

На правах рукописи



УДОВИЧЕНКО  
Антон Олегович

Разработка комплексной методики снижения влияния эффекта «старения»  
программного обеспечения на работу многомашинной вычислительной системы,  
построенной на основе технологии виртуальных машин

Специальность - 05.13.15  
Вычислительные машины, комплексы и компьютерные сети

АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук

МОСКВА – 2015

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Московский государственный университет путей сообщения» (МИИТ).

Научный руководитель: Соловьев Владимир Павлович, кандидат технических наук, доцент  
Заведующий кафедрой «Информационные технологии» ФГБОУ ВПО «Московский государственный университет путей сообщения»,  
ученый секретарь Ученого совета университета

Официальные оппоненты: Гагарина Лариса Геннадьевна, доктор технических наук, профессор  
Заведующий кафедрой «Информатика и программное обеспечение вычислительных систем»  
Московского института электронной техники  
(национального исследовательского университета)

Крепков Игорь Михайлович, кандидат технических наук, доцент  
Заведующий кафедрой «Прикладной и бизнес-информатики» Московского энергетического института (национального исследовательского университета)

Ведущая организация: ОАО «Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте»

Защита состоится «28» апреля 2015 г. в 12 часов на заседании диссертационного совета Д212.125.01 при Московском авиационном институте (национальном исследовательском университете) – МАИ по адресу: 125993, г. Москва, А-80, ГСП-3, Волоколамское шоссе, д. 4.

С диссертацией можно ознакомиться в библиотеке Московского авиационного института (национального исследовательского университета) – МАИ.

Отзывы, заверенные печатью, просьба высылать по адресу: 125993, г. Москва, А-80, ГСП-3, Волоколамское шоссе, д.4, МАИ, Ученый совет МАИ.

Автореферат разослан " \_\_\_\_ " \_\_\_\_\_ 2015 года.

Ученый секретарь  
диссертационного совета Д212.125.01  
кандидат технических наук, доцент



Корнеевкова А.В.

## Актуальность

В настоящее время компании и отдельные пользователи все больше зависят в своей работе от эффективности работы вычислительных систем (ВС) и ожидают своевременного и качественного получения доступа к информационным ресурсам. В таких условиях с каждым годом возрастают требования к эффективности функционирования ВС.

Сложность современных ВС и не всегда высокое качество ПО выступают одними из ключевых факторов нарушения работы ВС, которое проявляется в виде постепенного снижения производительности программного обеспечения с последующим сбоем. Рассмотрим пример. Очень распространенной причиной нарушения является невысвобождение оперативной памяти по завершению выполнения программой какой-либо задачи. Допустим, некоторый сервер, выполняющий обслуживание пользователей, выделяет 120 Кб оперативной памяти под каждый запрос, но по окончании его обработки не освобождает 20 Кб ввиду особенностей своей работы. В результате через некоторое время это приведет к исчерпанию свободной оперативной памяти и как следствие к снижению производительности сервера или сбою. О снижении производительности ПО в процессе его непрерывного выполнения с последующем сбоем также говорят как о «старении» процесса выполнения ПО (или эффекте «старения» ПО). В англоязычной научной литературе для данной проблемы используется специальный термин «software aging». Исследованию эффекта «старения» ПО посвящены работы зарубежных и отечественных исследователей, таких как Avritzer A., Castelli. V., Grottke M., Huang Y., Trivedi K., Vaidyanathan K., Ключников К. и др. Среди широко распространенных причин эффекта «старения» ПО, кроме невысвобождения оперативной памяти, выделяют невысвобождение файловых дескрипторов, накопление незавершенных процессов, дефрагментация оперативной памяти. Практически единственным решением, позволяющим избавиться от данной проблемы, является выявление её источника и его устранение, например, на основе замены ПО или внесение изменений в его исходный код. Сложность заключается в том, что источником эффекта «старения» ПО может выступать любая программа или даже группа программ, а наличие большого количества программ в ВС и сложность их взаимодействия делает решение данной задачи крайне сложной. Кроме того, устранение одного источника не гарантирует возникновения нового и не всегда может быть реализовано в силу экономических или технических причин. Как следствие, борьба с эффектом «старения» заключается в снижении его негативного влияния на работу ВС с использованием специальных методов, известных под термином «software rejuvenation». В научных работах по данной проблематике этот термин определяет метод регулярного восстановления рабочего состояния программы с целью предотвращения серьезных сбоев в будущем, где под рабочим состоянием программы понимается такое состояние её выполнения, при котором вероятность возникновения сбоя близка к нулю. Примерами данных методов являются перезапуск программы и перезагрузка ОС.

Одной из главных трудностей борьбы с эффектом «старения» является, тот факт, что существующие решения обладают высокими издержками, связанными, прежде всего, с остановкой выполнения программы в процессе восстановления. Кроме того, за последние несколько лет одним из наиболее значительных и быстро развивающихся направлений в корпоративных вычислениях стала технология виртуальных машин. Данная технология рассматривается специалистами как одна из ключевых технологий построения вычислительных систем. Работа технологии виртуальных машин строится

на основе разделения ресурсов физического компьютера между несколькими серверами. Одним из недостатков такого подхода является высокая зависимость эффективности работы серверов от работы платформы виртуализации, нарушение её работы может вести к нарушению работы всех серверов, размещенных на ней.

