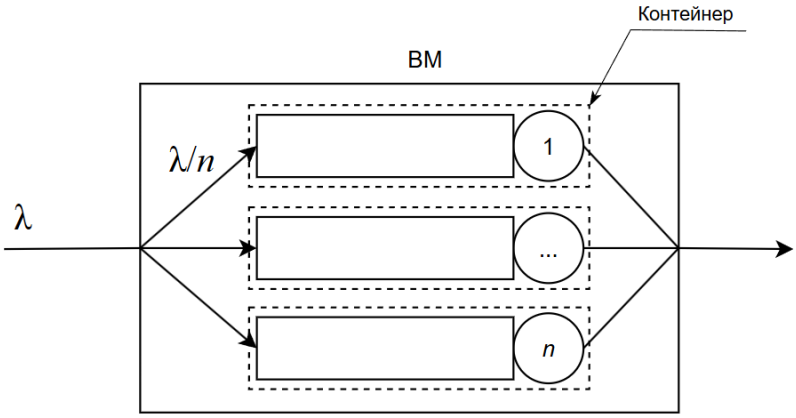


Рис. 1. Организация виртуального компьютерного кластера

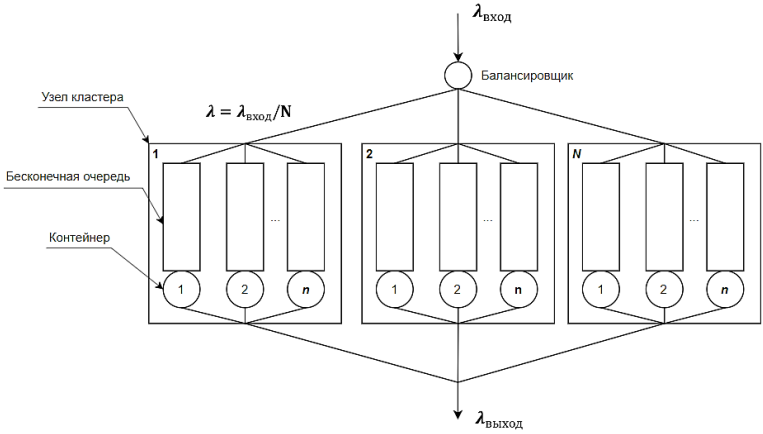


Рис. 2. Организация кластера *N* виртуальных машин с контейнеризацией без формирования очереди в балансировщике нагрузки

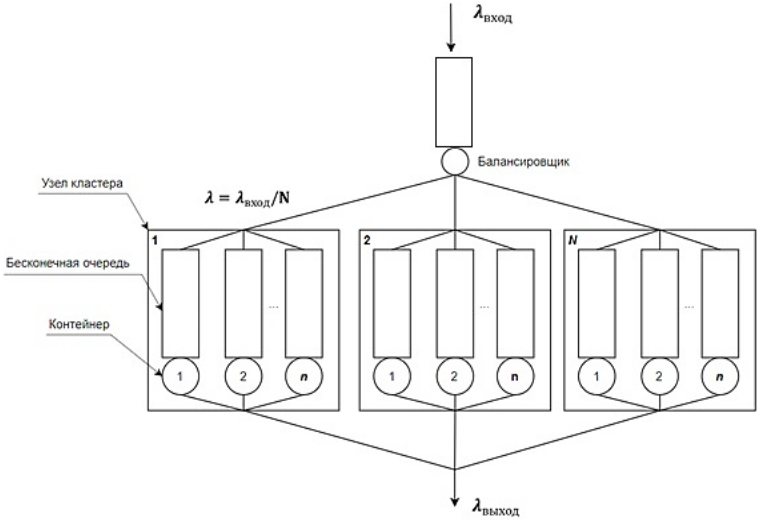


Рис. 3. Организация кластера виртуальных машин с контейнеризацией при формировании очереди в балансировщике нагрузки

3. Экспериментальное исследование влияния числа активных и общего числа контейнеров в узлах на интенсивность обслуживания в контейнерах. Эксперимент [30, 31] проведен на кластере (рисунок 4), в котором узлы представлены изолированными

виртуальными машинами Proxmox с конфигурацией 1 vCPU и 2 ГБ оперативной памяти, работающими на хосте с 4 процессорами Intel(R) Core(TM) i5-4570 @ 3,20 ГГц. Однопоточный веб-сервер, запакованный в каждый контейнер, написан на языке программирования Python и развёрнут на узлах кластера с использованием платформы k3 s.

Программа внутри веб-сервера включает в себя асинхронную отправку запросов в базу данных и выполнение простых операций умножения матриц в нескольких циклах. Размер кеша составляет 50 МБ.

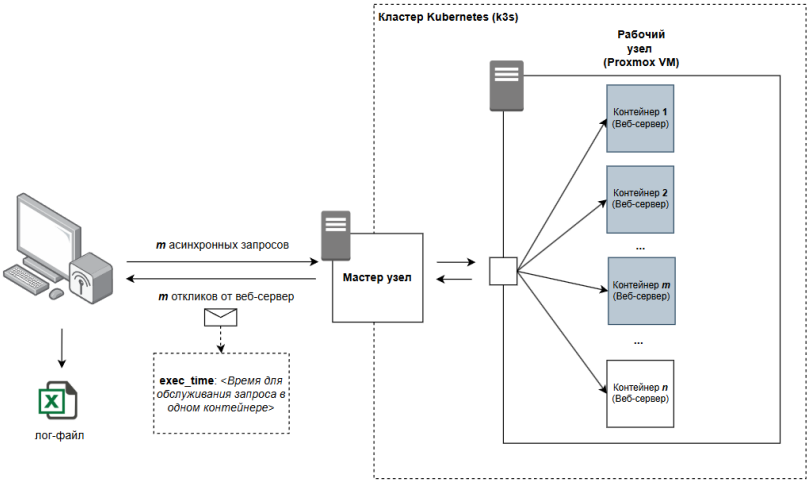


Рис. 4. Конфигурация системы для реализации эксперимента

В результате эксперимента установлена зависимость интенсивности обслуживания запросов в контейнере $\mu(n, m)$ от числа развернутых контейнеров n и от числа активных контейнеров m (выполняющих обслуживание запросов).

Интенсивности обслуживания, полученные в результате эксперимента, отражены матрицей **M**. Строки матрицы соответствуют числу развернутых контейнеров n (n=1, 2,…, n_0, где n_0 максимальное число контейнеров, предусмотренных к разворачиванию в узле), а столбцы числу активных контейнеров m (m=1 ,2,…, n). Таким образом, элемент матрицы **M**, расположенный в n-й строке и m-м столбце представляет интенсивность обслуживания запросов $\mu(n, m)$ в контейнере, когда в нем развернуто n контейнеров и m из них активно.

Пример вида матрицы **M** представлен ниже:

$$\begin{pmatrix}
22.3056 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.2484 & 14.9688 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.2421 & 13.4269 & 11.7887 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.1315 & 12.8368 & 10.4739 & 9.6111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.1379 & 12.4521 & 9.8015 & 8.6678 & 7.8260 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.0971 & 12.2769 & 9.4101 & 8.0412 & 7.2940 & 6.8094 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.0736 & 12.0474 & 8.9995 & 7.7675 & 6.7622 & 6.3337 & 5.9818 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.9897 & 11.9974 & 8.9397 & 7.4726 & 6.6169 & 5.8908 & 5.5987 & 5.2086 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22.0303 & 11.8065 & 8.5902 & 7.1756 & 6.1438 & 5.7449 & 5.3634 & 4.9734 & 4.8870 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.9299 & 11.6579 & 8.6411 & 6.9367 & 6.1595 & 5.5669 & 5.0955 & 4.7114 & 4.4394 & 4.2970 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.7162 & 11.5665 & 8.3528 & 6.7052 & 5.9114 & 5.2396 & 4.8631 & 4.4923 & 4.2513 & 4.0792 & 3.9427 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.7159 & 11.5296 & 8.2876 & 6.7299 & 5.7047 & 5.1878 & 4.6436 & 4.3049 & 4.0989 & 3.8672 & 3.6705 & 3.5818 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.7351 & 11.4611 & 8.2649 & 6.5284 & 5.7352 & 5.0660 & 4.4858 & 4.2710 & 3.9047 & 3.8288 & 3.7296 & 3.5496 & 3.4161 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.7273 & 11.4523 & 8.1945 & 6.6133 & 5.5663 & 4.8881 & 4.4374 & 4.0664 & 3.9208 & 3.6204 & 3.5038 & 3.3795 & 3.3012 & 3.0541 & 0 & 0 & 0 & 0 & 0 & 0 \\
21.6894 & 11.4576 & 8.0965 & 6.4295 & 5.5191 & 4.7594 & 4.3047 & 4.0417 & 3.7328 & 3.4820 & 3.3966 & 3.2322 & 3.2267 & 2.9882 & 2.9335 & 0 & 0 & 0 & 0 & 0 \\
21.6895 & 11.4234 & 8.0055 & 6.2783 & 5.4427 & 4.7469 & 4.2578 & 3.9279 & 3.7281 & 3.4541 & 3.2929 & 3.1300 & 3.0727 & 2.8806 & 2.8141 & 2.7312 & 0 & 0 & 0 & 0 \\
21.6282 & 11.3349 & 7.9741 & 6.3664 & 5.3621 & 4.5994 & 4.1624 & 3.8823 & 3.6295 & 3.3805 & 3.2936 & 3.1243 & 2.9631 & 2.7504 & 2.7581 & 2.6261 & 2.6605 & 0 & 0 & 0 \\
21.6281 & 11.3030 & 7.9727 & 6.2405 & 5.3050 & 4.5602 & 4.2300 & 3.7810 & 3.5101 & 3.3191 & 3.1837 & 3.0321 & 2.8750 & 2.7859 & 2.7066 & 2.6156 & 2.5579 & 2.4548 & 0 & 0 \\
21.6513 & 11.3169 & 7.9507 & 6.3076 & 5.3560 & 4.5686 & 4.1681 & 3.6999 & 3.4874 & 3.2733 & 3.0575 & 2.8461 & 2.7518 & 2.6416 & 2.5834 & 2.5033 & 2.4001 & 2.3939 & 2.3093 & 0 \\
21.6063 & 11.1883 & 7.9095 & 6.1469 & 5.2033 & 4.5864 & 4.0560 & 3.7413 & 3.4260 & 3.1818 & 2.9930 & 2.8750 & 2.7654 & 2.6354 & 2.6127 & 2.4944 & 2.4573 & 2.3296 & 2.2657 & 2.2206
\end{pmatrix}$$

Результаты эксперимента по оценке влияния количества общих и активных контейнеров на интенсивность обработки запросов контейнером и на измеряемую загрузку процессора узла представлены на рисунке 5(а, б). На рисунке 5(а) (слева) представлены результаты измерений интенсивности обслуживания запросов в контейнерах, а на рисунке 5(б) (справа) – результаты измерения загрузки процессора узла кластера.

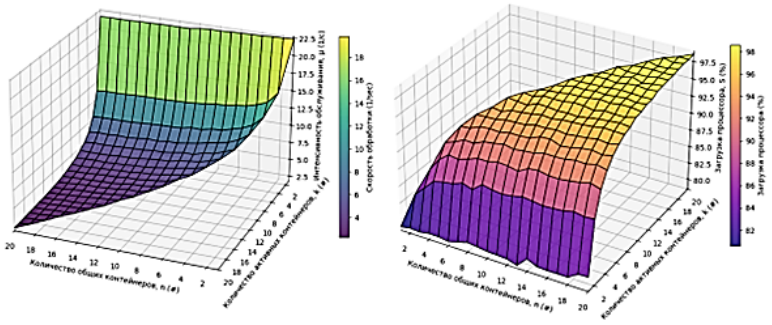
Результаты экспериментов показывают, что снижение интенсивности обработки запросов каждым контейнером, зависящее от числа активных контейнеров, нелинейно. Такая зависимость имеет место предположительно потому, что во время обработки запроса контейнеры потенциально не всегда используют ресурсы процессора. При этом ресурсы процессора могут быть задействованы для обслуживания запросов другими активными контейнерами.

Из анализа первого столбца таблицы заметно влияние затрат процессорных ресурсов на поддержку функционирования в режиме ожидания неактивных контейнеров. Анализ строк матрицы **M** показывает снижение интенсивности обработки запросов каждым активным контейнером в зависимости от числа активных контейнеров в узле. При этом видно, что анализируемая интенсивность обслуживания не снижается прямо пропорционально числу активных контейнеров. Вследствие динамического распределения ресурсов между активными контейнерами снижение рассматриваемой интенсивности обслуживания от числа активных контейнеров оказывается значительно меньшим.

Построение аналитической модели массового обслуживания для систем, предусматривающих организацию очередей к каждому контейнеру, связано с определенными трудностями. Поэтому в данной статье предлагается приближённая граничная (верхняя и нижняя)

и средняя оценка задержек в рассматриваемых системах контейнерной виртуализации.

Следует заметить, что некоторые результаты измерений, могут зависеть от конкретной реализации используемых вычислительных средств и программного обеспечения, а это в отдельных случаях может ограничивать применение модели в новых средах с отличающимися характеристиками. Таким образом, проведение экспериментов для повышения достоверности модели необходимо реализовать с применением конкретных реализаций узлов, которые предусмотрено использовать при проектировании кластера.



а) б)

Рис. 5. Результаты эксперимента по оценке влияния количества общих и активных контейнеров: а) на интенсивность обработки запросов контейнером; б) на загрузку процессора узла кластера

4. Приближенная модель оценки задержек обслуживания в компьютерной системе с контейнерной виртуализацией. При построении модели обслуживания рассматриваемых компьютерных систем предполагается, что узлы кластера функционируют в стационарном режиме. Поток запросов считается простейшим, а время обслуживания запросов – экспоненциальным. Обслуживание на разных узлах кластера (виртуальных машинах) предполагается независимым. Входной поток запросов с интенсивностью λ равномерно распределяется между контейнерами. Каждый контейнер моделируется как одноканальная система массового обслуживания с бесконечной очередью и простейшим входным потоком. Все контейнеры считаются однородными по функциональности и параметрам обслуживания. Предполагается, что запросы в очереди и во время обслуживания не теряются, а их репликация не осуществляется. Влияние отказов, сбоев и ошибок вычислений в данной модели не учитывается.

Для оценки задержек в узлах кластера (виртуальных машинах) предлагается использовать приближённые верхнюю, нижнюю и усреднённую оценки интенсивности обслуживания в каждом контейнере в зависимости от числа развернутых контейнеров и их активного числа, изменяющегося в процессе функционирования системы.

Построение приближённой модели проведём в соответствии со следующим алгоритмом:

1. Используя результаты эксперимента, представленного матрицей **M**, определим верхнее, нижнее и усреднённое значения интенсивности обслуживания запросов в контейнере.

2. По установленным (заданным) в пункте 1 значениям интенсивности обслуживания вычислим верхнюю, нижнюю и усреднённую оценки загрузки одного контейнера в зависимости от интенсивности входного потока запросов, равномерно распределяемого балансировщиком между узлами кластера и развернутыми в них контейнерами.

3. Определив загрузку контейнеров (вероятности их занятости), вычислим математическое ожидание числа активных контейнеров и вероятности различного числа активных контейнеров, обслуживающих функциональные запросы.

4. Оценим интенсивность обслуживания запросов в контейнерах, используя результаты расчётов вероятностей активности различного числа контейнеров m (m = 1, 2, 3, … , n), полученных на предыдущем этапе. Для этого умножим вычисленные вероятности на соответствующие значения интенсивности обслуживания $\mu(n, m)$, взятые из матрицы **M**, полученной экспериментально. Это позволит учесть влияние динамического разделения общедоступных ресурсов узла между активными и неактивными контейнерами.

5. Определив приближённое значение интенсивности обслуживания в контейнере с учётом влияния динамического разделения общедоступных ресурсов узла между активными и ожидающими запросов контейнерами, вычислим среднюю задержку запросов в очередях узлов кластера в зависимости от интенсивности поступающих в кластер запросов и числа контейнеров, развернутых в его узлах.

Аналогично (при повторении пунктов 1-4 алгоритма) может быть получена приближённая оценка вероятности своевременного обслуживания запросов в зависимости от интенсивности трафика и числа контейнеров, развернутых в узлах кластера. Оценка вероятности

своевременного обслуживания запросов проведена в разделе 5 данной статьи.

Рассмотрим построение модели обслуживания в системах с контейнерной виртуализацией более подробно.

Вначале рассмотрим обслуживание в узле кластера, укомплектованного одной виртуальной машиной с n развернутыми в ней контейнерами (рисунок 1).

При построении аналитической модели по п.1 приведённого алгоритма будем считать, что интенсивность обслуживания в контейнерах не зависит от изменений длины очереди (числа активных контейнеров). Загрузку контейнера при верхней, нижней и усреднённой оценке примем равной:

$$\rho = \frac{\lambda}{n \cdot \mu_g},$$

при этом μ_g – соответственно верхняя, нижняя либо усреднённая оценка интенсивности обслуживания запросов в контейнере, задаваемая в соответствии с пунктом 1 приведённого выше алгоритма.

Оценку интенсивности обслуживания μ_g можно получить с учётом результатов проведённых экспериментов. При этом возможны верхняя, нижняя и средняя оценки искомой загрузки контейнеров ρ.

При верхнем приближении загрузки контейнера ρ примем $\mu_g=\mu(n, n)$, при нижнем $\mu_g=\mu(n,1)$, а при усреднённом $\mu_g = \sum_{i=1}^{n} \mu(n,i) / n$. При этом значения $\mu(n,1)$, $\mu(n,n)$ и $\mu(n,i)$ берутся с учётом экспериментально установленных результатов, представленных в матрице **M**. Погрешность усредненной оценки μ_g обусловлена тем, что все состояния с разным числом активных контейнеров считаются равно вероятными.

Таким образом, загрузка контейнера при верхней, нижней и усредненной оценке задается как:

$$\rho = \begin{cases} \lambda / n\mu(n,1), \textit{при нижней оценке}, \\ \lambda / n\mu(n,n), \textit{при верхней оценке}, \\ \lambda \Big/ \sum_{i=1}^{n} \mu(n,i), \textit{при усредненной оценке}. \end{cases} \quad (1)$$

Упрощение модели обусловлено тем, что предполагается неизменность интенсивности обслуживания μ_g, а также загрузки контейнеров ρ, вне зависимости от текущей длины очереди.

В момент поступления запроса в некоторый контейнер среди остальных $n-1$ контейнеров могут быть заняты (активны) $i=0, 1, \ldots, n-1$ контейнеров. Вероятность того, что в рассматриваемый момент времени поступления запроса активно i контейнеров, оценим как:

$$\rho_i = C_{n-1}^{i} \cdot \rho^i \cdot (1-\rho)^{n-i-1}, \tag{2}$$

где ρ оценивается по формуле (1).

Далее определим интенсивность обслуживания запросов в контейнерах с учетом вероятности активности различного числа контейнеров. При этом вероятность активности некоторого контейнера зададим через ранее определенное значение его загрузки ρ, которое вычисляется по формуле (2) при условии неизменности интенсивности обслуживания в контейнерах вне зависимости от их активного числа.