Проведенный анализ существующих решений по борьбе с эффектом «старения» ПО показал:

- Применение существующих решений, либо связано с нарушением работы пользователей, например, остановкой выполнения программы в процессе восстановления её рабочего состояния, либо они обладают существенными ограничениями, например, область восстановления ограничена компонентами только одной программы.
- Существующие решения обладают различными недостатками, такими как: слабая устойчивость к изменениям работы программы и/или операционной системы (ОС), отсутствие возможности учета негативного воздействия метода восстановления на работу программы, высокая трудоемкость подготовки решения к работе. Кроме того, решения нацелены на улучшение одного показателя эффективности работы программы.
- Существующие решения не учитывают специфику технологии виртуальных машин при борьбе с эффектом «старения» ПО, а именно размещение нескольких серверов на одной физическом компьютере.

Подводя итог, можно сказать, что проблема снижения негативного влияния эффекта «старения» ПО на работу ВС актуальна, в связи с этим требуется разработка комплексной методики снижения влияния эффекта «старения» ПО на работу многомашинной вычислительной системы, построенной на основе технологии виртуальных машин.

**Целью диссертационной работы** является снижение доли потерянных запросов и среднего времени отклика серверов многомашинной вычислительной системы, построенной на основе технологии виртуальных машин, посредством применения разработанной комплексной методики.

Для разработки комплексной методики и достижения поставленной цели необходимо **решение следующих задач:**

1. Разработка методов восстановления рабочего состояния программы, которые обеспечивают восстановление вне зависимости от источника эффекта «старения» ПО, без прерывания обслуживания пользователей в процессе восстановления и не требуют изменения исходного кода восстанавливаемой программы.
2. Разработка методов определения времени начала восстановления, которые учитывают негативное влияние на работу целевой программы, как эффекта «старения» ПО, так и метода её восстановления.
3. Разработка метода планирования процессов восстановления рабочего состояния, который обеспечивает эффективное использование ресурсов ВС с точки зрения возможности реализации процессов восстановления и их согласование с целью улучшения двух показателей их реализации: своевременность и длительность процесса восстановления.
4. Разработка общей схемы взаимодействия компонентов методики и политики управления процессами восстановления, обеспечивающие формирование целостного решения и позволяющие улучшить эффективность работы ВС по двум показателям:

среднее время отклика и доля потерянных запросов.

5. Программная реализация разработанной комплексной методики.
6. Проведение экспериментов для оценки эффективности работы комплексной методики с точки зрения эффективности работы ВС.

**Методы исследования.** Для решения поставленных задач были использованы методы теории вероятностей, теории многокритериальной оптимизации, регрессионного анализа, теории приближения, системного анализа, декомпозиции систем. При разработке программного обеспечения был использован объектно-ориентированный подход.

**Новыми научными результатами и положениями, выносимыми на защиту,** являются:

1. Набор методов восстановления рабочего состояния программы для платформы виртуализации (метод VMMR) и сервера (метод VMS), которые обладают следующими преимуществами перед существующими методами восстановления рабочего состояния программы:
  - обеспечивают восстановление вне зависимости от источника эффекта «старения» ПО;
  - не приводят к прерыванию обслуживания пользователей в процессе восстановления;
  - не требуют модификации исходного кода восстанавливаемой программы.
2. Набор методов определения времени начала восстановления:
  - метод CBR основан на мониторинге характеристик выполнения целевой программы и/или ОС и отличается от существующих методов тем, что ориентирован на определение времени начала восстановления платформы виртуализации и обеспечивает учет характера изменения условий работы платформы виртуализации;
  - метод ROR основан на мониторинге объема работы, выполненной целевой программы, и отличается от существующих методов тем, что ориентирован на определение времени начала восстановления сервера и обеспечивает учет требований к эффективности его работы по двум показателям – время обработки запросов и коэффициент готовности.
3. Метод планирования процессов восстановления RPRR, который обладает следующими преимуществами перед существующими методами:
  - обеспечивает согласование процессов восстановления различных программ;
  - учитывает три показателя, характеризующих эффективность процессов восстановления: количество виртуальных машин, активность которых сохраняется в процессе восстановления, своевременность и длительность процессов восстановления;
  - учитывает возможность перераспределения ресурсов ВС при реализации процессов восстановления рабочего состояния.
4. Комплексная методика борьбы с эффектом «старения» ПО, которая отличается от известных решений борьбы с эффектом «старения» ПО, тем что учитывает особенности технологии виртуальных машин и позволяет улучшить эффективность ВС по двум показателям: доля потерянных запросов и среднее время отклика.

**Объектом исследования** является многомашинные вычислительные системы, построенные на основе технологии виртуальных машин, выполняющие сервисные (обслуживающие) функции по запросам клиентов, предоставляя им доступ к

определённым ресурсам или услугам.

**Предметом исследования** является методы снижения влияния эффекта «старения» ПО на работу многомашинной вычислительной системы, построенной на основе технологии виртуальных машин.

**Практическая ценность работы** состоит в следующем:

- разработана и экспериментально проверена новая комплексная методика снижения влияния эффекта «старения» ПО на работу многомашинной вычислительной системы, построенной на основе технологии виртуальных машин, по сравнению с аналогичными решениями по борьбе с эффектом «старения» ПО (в случае наличия достаточного объема ресурсов для реализации процессов восстановления рабочего состояния разработанная методика позволяет исключить потерю запросов и обеспечить среднее время отклика на уровне лучших результатов среди сравниваемых решений, при отсутствии достаточного объема ресурсов методика показывает лучшие результаты в отношении доли потерянных запросов и среднее время отклика на уровне лучших результатов);
- выполнена экспериментальная проверка разработанного метода планирования процессов восстановления рабочего состояния программ, которая показала его эффективность в условиях дефицита свободных ресурсов ВС в сравнении с классическим подходом к размещению виртуальных машин (доля потерянных запросов в среднем снижена более чем на 60%);
- на базе подготовленной комплексной методики разработан программный комплекс, который может быть использован в многомашинной вычислительной системе, построенной на основе технологии виртуальных машин, для снижения негативного влияния эффекта «старения» на её эффективность по двум показателям: среднее время отклика и доля потерянных запросов.