Интенсивность обслуживания в контейнере с учетом вероятности активности i контейнеров оценим следующим образом:

$$\mu = \sum_{i=0}^{n-1} \mu(n, i+1) \cdot C_{n-1}^{i} \cdot \rho^i (1-\rho)^{n-i-1}, \tag{3}$$

При этом $\mu(n, i+1)$ – интенсивность обслуживания запросов в контейнере при активности i контейнеров и развертывании n контейнеров на узле. Значение $\mu(n, i+1)$ определяется по матрице **M**, сформированной в результате описанного выше эксперимента.

Верхнюю, нижнюю и усредненную оценки математического ожидания интенсивности обслуживания в контейнере μ получаем соответственно при верхней μ_{upper}, нижней μ_{lower} и средней μ_{avg} оценках загрузки контейнеров ρ.

Результаты оценки интенсивности обслуживания в одном контейнере в зависимости от числа развернутых в виртуальной машине контейнеров для трех рассматриваемых приближений приведены на рисунке 6. Расчёт выполнен при интенсивности запросов $\lambda=19$ с$^{-1}$.

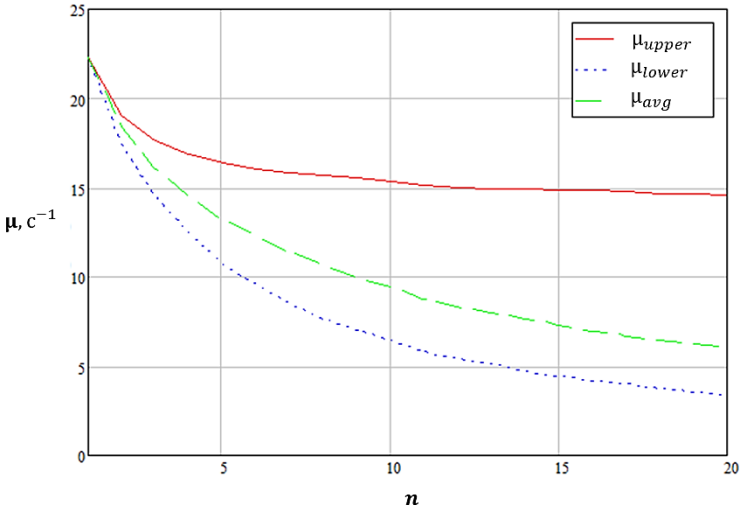
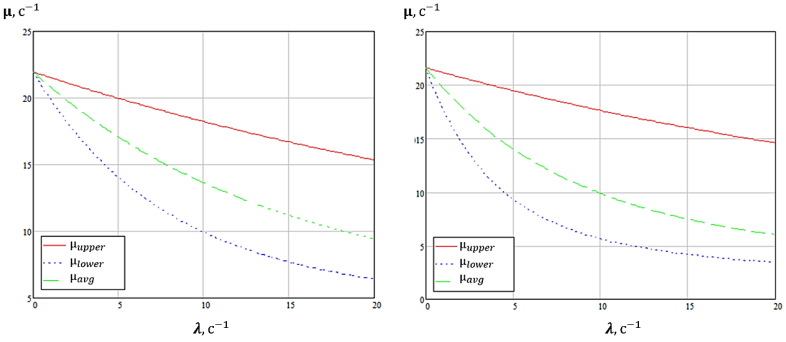


Рис. 6. Оценка интенсивности обслуживания контейнером в зависимости от числа развернутых контейнеров n, результаты получены при верхней, нижней и средней оценке загруженности контейнеров

Рисунок 6 показывает, что интенсивность обслуживания в одном контейнере уменьшается с увеличением количества развернутых в узле контейнеров n.

Зависимость интенсивности обслуживания в контейнере от интенсивности запросов λ при n=20 (слева) и n=10 (справа) приведена на рисунке 7(а) и 7(б) соответственно.



а б

Рис. 7. Оценка интенсивности обслуживания в контейнере в зависимости от интенсивности запросов λ при n=20 (а – слева) и n=10 (б – справа)

На рисунке 7 наблюдается, что интенсивность обслуживания в контейнере уменьшается при увеличении интенсивности запросов. Это объясняется тем, что с увеличением интенсивности запросов вероятность активации большего числа контейнеров также увеличивается, что приводит к уменьшению интенсивности обслуживания в каждом контейнере.

Верхнюю, нижнюю и среднюю оценки математического ожидания времени пребывания запросов T в узле кластера (в контейнере) вычислим как:

$$T_{lower} = \frac{\frac{1}{\mu_{upper}}}{1-\frac{\lambda}{n\cdot\mu_{upper}}},\ T_{upper} = \frac{\frac{1}{\mu_{lower}}}{1-\frac{\lambda}{n\cdot\mu_{lower}}},\ T_{avg} = \frac{\frac{1}{\mu_{avg}}}{1-\frac{\lambda}{n\cdot\mu_{avg}}}. \quad (4)$$

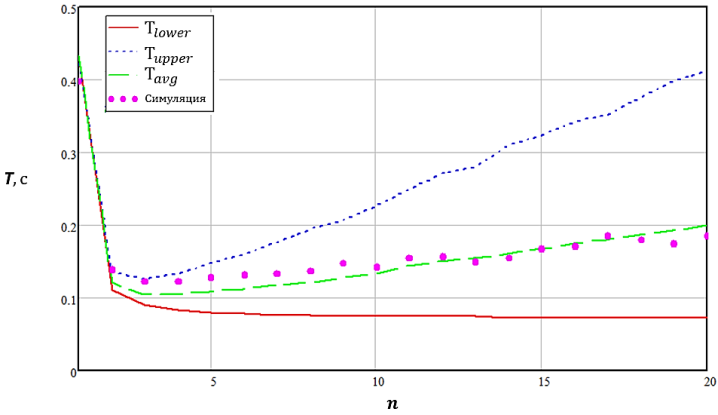


Рис. 8. Оценка среднего времени пребывания запросов в контейнере T в зависимости от числа развернутых в узле контейнеров n

Оценка среднего времени пребывания запросов в контейнере T от числа развернутых в узле контейнеров по предлагаемой модели и при имитационном моделировании представлены на рисунке 8 при интенсивности запросов $\lambda = 19$ с$^{-1}$. Имитационное моделирование [33] проведено с использованием языка Python и библиотеки моделирования SimPy [34 – 36], при этом контейнеры моделируются как объект службы с использованием класса SimPy.Resource.

Программа имитационного моделирования включает два ключевых этапа:

1. Преобразование потоков запросов, характеризующихся интервалами между ними, распределенными по экспоненциальному закону случайных величин, среднее значение которых задается пользователем.

2. Обслуживание запросов, когда время выполнения каждого запроса распределено экспоненциально.

На втором этапе учитывается зависимость интенсивности обслуживания в контейнере (service_rate) от общего числа контейнеров n и количества активных контейнеров m с использованием функции $\mu(n, m)$, отображаемой в матрице **M**. Эта взаимосвязь может быть учтена с помощью функции SimPy.Resource.count, которая предоставляет информацию о количестве занятых процессов в определенный момент времени, что является индикатором количества активных контейнеров.

Графики на рисунке 8 показывают, что верхняя, нижняя и усреднённая оценки не противоречат результатам имитационного моделирования. Средняя абсолютная процентная ошибка (MAPE) времени пребывания запросов в контейнере T_{avg} относительно результатов имитационного моделирования составляет 7,4%. Таким образом, можно считать, что предложенная аналитическая модель адекватно описывает поведение системы.

Из графика на рисунке 8 видно, что существует оптимальное число развернутых в узлах контейнеров n, при котором среднее время пребывания запросов T в системе минимально. При этом следует отметить, что искомое значение оптимального числа развернутых в узлах контейнеров n при верхней, нижней и усреднённой оценках времени пребывания запросов в системе практически совпадает.

Таким образом, из представленных зависимостей видно, что при проектировании систем контейнерной виртуализации предложенная аналитическая модель может использоваться для поиска оптимального числа реплик контейнеров (n), разворачиваемых в виртуальной машине.

На рисунке 9 отображено среднее время пребывания запросов в системе при различных значениях интенсивности запросов λ.

Расчет математического ожидания времени пребывания запросов в узле от интенсивности трафика при различных значениях количества развернутых контейнеров $n=5, 10, 15, 20$ отражен на рисунке 10.