Результаты работы были применены в проекте по модернизации вычислительной системы ЗАО «РНТ» г. Москва.

**Соответствие паспорту специальности.** Содержание диссертации соответствует п.1 «Теоретический анализ и экспериментальное исследование функционирования вычислительных машин, комплексов и компьютерных сетей с целью улучшения их технико-экономических и эксплуатационных характеристик» в части создания новых математических моделей и алгоритмов управления процессами восстановления рабочего состояния программ в многомашинной вычислительной системе, построенной на основе технологии виртуальных машин, с целью улучшения её эффективности по двум показателям, среднее время отклика и доля потерянных запросов, и п.6 «Разработка научных методов, алгоритмов и программ, обеспечивающих надежность, контроль и диагностику функционирования вычислительных машин, комплексов и компьютерных сетей» паспорта специальности 05.13.15 – Вычислительные машины, комплексы и компьютерные сети в части разработки методов и программ, обеспечивающих контроль работы многомашинной вычислительной системы, функционирующей при воздействии эффекта «старения» программного обеспечения.

**Достоверность результатов работы** подтверждена серией физических и модельных экспериментов.

**Внедрение результатов работы.** Комплексная методика снижения влияния эффекта «старения» ПО на эффективность функционирования вычислительной системы, была применена в проекте по модернизации вычислительной системы ЗАО «РНТ», обеспечивающей работу более 450 сотрудников компании. Разработанная комплексная

методика позволила добиться более высоких показателей эффективности функционирования ВС по сравнению с ранее использовавшимся решением, а именно снизить долю потерянных запросов пользователей компании при обращении к корпоративным информационным ресурсам и среднее время отклика.

**Апробация работы.** Основные положения диссертационной работы докладывались и обсуждались на заседаниях кафедры «Информационные технологии» МИИТа, 2009-2012 гг., а также на следующих конференциях: на XVII Всероссийской межвузовской научно-практической конференции студентов и аспирантов «Микроэлектроника и информатика» (Зеленоград, МИЭТ) 2010 г.; VII Международной научно-практической конференции "TRANS-MECH-ART-CHEM" (Москва, МИИТ) 2010 г.; VII Международной научно-практической конференции "Перспективы развития информационных технологий" (Новосибирск, НГТУ) 2012 г.; XIX научной конференции "Современные информационные технологии: тенденции и перспективы развития" (Ростов-на-Дону, ЮФУ) 2012 г.; научно-практической конференция «Неделя науки - 2012 "Наука МИИТа - транспорту" (Москва, МИИТ) 2012 г.

**Публикации.** По теме диссертации опубликовано 11 печатных работ, в том числе 6 работ в рецензируемых журналах из перечня ВАК.

**Личный вклад автора.** Все выносимые на защиту научные результаты получены соискателем лично. В работах, опубликованных в соавторстве, лично соискателем разработаны: в [2] – алгоритм расчета времени начала восстановления рабочего состояния программы с учетом требований к её эффективности, в [3] – алгоритм перераспределения ресурсов при размещении группы виртуальных машин, [5] – общая схема компонентов комплексной методики и политика управления процессами восстановления рабочего состояния программ.

**Структура и объем работы.** Диссертация состоит из введения, четырех глав и заключения. Диссертация содержит 127 страниц, 19 рисунков и 2 таблицы. Список литературы насчитывает 86 наименований.

## **СОДЕРЖАНИЕ РАБОТЫ**

**Во введении** обосновывается актуальность исследуемой в диссертации проблемы, формулируются основные цели и задачи исследования.

**В первой главе** даны определения основных терминов и понятий, используемых далее в работе. Проводится анализ причин возникновения и особенностей эффекта «старения», а также его негативного воздействия на работу программного обеспечения.

Исследуются существующие решения по борьбе с эффектом «старения», включая решения, ориентированные на применение в виртуальной ИТ-инфраструктуре. Проведенный анализ показал, что наиболее перспективным является применение проактивного подхода к борьбе с эффектом «старения». В основе решений, реализующих данный подход, лежат две группы методов, которые и определяют их достоинства и недостатки: методы восстановления рабочего состояния программы и методы определения времени начала восстановления.

Анализ широко используемых методов восстановления рабочего состояния программы показал, что основным их недостатком является остановка процесса выполнения программы. Снижение длительности остановки программы достигается, главным образом, за счет модификации исходного кода программы. Кроме того, проведенный анализ технологии виртуальных машин выявил дополнительные

трудности решения задачи восстановления рабочего состояния программы, связанные, прежде всего, с размещением нескольких виртуальных машин на одном компьютере.

При анализе существующих методов определения времени начала восстановления рабочего состояния программы были выделены две группы методов, обладающих различными свойствами: методы, не требующие мониторинга работы программы и/или ОС, и методы, основанные на мониторинге. Основными достоинствами методов первой группы является относительно невысокий объем исходных данных и возможность учета воздействия метода восстановления на целевую программу. Однако основной проблемой методов данной группы является то, что их результаты быстро становятся непрактичными в случае наличия определенной свободы в использовании программы и/или ОС, например, установки обновлений ОС. Методы второй группы являются более устойчивыми к изменениям в работе программы и/или ОС. Основные недостатки методов данной группы заключаются в отсутствии возможности планирования процессов восстановления и высокой трудоемкости их применения. Кроме того, методы обеих групп не учитывают условий выполнения программы и не обеспечивают согласования процессов восстановления различных программ.

На основе результатов проведенного анализа сделан вывод, что в настоящее время не существует открытого законченного решения повышения эффективности ВС, построенной на основе технологии виртуальных машин, при воздействии на неё эффекта «старения» ПО.

Современные организации, как правило, имеют вычислительную систему, работающую по принципу "клиент-сервер" и включающую совокупность серверов, предоставляющих различные сервисы, например, Web-сервер, сервер приложений, сервер баз данных, файл-сервер. Для оценки эффективности работы сервера используются различные показатели: время отклика (мс), производительность (количество запросов/с), коэффициент готовности (в процентах), доля обработанных запросов (в процентах), стоимость. В качестве основных показателей в диссертационной работе выбраны время отклика и доля потерянных запросов, т.к. именно они, главным образом, являются наиболее значимыми как для пользователей, так и владельцев ВС.

Как итог, опираясь на результаты проведенного исследования проблемы, раскрываются цели и задачи диссертационной работы.

**Вторая глава** посвящена разработке методов, направленных на решение ряда задач, возникающих при борьбе с эффектом «старения» ПО, а именно:

- восстановление рабочего состояния программы;
- определение времени начала восстановления программы;
- планирование процессов восстановления рабочего состояния.

Особо отмечается, что выделенные задачи должны решаться комплексно и подчиняться одной цели – снижение негативного воздействия эффекта «старения» ПО на эффективность работы ВС.

При решении задачи восстановления рабочего состояния программы предложено ориентироваться, прежде всего, на наиболее распространенные и уязвимые элементы ВС с точки зрения эффекта «старения» ПО, а именно:

- Сервер (объект типа 1). Под сервером рассматривается программа, выполняющая обслуживание пользователей через сеть.
- Платформа виртуализации (объект типа 2). Под платформой виртуализации



рассматривается гипервизор и набор вспомогательных компонентов, обеспечивающих контроль и управление виртуальными машинами.

Для каждого типа объекта разработаны различные методы восстановления рабочего состояния программы (далее - методы восстановления) с учетом их индивидуальных особенностей, а именно объект типа 1 ориентирован на предоставление пользователям различных сервисов, объект типа 2 ориентирован на управление и контроль виртуальными машинами. Для объекта типа 1 предложен метод подмены виртуальной машины (Virtual Machine Substitution, VMS). Для объекта типа 2 - метод перезапуска платформы виртуализации с перемещением виртуальных машин (Virtual Machine Monitor Rejuvenation with virtual machine migration, VMMR).

Основная идея метода VMS заключается в подмене виртуальной машины с сервером на новую, с постепенным переносом работы на неё. Для работы метода создается копия виртуальной машины с сервером (далее дублирующая виртуальная машина), которая размещается в хранилище виртуальных машин и находится в неактивном состоянии до начала процесса восстановления. Процесс восстановления состоит из трёх этапов:

1. **Начальная конфигурация виртуальных машин.** На данном этапе выполняется запуск дублирующей виртуальной машины на хосте с достаточным объемом ресурсов и согласование времени начала процесса переноса обслуживания пользователей с основной виртуальной машины на дублирующую.
2. **Перенос обслуживания пользователей.** Процесс переноса обслуживания пользователей строится на основе перераспределения запросов между основной и дублирующей виртуальными машинами: запросы в рамках новых соединений направляются на дублирующую, а остальные – на основную. Второй этап завершается после завершения работы пользователей с основной виртуальной машиной, т.е. когда все соединения пользователей с ней будут закрыты.
3. **Заключительная конфигурация виртуальных машин.** На данном этапе выполняется смена ролей, а именно: дублирующая виртуальная машина назначается основной и продолжает обслуживание пользователей, в то время как первая машина отключается для высвобождения ресурсов.

Принцип работы метода VMMR заключается в освобождении платформы виртуализации от управления виртуальными машинами на время её восстановления. Освобождение выполняется с использованием технологии «горячей» миграции виртуальной машины, которая обеспечивает сохранение активности виртуальной машины при её перемещении между хостами. Блок-схема алгоритма восстановления рабочего состояния платформы виртуализации методом VMMR представлена на рисунке 1.

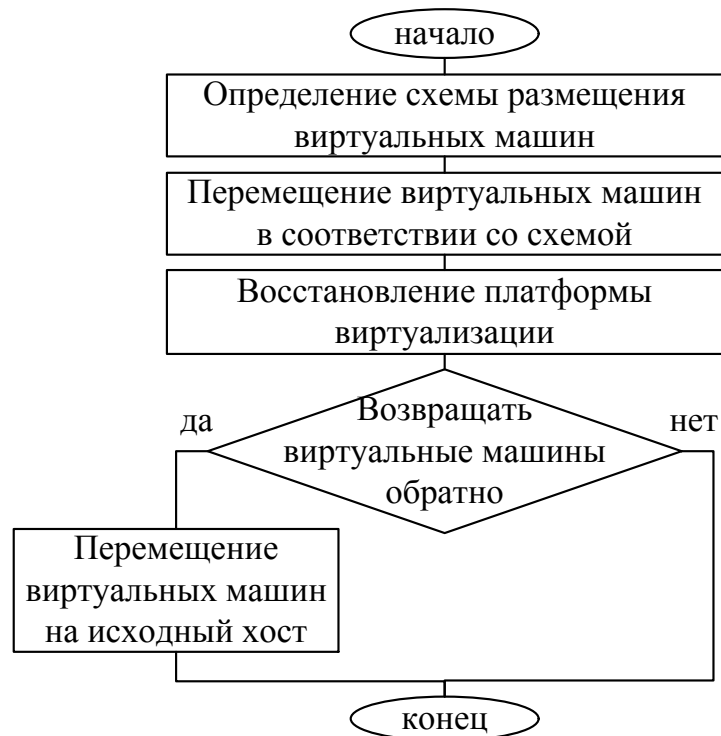


Рисунок 1. Блок-схема алгоритма восстановления рабочего состояния платформы виртуализации методом VMMR