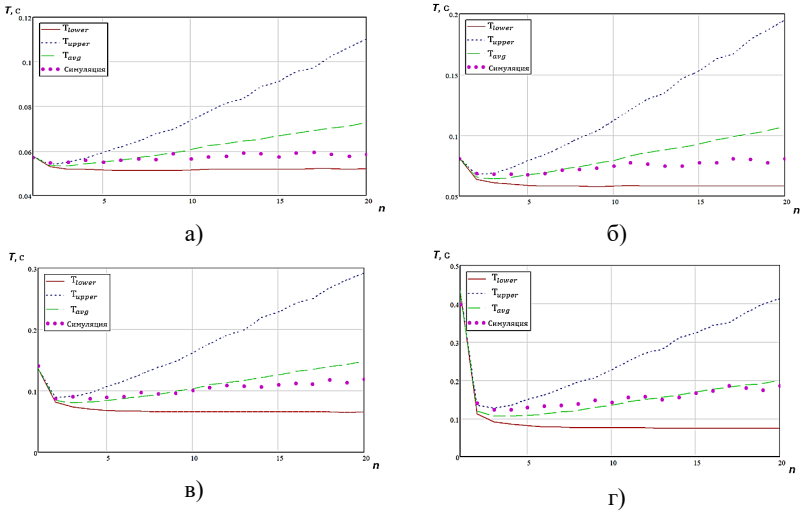


Рис. 9. Время пребывания запросов в системе в зависимости от числа развернутых контейнеров n при значениях интенсивности запросов: а) λ=5; б) 10; в)15; г) 20 c^{-1}

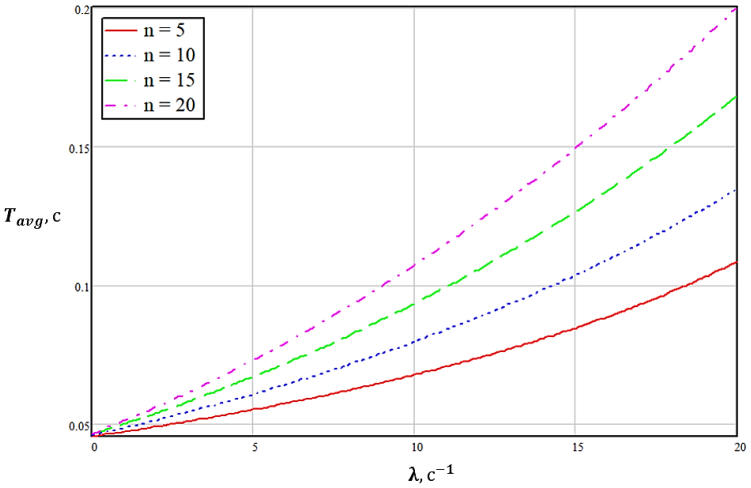


Рис. 10. Расчет математического ожидания времени пребывания запросов в узле от интенсивности трафика

Использование модели M/G/1. Установленная зависимость интенсивности обслуживания в контейнерах от числа *n* развернутых в

узле контейнеров и их активной части, зависящей от интенсивности входного потока, позволяет более точно представить процесс обслуживания в каждом контейнере моделью M/G/1.

В этом случае для определения среднего времени пребывания запроса в очереди контейнера воспользуемся формулой Полячека-Хинчина [16]:

$$w = \frac{\lambda_0 v^{(2)}}{2(1-\rho_0)} = \frac{\lambda_0 v^{(2)}}{2(1-\lambda_0 v)},$$

где $\lambda_0 = \lambda/n$, а v и $v^{(2)}$ первый и второй начальный момент времени обслуживания запросов в контейнере при заданном числе развернутых контейнеров n. Искомые первый и второй начальные моменты найдем на основе перебора гипотез с определением вероятности активности различного числа из развернутых контейнеров:

$$v = \sum_{i=0}^{n-1} \left(\frac{1}{\mu(n,i+1)} \right) \cdot C_{n-1}^{i} \cdot \left(\frac{\lambda}{n\mu_s} \right)^{i} \left(1 - \frac{\lambda}{n\mu_s} \right)^{n-i-1},$$

$$v^{(2)} = \sum_{i=0}^{n-1} \left(\frac{1}{\mu(n,i+1)} \right)^{2} \cdot C_{n-1}^{i} \cdot \left(\frac{\lambda}{n\mu_s} \right)^{i} \left(1 - \frac{\lambda}{n\mu_s} \right)^{n-i-1},$$

при этом $\frac{\lambda}{n\mu_s} < 1$ вероятность, что некоторой контейнер активен, а уточнение оценки интенсивности обслуживания в контейнере

$$\mu_s = \sum_{i=0}^{n-1} \mu(n,i+1) \cdot C_{n-1}^{i} \cdot \left(\frac{\lambda}{n\mu} \right)^{i} \left(1 - \frac{\lambda}{n\mu} \right)^{n-i-1},$$

где μ определяйся по формуле (3).

Учет задержки распределения запросов. Для кластеров, объединяющих N виртуальных машин, в каждой из которых развернуто n контейнеров, при вычислении среднего времени пребывания запросов в системе помимо рассмотренной выше оценки задержек в узле кластера необходимо учитывать дополнительную

задержку распределения запросов T_p по узлам кластера при балансировке нагрузки.

При распределении запросов с балансировкой нагрузки интенсивность запросов, поступающих в кластер $\lambda_{вх}$, равномерно распределяется по его узлам, то есть интенсивность поступления запросов в узел кластера будет равна $\lambda=\lambda_{вх}/N$. Если при балансировке загрузки в кластере очереди не формируется (рисунок 2), то время распределения запросов $T_p = \nu_p$, где ν_p – средние задержки в балансировщике при распределении запросов. Если в кластере очередь на диспетчеризацию (распределение с балансировкой) образуется (рис.3), то при представлении диспетчера (балансировщика) в виде классической одноканальной системы с бесконечной очередью типа M/M/1, имеем задержки распределения запросов:

$$T_p = \frac{\nu_p}{(1 - \lambda_{вход} \cdot \nu_p)}.$$

Представленная приближённая модель позволяет установить аналитическую зависимость задержек ожидания запросов в очередях для кластерных систем с контейнерной виртуализацией.

Развитие предложенных моделей может включать исследование влияния на задержки обслуживания и нарушения непрерывности функционирования из-за отказов, приводящих к потере времени на реконфигурацию структуры и контроль функционирования. Уточнение моделей может проводиться с учетом возможной неопределенности трафика и его периодических изменений. Представляется целесообразным исследование потенциальных возможностей уточнения предлагаемых моделей на основе композиционного подхода к имитационному моделированию систем массового обслуживания с параметрической неопределенностью [33].

5. Оценка вероятности своевременного обслуживания запросов. Предлагаемые выше приближенные модели массового обслуживания кластеров с контейнерной виртуализацией позволяют оценить среднее время пребывания запросов с учетом влияния на интенсивность их обслуживания числа развернутых и активных контейнеров, совместно использующих общие вычислительные ресурсы узлов.

Для функционирования кластера при ограничении на допустимое время ожидания важна оценка вероятности обслуживания запросов за время, меньшее предельно допустимого t_0.

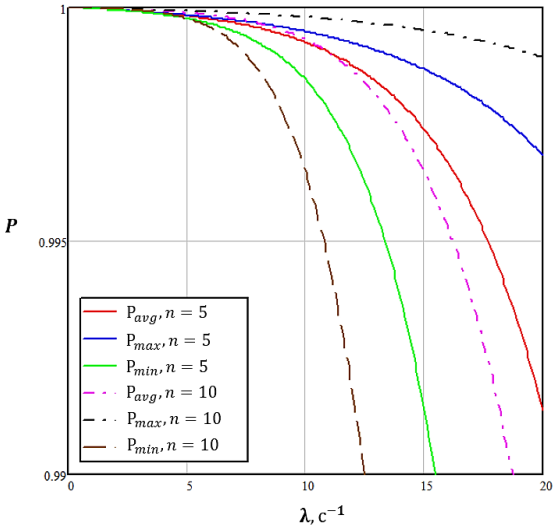
Функционирование контейнера представим моделью массового обслуживания с неограниченной очередью типа M/M/1. По результатам оценки математического ожидания интенсивности μ, с учетом влияния общего числа развернутых и активных контейнеров (пункты 1-4 приведенного выше алгоритма построения приближенной модели обслуживания), вероятность не превышения при ожидании в контейнере предельно допустимого времени t_0 определим как:

$$P = 1 - \frac{\lambda}{n\mu} \mathrm{e}^{\left(\frac{\lambda}{n} - \mu\right) t_0}. \quad (5)$$

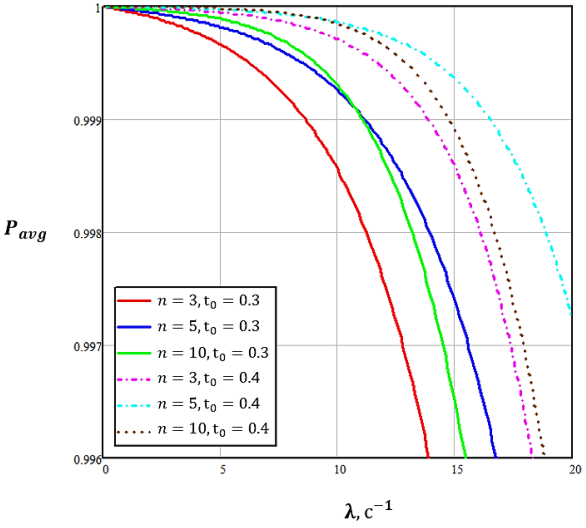
При расчетах может быть взято верхнее нижнее или усредненное значение интенсивности обслуживания запросов, полученное выше по формуле (3).

Искомая зависимость вероятности своевременного обслуживания запросов в узле кластера P за время, меньшее t_0, от интенсивности поступления запросов в узел кластера λ приведена на рисунке 11(а) для верхнего (P_{max}), нижнего (P_{min}) и усреднённого (P_{avg}) приближений. Из рисунка 11(а) видно, что разброс между верхней и нижней оценками увеличивается по мере роста интенсивности запросов λ (1/с) и числа развернутых в узле контейнеров. Расчёт проведён при предельно допустимом времени ожидания t_0=0,3 с. На рисунке 11(б) отражена зависимость усреднённой оценки вероятности своевременного обслуживания P от интенсивности запросов λ (1/с) при развертывании в виртуальной машине n=3, n=5 и n=10 контейнеров при допустимом времени ожидания t_0=0,3 с и t_0=0,4 с. Из рисунка 11(б) видно, что увеличение числа развертываемых в узле контейнеров не всегда однозначно приводит к повышению вероятности обслуживания P за время, не превосходящее предельно допустимое t_0 время ожидания.

Результаты граничных (верхней и нижней) и усредненной оценок вероятности выполнения запросов P за время меньшее t_0 при изменениях числа контейнеров в узле n представлены на рисунке 12.