Процесс восстановления методом VMMR включает следующие этапы:

1. **Определение схемы размещения виртуальных машин.** Схема размещения виртуальных машин задает места их нового размещения и порядок перемещения. Схема формируется с использованием разработанного метода планирования процессов восстановления RPRR.
2. **Перемещение виртуальных машин.** На данном этапе выполняется перемещение виртуальных машин в соответствии с подготовленной схемой размещения.
3. **Восстановление платформы виртуализации.** После завершения освобождения платформы виртуализации выполняется её перезапуск.
4. **Возвращение виртуальных машин на исходный хост.** Данный этап предполагает возможность возвращения виртуальных машин на исходный хост на основе технологии «горячей» миграции.

Для обеспечения высокой гибкости работы метода VMMR в условиях дефицита ресурсов используются приоритеты перемещения виртуальных машин, которые характеризуют их важность для ВС с точки зрения сохранения их активности в процессе восстановления.

Для решения задачи определения времени начала восстановления для каждого типа объектов также разработаны отдельные методы с учетом отмеченных ранее их индивидуальных особенностей. Для объекта типа 2 предложен метод Condition-Based Rejuvenation (CBR). Для объекта типа 1 - метод Requirement-Oriented Rejuvenation (ROR).

Работа метода CBR строится на основе выявления характеристики работы целевой программы и/или ОС, наиболее стабильно отражающей процесс «старения», и предусматривает определение времени начала восстановления на основе её мониторинга. Отличительной особенностью метода CBR от существующих методов

является возможность учета изменения условий работы программы в процессе выполнения, что позволяет снизить количество потерянных запросов к ВС в процессе восстановления. Метод CBR состоит из двух этапов:

1. Этап подготовки метода CBR включает следующие шаги:
  - Выявление индикатора «старения». Под индикатором «старения» понимается характеристика работы ОС или целевой программы (например, объем свободной оперативной памяти, количество файловых дескрипторов), наиболее стабильно отражающая процесс «старения». Выбор режима работы метода (описание режимов приведено ниже).
  - Расчет нижней границы запуска процесса восстановления. Под нижней границей понимается такое значение индикатора «старения», начиная с которого допускается запуск процесса восстановления. Время начала восстановления впоследствии уточняется с учетом заданного режима работы метода.
  - Расчет момента запуска процесса определения времени начала восстановления.
2. Этап определения времени начала восстановления. На данном этапе осуществляется мониторинг значений индикатора «старения» и расчет времени достижения индикатором нижней границы запуска процесса восстановления с учетом выбранного режима работы метода. Метод CBR предполагает возможность использования более одного индикатора «старения», в таком случае время начала восстановления устанавливается наименьшим по всем используемым индикаторам.

Достаточно часто работа пользователей с ВС является неравномерной и сопровождается интервалами низкой активности, вплоть до её отсутствия. Например, высокая активность пользователей часто проявляется в дневные часы и практически сходит на нет в ночное время. Как результат учет данного обстоятельства позволяет снизить количество потерянных запросов к ВС в процессе восстановления. С этой целью метод CBR предусматривает два режима работы:

**Режим №1. С учетом заданных интервалов времени восстановления.** Данный режим предполагает использование интервалов времени низкой активностью пользователей при работе с ВС, которые характеризуются малым количеством запросов к ВС в единицу времени. Например, такой интервал может быть назначен на ночные часы, которые в некоторых организациях характеризуются малым числом запросов к ВС. Данный режим предусматривает запуск процесса восстановления в ближайший заданный интервал времени восстановления после достижения индикатором «старения» нижней границы запуска процесса восстановления.

**Режим №2. Без учета заданных интервалов времени восстановления.** Применение данного режима предполагается при невозможности задания интервалов времени восстановления. Данный режим предусматривает определение времени начала восстановления только с учетом длительность процесса восстановления.

Работа второго разработанного метода ROR строится на основе мониторинга объема работы, выполненной программой с момента её запуска, а именно количества обработанных запросов. Отличительной особенностью разработанного метода от существующих методов является возможность учета требований к работе программы по двум показателям эффективности: время обработки запросов и коэффициент готовности. Установленные значения требований характеризуют приемлемый уровень изменения каждого из показателей эффективности. Метод ROR включает два этапа:

1. Этап подготовки. На данном этапе выполняется построение функции изменения

времени обработки запросов от объема работы, выполненной программой, на основе исторических данных о её работе. Для этой цели используется регрессионный сплайн со штрафной функцией, по причине его хорошей применимости к зашумленным данным, возможностью управления величиной сглаживания, а также быстрой и простой настройки.

2. Этап определения времени начала восстановления. Логика работы метода на данном этапе состоит в следующем:

- время начала восстановления устанавливается равным времени, когда расчетное время обработки запросов ( $b_1$ ) достигает установленного требования по данному показателю ( $a_1$ ), при условии, что расчетный коэффициент готовности ( $b_2$ ) больше или равен соответствующему ему требованию ( $a_2$ ):

$$b_1 = a_1, b_2 \geq a_2. \quad (1)$$

- в остальных случаях происходит нарушение хотя бы одним из показателей установленных требований. Особенность поведения выбранных показателей эффективности под воздействием эффекта «старения» ПО заключается в том, что улучшение одного из них ведет к ухудшению другого. В связи с этим требуется поиск компромиссного решения, которое обеспечивало бы наименьшее отклонение выбранных показателей эффективности от установленных к ним требований. Поиск такого решения выполняется на основе метода идеальной точки, который предполагает поиск наилучшего решения с точки зрения близости к некоторой заданной (идеальной) точке. Здесь идеальная точка задается установленными требованиями к эффективности программы. Расстояние между точками пространства критериев поиска определяется по следующей формуле:

$$r(A, C) = \min_{i=1,2} |a_i - c_i|, \quad (2)$$

где  $a_1, a_2$  – требования к времени обработки запросов и коэффициенту готовности;  $c_1, c_2$  – расчетные значения времени обработки запросов и коэффициента готовности.

С целью согласования процессов восстановления рабочего состояния различных программ в рамках комплексной методики был разработан метод планирования процессов восстановления с перераспределением ресурсов (Rejuvenation Planning with Resource Reallocation, RPRR). Задача планирования процессов восстановления представляет собой задачу многокритериальной оптимизации:

$$\varphi(Y(t_{calc}^d, t_{pl}^d, T_{rj}^d), Z(M)) \rightarrow \min. \quad (3)$$

где  $Z(M)$  - доля потерянных запросов;

$Y(t_{calc}^d, t_{pl}^d, T_{rj}^d)$  - среднее время отклика;

$M$  – количество виртуальных машин, активность которых сохраняется в процессе восстановления;

$t_{calc}^d$  - время запуска процесса восстановления объекта  $d$ , рассчитанное на основе метода определения времени начала восстановления;

$t_{pl}^d$  - время запуска процесса восстановления объекта  $d$ , рассчитанное с учетом взаимного влияния процессов восстановления и состояния ресурсов ВС;

$T_{rj}^d$  - длительность процесса восстановления объекта  $d$ .

Основной трудностью этой задачи, кроме многокритериальности, является необходимость принять во внимание ряд особенностей, связанных с реализацией процессов восстановления:

- по мере реализации запланированных процессов восстановления выполняется добавление в план новых процессов восстановления;
- по мере работы объектов восстановления параметры их восстановления (например, время начала восстановления) или состояние ресурсов ВС могут меняться, что приводит к необходимости проверки параметров остальных процессов восстановления (например, время начала восстановления, порядок перемещения виртуальных машин, место размещения виртуальной машины) и их корректировки в случае необходимости.

Планирование процессов восстановления представляет собой поиск места размещения виртуальных машин с целью размещения наибольшего их количества ( $M$ ) для каждого объекта восстановления, а также обеспечения своевременности реализации процесса восстановления ( $t_{pl} - t_{calc}$ ) и минимизации длительности данного процесса  $T_{rj}$ . Важно отметить, что не все из рассматриваемых критериев поиска являются сравнимыми. Первый критерий, количество размещенных виртуальных машин, т.е. тех чья активность сохраняется в процессе восстановления, является наиболее важным, тогда как два других, своевременность и длительность процесса восстановления, являются относительно сравнимыми.

Разработанный метод основан на разбиении поставленной задачи на более простые подзадачи – определение последовательности перемещения виртуальных машин и определение параметров размещения каждой из них. Блок-схема алгоритма планирования процессов восстановления приведена на рисунке 2.

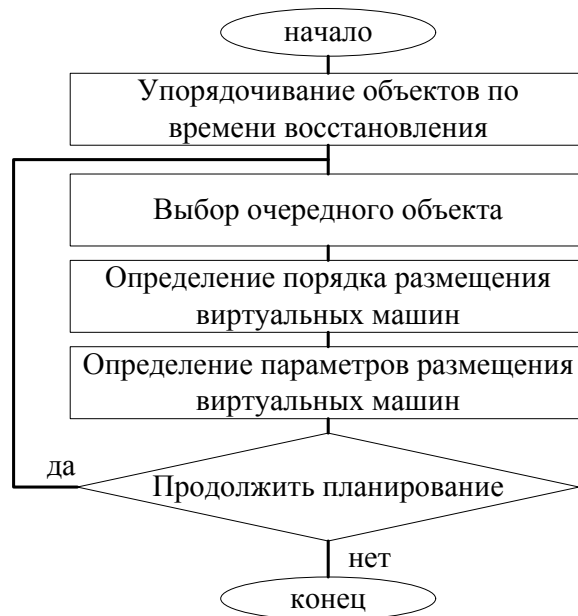


Рисунок 2. Блок-схема алгоритма планирования процессов восстановления

Целью решения первой подзадачи является определение наилучшей последовательности перемещения виртуальных машин с точки зрения возможности размещения наибольшего их количества на соседних хостах. Последовательность формируется в порядке возрастания их требований к ресурсам. При размещении

виртуальных машин могут учитываться их требования по нескольким типам ресурсов, что превращает данную задачу в задачу многокритериальной оптимизации. Для преобразования данной многокритериальной задачи в однокритериальную используется метод линейной свертки критериев:

$$V_{agr}^i = \sum_{j=1}^K c_j V_j^i, \quad (4)$$

где  $V_j^i$  - требование  $i$ -ой виртуальной машины к ресурсу  $j$ -ого типа;

$V_{agr}^i$  - обобщенная оценка требований  $i$ -ой виртуальной машины к ресурсам;

$c_j$  - относительная важность ресурса  $j$ -ого типа.