а)



б)

Рис. 11. Результаты граничных (верхней и нижней) и усредненной оценок вероятности выполнения запросов P за время меньшее t_0 при изменениях интенсивности λ (1/с) трафика

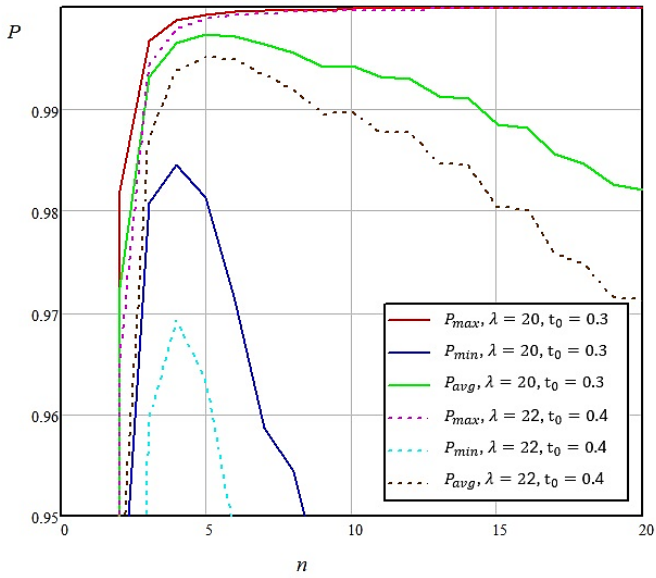


Рис. 12. Результаты граничных (верхней и нижней) и усредненной оценок вероятности выполнения запросов P за время меньшее t_0 при изменениях числа контейнеров в узле n

Представленные на рисунке 12 графики показывают существование оптимального числа развернутых в узлах кластера контейнеров, при котором достигается максимум вероятности своевременного обслуживания запросов.

Проведенные исследования показывают, что при использовании верхней, нижней и усредненной оценки интенсивности обслуживания в контейнерах искомое оптимальное значение числа развертываемых в узле контейнеров имеет близкие целочисленные значения.

В данной статье (рисунок 8) показано существование оптимального числа развернутых контейнеров в узле, при котором обеспечивается минимум времени пребывание запросов в узле. Постановка и решение задачи оптимизации выходит за рамки данного исследования и требует дополнительных исследований, в том числе при возможности изменений интенсивности входного потока (например, его периодических изменений в течение суток). В простейшем случае, когда интенсивность входного потока задана, возможна следующая простейшая постановка оптимизации.

Для систем, к которым не предъявляется требование выполнения запросов за предельно допустимое время, в качестве

критерия оптимальности (целевой функции) используется минимум среднего времени пребывания запросов в системе.

Для систем, к которым предъявляется требование выполнения запросов за предельно допустимое время, в качестве критерия оптимальности используется максимум вероятности не превышения задержек ожидания предельно допустимого времени t_0, определяемой по формуле (5).

Для каждого из сформулированных критерием в качестве параметра управления выступает число контейнеров n разворачиваемых в узле. В качестве ограничений рассматривается условие стационарности $\frac{\lambda}{n \cdot \mu} < 1$, где μ определяется по формуле (3).

В качестве ограничения также может вводиться наибольшее число разворачиваемых в узле контейнеров.

Предложенные приближенные модели массового обслуживания в кластерных системах с компьютерной виртуализацией позволяют установить влияния числа разворачиваемых в узлах кластера контейнеров на задержку и вероятность своевременного обслуживания запросов.

Обсуждение результатов и направления развития. Новизна предлагаемой модели массового обслуживания кластера с контейнерной виртуализацией заключается в учете влияния разделения общих ресурсов между активными и неактивными контейнерами на снижение интенсивности обслуживания в каждом контейнере.

Отличие предлагаемой модели кластера с контейнерной виртуализацией состоит в представлении кластера группой одноканальных взаимозависимых СМО, каждая из которых соответствует отдельному контейнеру. Взаимозависимость одноканальных СМО, объединяемых в группу, обусловлена тем, что распределение общих ограниченных ресурсов между активными и неактивными контейнерами, приводит к снижению интенсивности обслуживания в каждом из них. Особенность предлагаемой модели заключается в учете этой взаимозависимости отдельных одноканальных СМО.

Развитие предлагаемых моделей для структурно-параметрической оптимизации кластерных систем с контейнерной виртуализацией предусматривается в направлении учета влияния отказов, сбоев и ошибок вычислений [37 – 40]. Модели должны быть ориентированы на оптимизацию кластера в условиях многокритериальности и стохастической неопределенности трафика,

в том числе при его функциональной и параметрической неоднородности, включая неоднородность запросов по критичности к допустимому времени их ожидания [13, 14, 41, 42]. Для распределенных компьютерных систем, предусматривающих межмашинный обмен данными через сеть [13, 14, 43], при оценке надежности и задержек передач необходим учет избыточности их топологии и возможности организации многопутевых передач, в том числе с репликацией и сегментацией передаваемых данных [13, 14, 44].

В дальнейшем предполагается построение модели надежности рассматриваемых систем и постановку задачи многокритериальной оптимизации. В качестве критерия, объединяющего рассмотренные в данной статье показатели задержек и показатели надежности, предполагается использовать, в частности, коэффициент готовности к выполнению запросов в установленные сроки [45]. Заметим, что этот критерий не формальный, а имеет физический смысл.

В направлении исследование возможностей изоляции различных вычислений, выполняемых одновременно, в рамках одного вычислительного устройства известных из [15] предполагается построение моделей, не ограниченных простейшими потоками и экспоненциальным обслуживанием [46], с учетом особенностей контейнерной виртуализации. Указанные направления развития работы ориентированы на возможность количественного обоснования выбора решений по контейнерной виртуализации, а также сравнения и обоснования выбора с учетом других технологий виртуализации [47 – 52].

6. Заключение. Предложена приближенная аналитическая оценка вероятностно временных показателей качества обслуживания запросов в кластерных системах конвейерной виртуализации, получаемая при сочетании экспериментальных исследований и аналитического моделирования.

При построении аналитической модели обслуживания каждый контейнер представляется как отдельная одноканальная система массового обслуживания с бесконечной очередью и простейшим входным потоком. Основное отличие предлагаемой модели виртуального кластера заключается в граничной верхней, нижней и усредненной оценке возможного снижения интенсивности обслуживания в контейнерах из-за разделения между ними ограниченных общедоступных вычислительных ресурсов узла кластера в зависимости от количества развернутых в нем контейнеров

и изменяющегося числа активных контейнеров, зависящем от интенсивности входного потока.

На основе предлагаемых приближенных граничных моделей оценки среднего времени пребывания запросов в кластере и вероятности их своевременного обслуживания за время, не превышающее предельно допустимое значение, проведено исследование возможностей повышения эффективности кластерных систем конвейерной виртуализации. Оценка вероятности обслуживания запросов за время не превышающее предельно допустимые время особенно важно для систем, критичных к своевременности и непрерывности реализации вычислительного процесса.

Показано существование оптимального числа развернутых в узлах контейнеров, при котором среднее время пребывания запросов в системе минимально либо вероятность выполнения запросов за время меньшее предельно допустимого максимальна. При этом следует отметить, что целочисленные значения оптимального числа развернутых в узлах контейнеров, искомые при верхней, нижней и усредненной оценке, имеют достаточно близкие значения.

Предлагаемые модели могут быть применены при структурно-параметрической оптимизации кластеров с конвейерной виртуализацией, в том числе в случае масштабирования и адаптивной реконфигурации к изменениям трафика в системе. Реконфигурация кластера при изменениях трафика реализуется в результате перенастройки числа развернутых контейнеров.

Литература

1. Kumari P., Kaur P. A survey of fault tolerance in cloud computing // Journal of King Saud University-Computer and Information Sciences. 2021. vol. 33(10). pp. 1159-1176. DOI: 10.1016/j.jksuci.2018.09.021.
2. Половко А.М., Гуров С.В. Основы теории надежности // СПб.: БХВ-Петербург, 2006. 702 с.
3. Goyal P., Deora S.S. Reliability of Trust Management Systems in Cloud Computing // Indian Journal of Cryptography and Network Security (IJCNS). 2022. vol. 2. no. 1. pp. 1-5. DOI: 10.54105/ijcns.C1417.051322.
4. Chen G., Guan N., Huang K., Yi W. Fault-tolerant real-time tasks scheduling with dynamic fault handling // Journal of Systems Architecture. 2020. vol. 102. DOI: 10.1016/j.sysarc.2019.101688.
5. Shubinsky I.B., Rozenberg I.N., Papic L. Adaptive fault tolerance in real-time information systems // Reliability: Theory & Applications. 2017. vol. 12. no 1(44). pp. 18–25.
6. Alam K., Sharif K., Li F., Latif Z., Karim M.M., Biswas S., Nour B., Wang Y. A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV // ACM Computing Surveys (CSUR). 2020. vol. 53. no. 2. pp. 1–40. DOI: 10.1145/3379444.