После формирования последовательности виртуальных машин определяются параметры размещения каждой из них. Особо отмечается, что первостепенным критерием основной задачи является количество виртуальных машин, активность которых сохраняется в процессе восстановления (далее «количество активных виртуальных машин»). Важности двух других критериев - своевременность и длительность процесса восстановления - являются относительно сравнимыми. В соответствии с этим используется метод линейной свертки критериев для их объединения. Затем используется метод лексикографического упорядочивания критериев, который определяет приоритет критерия «количество активных виртуальных машин» перед критерием, полученным на основе метода линейной свертки.

Разработанный алгоритм определения параметров размещения виртуальной машины учитывает возможность перераспределения ресурсов ВС, которое выполняется на основе технологии «горячей» миграции. Блок-схема разработанного алгоритма приведена на рисунке 3.

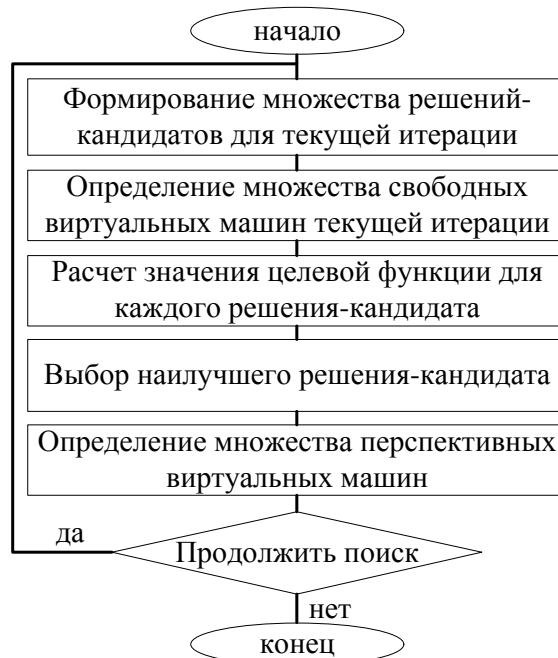


Рисунок 3. Блок-схема алгоритма определения параметров размещения виртуальной машины

Разработанный алгоритм включает следующие шаги:

1. **Формирование множества решений-кандидатов для текущей итерации.** Решение-кандидат представляет собой хост, перспективный с точки зрения размещения заданной виртуальной машины, и порядок перераспределения ресурсов, определяющий объем ресурсов хоста, который может быть использован для её размещения.
2. **Формирование множества свободных виртуальных машин.** Множество свободных виртуальных машин представляет собой виртуальные машины, которые могут быть задействованы при перераспределении ресурсов на последующих итерациях. Данное множество формируется на основе исключения из множества свободных виртуальных машин предшествующей итерации множества виртуальных машин, задействованных при формировании решений-кандидатов.
3. **Расчет значений целевой функции для каждого решения-кандидата.** Для каждого решения-кандидата выполняется расчет значения целевой функции, которая представляет собой линейную свертку двух критериев, своевременность и длительность процесса восстановления:

$$w_u^q = |t_{calc}^{d,u} - t_{pl}^{d,u}| + T_{rj}^{d,u}, \quad (5)$$

где  $w_u^q$  - значение целевой функции решения-кандидата  $u$  итерации  $q$ ;

$t_{calc}^{d,u}$  - время начала восстановления объекта  $d$ , рассчитанное на основе метода определения времени начала восстановления;

$t_{pl}^{d,u}$  - время начала восстановления объекта  $d$ , рассчитываемое с учетом взаимного влияния процессов восстановления и состояния ресурсов ВС;

$T_{rj}^{d,u}$  - длительность процесса восстановления объекта  $d$ .

4. **Выбор наилучшего решения-кандидата.** Выбор наилучшего решения-кандидата включает проверку двух условий:

- Проверка реализации решения-кандидата. Реализация решения-кандидата считается успешной, если связанные с ним процессы перераспределения ресурсов и размещения заданной виртуальной машины, не приводят к превышению базового уровня ресурсов хостов, а именно для любого  $i$ -ого хоста ( $i=1, \dots, NH$ ) выполняется:

$$\begin{cases} \sum_{b=0}^{NV-1} Q_j^b x_i^b \leq H_j^i - h_j^i, j = 1, \dots, NR \\ \forall b \in [1, NV]: x_i^b \in \{0,1\} \end{cases}, \quad (6)$$

где  $Q_j^b$  - требование  $b$ -ой виртуальной машины к  $j$ -ому ресурсу;

$H_j^i$  - базовый уровень  $j$ -ого ресурса  $i$ -ом хосте;

$h_j^i$  - требование платформы виртуализации  $i$ -ого хоста к  $j$ -ому ресурсу;

$x_i^b$  - элемент двоичной матрицы, принимающий значение «1», если  $b$ -ая ВМ размещается на  $i$ -ой хосте, «0» - в противном случае;

$NH$  – число хостов;

$NV$  – число виртуальных машин;

$NR$  – число типов ресурсов, по которым учитываются при размещении виртуальных машин.

- Решение-кандидат является наилучшим, если значение его целевой функции является наименьшим:

$$w_{min}^q = \min_{u \in U_q} \{w_u^q, w_{min}^{q-1}\}, \quad (7)$$

где  $w_{min}^q$  - наилучшее значение целевой функции после итерации  $q$ ;

$w_u^q$  - значение целевой функции для решения-кандидата  $u$  итерации  $q$ ;

$U_q$  - множество решений-кандидатов итерации  $q$ .

5. **Формирование множества перемещаемых виртуальных машин.** На данном шаге выявляются виртуальные машины, которые будут использованы при поиске решений-кандидатов на следующей итерации. Данное множество формируется на основе множества свободных виртуальных машин, для которых существуют решения-кандидаты текущей итерации, удовлетворяющие их требованиям к ресурсам.