7. Shukur H., Zeebaree S., Zebari R., Zeebaree D., Ahmed O. Cloud computing virtualization of resources allocation for distributed systems // Journal of Applied Science and Technology Trends. 2020. vol. 1. no. 2. pp. 98–105. DOI: 10.38094/jastt1331.
8. Compastié M., Badonnel R., Festor O., He R. From virtualization security issues to cloud protection opportunities: An in-depth analysis of system virtualization models // Computers & Security. 2020. vol. 97. DOI: 10.1016/j.cose.2020.101905.
9. Li Z., Jin H., Zou D., Yuan B. Exploring New Opportunities to Defeat Low-Rate DDoS Attack in Container-Based Cloud Environment // IEEE Transactions on Parallel and Distributed Systems. 2020. vol. 31. no. 3. pp. 695–706. DOI: 10.1109/TPDS.2019.2942591.
10. Chen H., Qin W., Wang L. Task partitioning and offloading in IoT cloud-edge collaborative computing framework: a survey // Journal of Cloud Computing. 2022. vol. 11. no. 1. DOI: 10.1186/s13677-022-00365-8.
11. Kushchazli A., Safargalieva A., Kochetkova I., Gorshenin A. Queuing Model with Customer Class Movement across Server Groups for Analyzing Virtual Machine Migration in Cloud Computing // Mathematics. 2024. vol. 12. no. 3. DOI: 10.3390/math12030468.
12. Choudhary A., Govil M.C., Singh G., Awasthi L.K., Pilli E.S., Kapil D. A critical survey of live virtual machine migration techniques // Journal of Cloud Computing. 2017. vol. 6. DOI: 10.1186/s13677-017-0092-1.
13. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. The probability of timeliness of a fully connected exchange in a redundant real-time communication system // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF 2020). 2020. pp. 1-4, DOI: 10.1109/WECONF48837.2020.9131517.
14. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. Multipath Transmission of Heterogeneous Traffic in Acceptable Delays with Packet Replication and Destruction of Expired Replicas in the Nodes that Make Up the Path // Communications in Computer and Information Science. 2023. vol. 1748. pp. 104–121.
15. Клейнрок Л. Вычислительные системы с очередями / перевод с английского под редакцией д-ра техн. наук Б.С. Цыбакова. Москва: Мир, 1979. 600 с.
16. Клейнрок Л. Теория массового обслуживания // М.: Машиностроение, 1979. 432 с.
17. Ejem A., Njoku C.N., Uzoh O.F., Odii J.N. Queue Control Model in a Clustered Computer Network using M/M/m Approach // International Journal of Computer Trends and Technology (IJCTT). 2016. vol. 35. no. 1. pp. 12–20. DOI: 10.14445/22312803/IJCTT-V35P103.
18. Khalill M.M., Khomonenko A.D., Gindin S.I. Load balancing cloud computing with web-interface using multi-channel queuing systems with warming up and cooling // Intelligent Distributed Computing XIII. 2020. vol. 868. pp. 385–393. DOI: 10.1007/978-3-030-32258-8_45.
19. Mochalov V.P., Bratchenko N.Yu., Linets G.I., Palkanov I.S. Methods and models of resource allocation in load balancing clusters s for data centers // Modeling, Optimization and Information Technology. 2022. vol. 10. no. 2. pp. 1–15. DOI: 10.26102/2310-6018/2022.37.2.030.
20. Volkov A.O. Evaluation of cloud computing cluster performance // T-Comm. 2020. vol. 14. no. 12. pp. 72–79. DOI: 10.13140/RG.2.2.32529.86885.
21. Goncharenko V.A., Lokhvitsky V.A. Cluster Load Balancing Algorithms Based on Shortest Queue Models // Intellectual Technologies on Transport. 2022. vol. 3. no. 31. pp. 37–45. DOI: 10.24412/2413-2527-2022-331-37-45.

22. Мартынчук И.Г. Прогнозирование мультисезонных нагрузочных процессов в эластичных системах // Изв. вузов. Приборостроени. 2023. Т. 66. № 11. С. 907–916. DOI: 10.17586/0021-3454-2023-66-11-907-916.
23. Singh P. Mathematical Rendition of Generic Process Model-based Design for Decision Making about Cloud Instance Autoscaling Actions // IOSR Journal of Mathematics (IOSR-JM). 2021. vol. 17. no. 3. pp. 49–56. DOI: 10.9790/5728-1703024956.
24. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Recovery of Real-Time Clusters with the Division of Computing Resources into the Execution of Functional Queries and the Restoration of Data Generated Since the Last Backup // International Conference on Distributed Computer and Communication Networks. 2023. pp. 236–250. DOI: 10.1007/978-3-031-50482-2_19.
25. Богатырев В.А., Богатырев С.В., Богатырев А.В. Оценка готовности компьютерной системы к своевременному обслуживанию запросов при его совмещении с информационным восстановлением памяти после отказов // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23. № 3. С. 608–617. DOI: 10.17586/2226-1494-2023-23-3-608-617.
26. Srivastava A., Kumar N. Queueing model based dynamic scalability for containerized cloud // International Journal of Advanced Computer Science and Applications. 2023. vol. 14. no. 1. pp. 465–472. DOI: 10.14569/IJACSA.2023.0140150.
27. Khazaei H., Barna C., Beigi-Mohammadi N., Litoiu M. Efficiency analysis of provisioning microservices // Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom). IEEE. 2016. pp. 261–268.
28. Liu B., Chen Y. A scalable fine-grained analytic model for container cloud data centres // Int. J. Internet Technol. Secur. Trans. 2019. vol. 9. no. 4. pp. 355–389.
29. Ye T., Guangtao X., Shiyou Q., Minglu L. An auto-scaling framework for containerized elastic applications // Proceedings of the 3rd International Conference on Big Data Computing and Communications (BIGCOM), IEEE. 2017. pp. 422–430.
30. El Kafhali S., El Mir I., Salah K., Hanini M. Dynamic Scalability Model for Containerized Cloud Services // Arabian Journal for Science and Engineering. 2020. vol. 45(12). pp. 10693–10708. DOI: 10.1007/s13369-020-04847-2.
31. Фунг В., Богатырев В.А., Кармановский Н.С., Лэ В. Оценка вероятностно-временных характеристик компьютерной системы с контейнерной виртуализацией // Научно-технический вестник информационных технологий, механики и оптики. 2024. Т. 24. № 2. С. 249–255.
32. Фунг В., Богатырев В.А. Задержки и надежность обслуживания запросов в виртуальном компьютерном кластере // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2024. № 68. С. 48–58.
33. Гончаренко В.А., Хомоненко А.Д., Абу Хасан Р. Композиционный подход к имитационному моделированию систем массового обслуживания со случайными параметрами // Информатика и автоматизация. 2024. Т. 23. № 6. С. 1577–1608. DOI: 10.15622/ia.23.6.1
34. Syed Z.A., Gummadi S., Mahima E.L., Naina S.R., Eswaran S., Honnavalli P. Performance Analysis Of 5G Network Slicing Simulations Using SimPy // Proceedings of the 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). Bangalore, India: IEEE. 2022. pp. 1–6. DOI: 10.1109/CONECCT55679.2022.9865799.
35. Peyman M., Copado P., Panadero J., Juan A.A., Dehghanimohammadabadi M. A Tutorial on how to Connect Python with Different Simulation Software to Develop Rich Simheuristics // Proceedings of the 2021 Winter Simulation Conference (WSC). Phoenix, AZ, USA: IEEE, 2021. pp. 1–12. DOI: 10.1109/WSC52266.2021.9715511.

36. Nam S.H., Oh S.H., Yoon H.C., Cho Y.I., Cho K.Y., Kwak D.H., Woo J.H. Development of DES Application for Factory Material Flow Simulation with SimPy // Proceedings of the 2022 Winter Simulation Conference (WSC), Singapore. IEEE. 2022. pp. 1545–1556. DOI: 10.1109/WSC57314.2022.10015508.
37. Oszczypała M., Ziółkowski J., Małachowski J. Redundancy allocation problem in repairable k-out-of-n systems with cold, warm, and hot standby: A genetic algorithm for availability optimization // Applied Soft Computing. 2024. vol. 165. DOI: 10.1016/j.asoc.2024.112041.
38. Oszczypała M., Konwerski J., Ziółkowski J., Małachowski J. Reliability analysis and redundancy optimization of k-out-of-n systems with random variable k using continuous time Markov chain and Monte Carlo simulation // Reliability Engineering & System Safety. 2024. vol. 242. DOI: 10.1016/j.ress.2023.109780.
39. Komari I.E., Fedorenko M., Kharchenko V., Yehorova Y., Bardis N., Lutai L. The neural modules network with collective relearning for the recognition of diseases: Fault-tolerant structures and reliability assessment // Neural Networks. 2020. vol. 1. DOI: 10.46300/9106.2020.14.102.
40. Kumar A., Saini M., Saini D.K., Badiwal, N. Cyber physical systems-reliability modelling: critical perspective and its impact // International Journal of System Assurance Engineering and Management. 2021. vol. 12. pp. 1334–1347. DOI: 10.1007/s13198-021-01305-6.
41. Lambropoulos G., Mitropoulos S., Douligeris C. Improving Business Performance by Employing Virtualization Technology: A Case Study in the Financial Sector // Computers. 2021. vol. 10. no. 4. DOI: 10.3390/computers10040052.
42. Shi F., Lin J. Virtual machine resource allocation optimization in cloud computing based on multiobjective genetic algorithm // Computational Intelligence and Neuroscience. 2022. vol. 2022. no. 1. DOI: 10.1155/2022/7873131.
43. Al-Dulaimi M.K.H., Al-Dulaimi O.M.K., Al-Dulaimi A.M.K., Alexandra M.O., Jihad N. Deep Learning for Wireless Network Distribution (5G/LTE) // Proceedings of the 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES). Plovdiv, Bulgaria: IEEE, 2023. pp. 1–6. DOI: 10.1109/CIEES58940.2023.10378816.
44. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Control of Multipath Transmissions in the Nodes of Switching Segments of Reserved Paths // International Conference on Information, Control, and Communication Technologies (ICCT). IEEE, 2022. pp. 1–5.
45. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Reliability and Timeliness of Servicing Requests in Infocommunication Systems, Taking into Account the Physical and Information Recovery of Redundant Storage Devices // International Conference on Information, Control, and Communication Technologies (ICCT). IEEE, 2022. pp. 1–4.
46. Хомоненко А.Д., Благовещенская Е.А., Проурзин О.В., Андрук А.А. Прогноз надежности кластерной вычислительной системы с помощью полумарковской модели альтернирующих процессов и мониторинга // Наукоемкие технологии в космических исследованиях Земли. 2018. Т. 10. № 4. С. 72–82. DOI: 10.24411/2409-5419-2018-10099.
47. Sturley H., Fourmier A., Salcedo-Navarro A., Garcia-Pineda M., Segura-Garcia J. Virtualization vs. containerization, a comparative approach for application deployment in the continuum focused on the edge // Future Internet. 2024. vol. 16. no. 11.
48. Aniruddh M., Dinkar A., Mouli S.C., Sahana B., Deshpande A.A. Comparison of containerization and virtualization in cloud architectures // 2021 IEEE International conference on electronics, computing and communication technologies (CONECT). 2021. pp. 1–5.

49. Abuabdo A., Al-Sharif Z.A. Virtualization vs. containerization: towards a multithreaded performance evaluation approach // 2019 IEEE/ACS 16th international conference on computer systems and applications (AICCSA). 2019. pp. 1–6.
50. Kumar S. A brief study on virtualization and containerization // Cloud computing. 2022. pp. 37–46.
51. Xu J., Fortes J.A. Multi-objective virtual machine placement in virtualized data center environments // 2010 IEEE/ACM International conference on green computing and communications (GreenCom) and International conference on cyber, physical and social computing (CPSCom). 2010. pp. 179–188.
52. Ходосов М.А. Анализ технологий виртуализации // ИИАСУ'23 – Искусственный интеллект в автоматизированных системах управления и обработки данных: Сборник статей II Всероссийской научной конференции (Москва, 27–28 апреля 2023 г.): в 5 т. М.: «КДУ», «Добросвет», 2024. Т. 4. С. 475–481. DOI: 10.31453/kdu.ru.978-5-7913-1354-6-2024-488.

Богатырев Владимир Анатольевич — д-р техн. наук, профессор факультета, факультет программной инженерии и компьютерной техники, Университет ИТМО; профессор кафедры, кафедра информационной безопасности, Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП). Область научных интересов: теория и методы обеспечения надежности, отказоустойчивости и эффективности компьютерных систем и сетей. Число научных публикаций — 460. vladimir.bogatyrev@gmail.com; Кронверкский проспект, 49А, 197101, Санкт-Петербург, Россия; р.т.: +7(812)232-5278.

Фунг Ван Кю — аспирант, факультет программной инженерии и компьютерной техники, Университет ИТМО. Область научных интересов: система интернета вещей, теория надежности, теория массового обслуживания, отказоустойчивости и эффективности компьютерных систем и сетей. Число научных публикаций — 5. phungvanquy97@gmail.com; Кронверкский проспект, 49А, 197101, Санкт-Петербург, Россия; р.т.: +7(995)990-8193.

DOI 10.15622/ia.24.3.7

V. BOGATYREV, V. PHUNG

AN APPROXIMATE ASSESSMENT OF LATENCY IN A COMPUTER SYSTEM WITH CONTAINER VIRTUALIZATION

Bogatyrev V., Phung V. **An Approximate Assessment of Latency in a Computer System with Container Virtualization.**

Abstract. The key role in achieving high reliability, security, fault tolerance, and low latency of query service in distributed systems (including cloud computing) is played by the consolidation of data processing and storage resources in clusters, the efficiency of which increases with the use of virtual machine technologies and container virtualization. The complexity of building queuing models for container virtualization systems is caused by the fact that the intensity of query execution in each container is associated with the dynamic division of shared resources between active (performing functional tasks) containers and the costs of supporting all containers deployed in the VM, including inactive containers waiting for service requests to be sent to them. The reduction in service intensity in each container due to shared resource allocation depends on many factors that are difficult to investigate. For clusters with container virtualization, this article provides an approximate boundary estimate of the average request waiting time and the probability of timely service. When building an analytical model, each container is represented as a separate single-channel queuing system with an infinite queue and the simplest input stream. The key feature of the proposed virtual cluster model is the estimation of upper, lower, and average bounds for the potential service intensity reduction in containers, resulting from the allocation of a node's limited computing resources among them. This depends on the number of deployed containers and the dynamically varying count of active containers, which is influenced by the input stream intensity. The study demonstrates the existence of an optimal number of containers per node, minimizing the average request processing time or maximizing the probability of timely request execution. The proposed models can be applied to the structural and parametric optimization of clusters with pipelined virtualization, including in the case of scaling and reconfiguration adaptive to traffic changes by disconnecting or connecting some of the deployed containers depending on changes in the load in the system.

Keywords: container, container virtualization, cluster, resource allocation, latency.

References

1. Kumari P. Kaur P. A survey of fault tolerance in cloud computing. Journal of King Saud University-Computer and Information Sciences. 2021. vol. 33(10). pp. 1159-1176. DOI: 10.1016/j.jksuci.2018.09.021.
2. Polovko A.M., Gurov S.V. Osnovy teorii nadezhnosti [Fundamentals of Reliability Theory]. SPb.: BHV-Peterburg. 2006. 702 p. (In Russ.).
3. Goyal P., Deora S.S. Reliability of Trust Management Systems in Cloud Computing. Indian Journal of Cryptography and Network Security (IJCNS). 2022. vol. 2. no. 1. pp. 1-5. DOI: 10.54105/ijcns.C1417.051322.
4. Chen G., Guan N., Huang K., Yi W. Fault-tolerant real-time tasks scheduling with dynamic fault handling. Journal of Systems Architecture. 2020. vol. 102. DOI: 10.1016/j.sysarc.2019.101688.
5. Shubinsky I.B., Rozenberg I.N., Papic L. Adaptive fault tolerance in real-time information systems. Reliability: Theory & Applications. 2017. vol. 12. no 1(44). pp. 18–25.

6. Alam K., Sharif K., Li F., Latif Z., Karim M.M., Biswas S., Nour B., Wang Y. A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV. ACM Computing Surveys (CSUR). 2020. vol. 53. no. 2. pp. 1–40. DOI: 10.1145/3379444.
7. Shukur H., Zeebaree S., Zebari R., Zeebaree D., Ahmed O. Cloud computing virtualization of resources allocation for distributed systems. Journal of Applied Science and Technology Trends. 2020. vol. 1. no. 2. pp. 98–105. DOI: 10.38094/jastt1331.
8. Compastié M., Badonnel R., Festor O., He R. From virtualization security issues to cloud protection opportunities: An in-depth analysis of system virtualization models. Computers & Security. 2020. vol. 97. DOI: 10.1016/j.cose.2020.101905.
9. Li Z., Jin H., Zou D., Yuan B. Exploring New Opportunities to Defeat Low-Rate DDoS Attack in Container-Based Cloud Environment. IEEE Transactions on Parallel and Distributed Systems. 2020. vol. 31. no. 3. pp. 695–706. DOI: 10.1109/TPDS.2019.2942591.
10. Chen H., Qin W., Wang L. Task partitioning and offloading in IoT cloud-edge collaborative computing framework: a survey. Journal of Cloud Computing. 2022. vol. 11. no. 1. DOI: 10.1186/s13677-022-00365-8.
11. Kushchazli A., Safargalieva A., Kochetkova I., Gorshenin A. Queuing Model with Customer Class Movement across Server Groups for Analyzing Virtual Machine Migration in Cloud Computing. Mathematics. 2024. vol. 12. no. 3. DOI: 10.3390/math12030468.
12. Choudhary A., Govil M.C., Singh G., Awasthi L.K., Pilli E.S., Kapil D. A critical survey of live virtual machine migration techniques. Journal of Cloud Computing. 2017. vol. 6. DOI: 10.1186/s13677-017-0092-1.
13. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. The probability of timeliness of a fully connected exchange in a redundant real-time communication system. Wave Electronics and its Application in Information and Telecommunication Systems (WECONF 2020). 2020. pp. 1-4, DOI: 10.1109/WECONF48837.2020.9131517.
14. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. Multipath Transmission of Heterogeneous Traffic in Acceptable Delays with Packet Replication and Destruction of Expired Replicas in the Nodes that Make Up the Path. Communications in Computer and Information Science. 2023. vol. 1748. pp. 104–121.
15. Kleinrock, L. Vychislitel'nye sistemy s ocheredjami [Computing systems with queues]. Moscow : Mir Publ., 1979. 600 p. (In Russ.).
16. Klejnrok L. Teorija massovogo obsluzhivanija [Theory of Queueing Systems]. M.: Mashinostroenie. 1979. 432 p. (In Russ.).
17. Ejem A., Njoku C.N., Uzoh O.F., Odii J.N. Queue Control Model in a Clustered Computer Network using M/M/m Approach. International Journal of Computer Trends and Technology (IJCTT). 2016. vol. 35. no. 1. pp. 12–20. DOI: 10.14445/22312803/IJCTT-V35P103.
18. Khalill M.M., Khomonenko A.D., Gindin S.I. Load balancing cloud computing with web-interface using multi-channel queuing systems with warming up and cooling. Intelligent Distributed Computing XIII. 2020. vol. 868. pp. 385–393. DOI: 10.1007/978-3-030-32258-8_45.
19. Mochalov V.P., Bratchenko N.Yu., Linets G.I., Palkanov I.S. Methods and models of resource allocation in load balancing clusters s for data centers. Modeling, Optimization and Information Technology. 2022. vol. 10. no. 2. pp. 1–15. DOI: 10.26102/2310-6018/2022.37.2.030.
20. Volkov A.O. Evaluation of cloud computing cluster performance. T-Comm. 2020. vol. 14. no. 12. pp. 72–79. DOI: 10.13140/RG.2.2.32529.86885.

21. Goncharenko V.A., Lokhvitsky V.A. Cluster Load Balancing Algorithms Based on Shortest Queue Models. Intellectual Technologies on Transport. 2022. vol. 3. no. 31. pp. 37–45. DOI: 10.24412/2413-2527-2022-331-37-45.
22. Martynchuk I.G. [Forecasting multi-seasonal load processes in elastic computing systems]. Journal of Instrument Engineering. 2023. vol. 66. no. 11. pp. 907–916. DOI: 10.17586/0021-3454-2023-66-11-907-916. (In Russ.).
23. Singh P. Mathematical Rendition of Generic Process Model-based Design for Decision Making about Cloud Instance Autoscaling Actions. IOSR Journal of Mathematics (IOSR-JM). 2021. vol. 17. no. 3. pp. 49–56. DOI: 10.9790/5728-1703024956.
24. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Recovery of Real-Time Clusters with the Division of Computing Resources into the Execution of Functional Queries and the Restoration of Data Generated Since the Last Backup. International Conference on Distributed Computer and Communication Networks. 2023. pp. 236–250. DOI: 10.1007/978-3-031-50482-2_19.
25. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. [Assessment of the readiness of a computer system for timely servicing of requests when combined with information recovery after failures] Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki – Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics. 2023. vol. 23. no. 3. pp. 608–617. DOI: 10.17586/2226-1494-2023-23-3-608-617. (In Russ.).
26. Srivastava A., Kumar N. Queueing model based dynamic scalability for containerized cloud. International Journal of Advanced Computer Science and Applications. 2023. vol. 14. no. 1. pp. 465–472. DOI: 10.14569/IJACSA.2023.0140150.
27. Khazaei H., Barna C., Beigi-Mohammadi N., Litoiu M. Efficiency analysis of provisioning microservices. Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom). IEEE. 2016. pp. 261–268.
28. Liu B., Chen Y. A scalable fine-grained analytic model for container cloud data centres. Int. J. Internet Technol. Secur. Trans. 2019. vol. 9. no. 4. pp. 355–389.
29. Ye T., Guangtao X., Shiyou Q., Minglu L. An auto-scaling framework for containerized elastic applications. Proceedings of the 3rd International Conference on Big Data Computing and Communications (BIGCOM), IEEE. 2017. pp. 422–430.
30. El Kafhali S., El Mir I., Salah K., Hanini M.. Dynamic Scalability Model for Containerized Cloud Services. Arabian Journal for Science and Engineering. 2020. vol. 45(12). pp. 10693–10708. DOI: 10.1007/s13369-020-04847-2.
31. [Assessment of probabilistic-temporal characteristics of a computer system with container virtualization]. Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki – Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics. 2024. vol. 24. no. 2. pp. 249–255. (In Russ.).
32. Phung V., Bogatyrev V.A. [Delays and Reliability of Query Processing in a Virtual Computer Cluster]. Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika – Bulletin of Tomsk State University. Management, computer engineering and computer science. 2024. no. 68. pp. 48–58. (In Russ.).
33. Goncharenko V., Khomonenko A., Abu Khasan R. [A Compositional Approach to the Simulation of Queuing Systems with Random Parameters]. Informatics and Automation. 2024. vol. 23. no. 6. pp. 1577–1608. DOI: 10.15622/ia.23.6.1. (In Russ.).
34. Syed Z.A., Gummadi S., Mahima E.L., Naina S.R., Eswaran S., Honnavalli P. Performance Analysis Of 5G Network Slicing Simulations Using SimPy. Proceedings of the 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). Bangalore, India: IEEE. 2022. pp. 1–6. DOI: 10.1109/CONECCT55679.2022.9865799.

35. Peyman M., Copado P., Panadero J., Juan A.A., Dehghanimohammadabadi M. A Tutorial on how to Connect Python with Different Simulation Software to Develop Rich Simheuristics. Proceedings of the 2021 Winter Simulation Conference (WSC). Phoenix, AZ, USA: IEEE, 2021. pp. 1–12. DOI: 10.1109/WSC52266.2021.9715511.
36. Nam S.H., Oh S.H., Yoon H.C., Cho Y.I., Cho K.Y., Kwak D.H., Woo J.H. Development of DES Application for Factory Material Flow Simulation with SimPy. Proceedings of the 2022 Winter Simulation Conference (WSC), Singapore. IEEE. 2022. pp. 1545–1556. DOI: 10.1109/WSC57314.2022.10015508.
37. Oszczypała M., Ziółkowski J., Małachowski J. Redundancy allocation problem in repairable k-out-of-n systems with cold, warm, and hot standby: A genetic algorithm for availability optimization. Applied Soft Computing. 2024. vol. 165. DOI: 10.1016/j.asoc.2024.112041.
38. Oszczypała M., Konwerski J., Ziółkowski J., Małachowski J. Reliability analysis and redundancy optimization of k-out-of-n systems with random variable k using continuous time Markov chain and Monte Carlo simulation. Reliability Engineering & System Safety. 2024. vol. 242. DOI: 10.1016/j.ress.2023.109780.
39. Komari I.E., Fedorenko M., Kharchenko V., Yehorova Y., Bardis N., Lutai L. The neural modules network with collective relearning for the recognition of diseases: Fault-tolerant structures and reliability assessment. Neural Networks. 2020. vol. 1. DOI: 10.46300/9106.2020.14.102.
40. Kumar A., Saini M., Saini D.K., Badiwal, N. Cyber physical systems-reliability modelling: critical perspective and its impact. International Journal of System Assurance Engineering and Management. 2021. vol. 12. pp. 1334–1347. DOI: 10.1007/s13198-021-01305-6.
41. Lambropoulos G., Mitropoulos S., Douligeris C. Improving Business Performance by Employing Virtualization Technology: A Case Study in the Financial Sector. Computers. 2021. vol. 10. no. 4. DOI: 10.3390/computers10040052.
42. Shi F., Lin J. Virtual machine resource allocation optimization in cloud computing based on multiobjective genetic algorithm. Computational Intelligence and Neuroscience. 2022. vol. 2022. no. 1. DOI: 10.1155/2022/7873131.
43. Al-Dulaimi M.K.H., Al-Dulaimi O.M.K., Al-Dulaimi A.M.K., Alexandra M.O., Jihad N. Deep Learning for Wireless Network Distribution (5G/LTE). Proceedings of the 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES). Plovdiv, Bulgaria: IEEE, 2023. pp. 1–6. DOI: 10.1109/CIEES58940.2023.10378816.
44. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Control of Multipath Transmissions in the Nodes of Switching Segments of Reserved Paths. International Conference on Information, Control, and Communication Technologies (ICCT). IEEE, 2022. pp. 1–5.
45. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Reliability and Timeliness of Servicing Requests in Infocommunication Systems, Taking into Account the Physical and Information Recovery of Redundant Storage Devices. International Conference on Information, Control, and Communication Technologies (ICCT). IEEE, 2022. pp. 1–4.
46. Khomonenko A.D., Blagoveshchenskaya E.A., Prourzin O.V., Andruk A.A. [Forecasting the reliability of a cluster computing system using a semi-Markov model of alternating processes and monitoring]. High-tech technologies in space exploration of the Earth. 2018. vol. 10. no. 4. pp. 72–82. DOI: 10.24411/2409-5419-2018-10099. (In Russ.).
47. Sturley H., Fourmier A., Salcedo-Navarro A., Garcia-Pineda M., Segura-Garcia J. Virtualization vs. containerization, a comparative approach for application deployment in the continuum focused on the edge. Future Internet. 2024. vol. 16. no. 11.

48. Aniruddh M., Dinkar A., Mouli S.C., Sahana B., Deshpande A.A. Comparison of containerization and virtualization in cloud architectures. 2021 IEEE International conference on electronics, computing and communication technologies (CONECT). 2021. pp. 1–5.
49. Abuabdo A., Al-Sharif Z.A. Virtualization vs. containerization: towards a multithreaded performance evaluation approach. 2019 IEEE/ACS 16th international conference on computer systems and applications (AICCSA). 2019. pp. 1–6.
50. Kumar S. A brief study on virtualization and containerization. Cloud computing. 2022. pp. 37–46.
51. Xu J., Fortes J.A. Multi-objective virtual machine placement in virtualized data center environments. 2010 IEEE/ACM International conference on green computing and communications (GreenCom) and International conference on cyber, physical and social computing (CPSCom). 2010. pp. 179–188.
52. Khodosov M.A. Analiz tekhnologij virtualizacii [Analysis of virtualization technologies]. IIASU'23 – Iskusstvennyj intellekt v avtomatizirovannyh sistemah upravleniya i obrabotki dannyh: Sbornik statej II Vserossijskoj nauchnoj konferencii: v 5 t. [IIASU'23 – Artificial intelligence in automated control and data processing systems: Collection of articles of the II All-Russian Scientific Conference: in 5 volumes]. Moscow: KDU, Dobrosvet, 2024. vol. 4. pp. 475–481. DOI: 10.31453/kdu.ru.978-5-7913-1354-6-2024-488. (In Russ.).

Bogatyrev Vladimir — Ph.D., Dr.Sci., Professor of the faculty, Faculty of software engineering and computer technology, ITMO University; Professor of the department, Department of information security, St. Petersburg State University of Aerospace Instrumentation. Research interests: theory and methods of ensuring reliability, fault tolerance, and efficiency of computer systems and networks. The number of publications — 460. vladimir.bogatyrev@gmail.com; 49A, Kronverksky Ave., 197101, St. Petersburg, Russia; office phone: +7(812)232-5278.

Phung Van Quy — Ph.D. student, Faculty of software engineering and computer technology, ITMO University. Research interests: internet of Things system, reliability theory, queuing theory, fault tolerance, and efficiency of computer systems and networks. The number of publications — 5. phungvanquy97@gmail.com; 49A, Kronverksky Ave., 197101, St. Petersburg, Russia; office phone: +7(995)990-8193.