Таким образом, в данной главе были разработаны основные составляющие комплексной методики. В завершении отмечаются преимущества разработанных методов восстановления рабочего состояния программы для платформы виртуализации (метод VRRM) и сервера (метод VMS) перед существующими методами:

- обеспечивают восстановление вне зависимости от источника эффекта «старения» ПО;
- не приводят к прерыванию обслуживания пользователей в процессе восстановления;
- не требуют модификации исходного кода восстанавливаемой программы.

Набор методов определения времени начала восстановления:

- метод CBR основан на мониторинге характеристик выполнения целевой программы и/или ОС и отличается от существующих методов тем, что ориентирован на определение времени начала восстановления платформы виртуализации и учитывает характера изменения условий её работы;
- метод ROR основан на мониторинге объема работы, выполненной целевой программы, и отличается от существующих методов тем, что ориентирован на определение времени начала восстановления сервера и обеспечивает учет требований к эффективности его работы по двум показателям – время обработки запросов и коэффициент готовности.

Метод планирования процессов восстановления RPRR обладает следующими преимуществами перед существующими методами:

- возможность согласования процессов восстановления различных программ с учетом текущего состояния ВС;
- учет трех показателей, характеризующих эффективность процессов восстановления: количество виртуальных машин, активность которых сохраняется в процессе восстановления, своевременность и длительность процесса восстановления;
- учитывает возможность перераспределения ресурсов ВС при реализации процессов восстановления рабочего состояния.

**Третья глава** посвящена разработке общей схемы взаимодействия компонентов разработанной методики и политики управления процессами восстановления. Разработанная комплексная методика включает в себя набор методов, приведенный на рисунке 4.



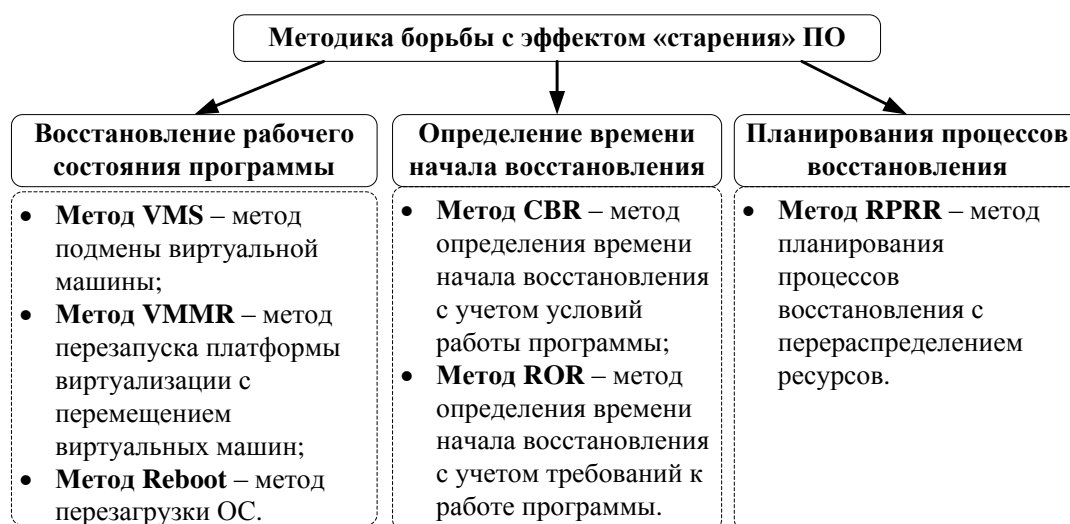


Рисунок 4. Набор методов комплексной методике

Основные задачи, решаемые в рамках комплексной методике, были разбиты на множество компонентов и определена общая схема их взаимодействия, представленная на рисунке 5. В предложенной общей схеме присутствуют следующие компоненты:

- **Блок мониторинга работы объектов восстановления (Блок №1).** Данный блок отвечает за сбор данных о работе объектов восстановления. Конкретный набор данных определяется методом определения времени начала восстановления, используемым для заданного объекта.
- **Блок мониторинга ВС (Блок №2).** Данный блок отвечает за сбор данных о состоянии ресурсов на хостах, распределение виртуальных машин на хостах и требования виртуальных машин к ресурсам.
- **Блок определения времени начала восстановления объекта типа 2 (Блок №3).** Основой данного блока является метод CBR, который, главным образом, и определяет порядок его работы.
- **Блок определения времени начала восстановления объекта типа 1 (Блок №4).** Данный блок предназначен для определения времени начала восстановления объекта типа 1 при использовании метода VMS.
- **Блок согласования процессов восстановления (Блок №5).** В основе работы блока лежит разработанный метод планирования процессов восстановления RPRR, который и определяет порядок его работы.
- **Блок корректировки плана восстановления (Блок №6).** Данный блок отвечает за корректировку плана восстановления в отношении объектов типа 1, для которых назначен метод восстановления - перезагрузка ОС.
- **Блок контроля процессов восстановления (Блок №7).** Данный блок отвечает за реализацию плана восстановления, а именно запуск и контроль выполнения процессов восстановления предусмотренными комплексной методикой методами восстановления рабочего состояния программы, а именно методом VMS, методом VMMP и методом Reboot.
- **Блок подготовки набора параметров метода ROR (Блок А).** Данный блок отвечает за подготовку параметров метода ROR. Порядок подготовки данных в блоке соответствует этапу подготовки метода ROR.
- **Блок подготовки набора параметров метода CBR (Блок Б).** Данный блок отвечает за подготовку параметров CBR. Порядок подготовки данных в блоке соответствует

этапу подготовки метода CBR.

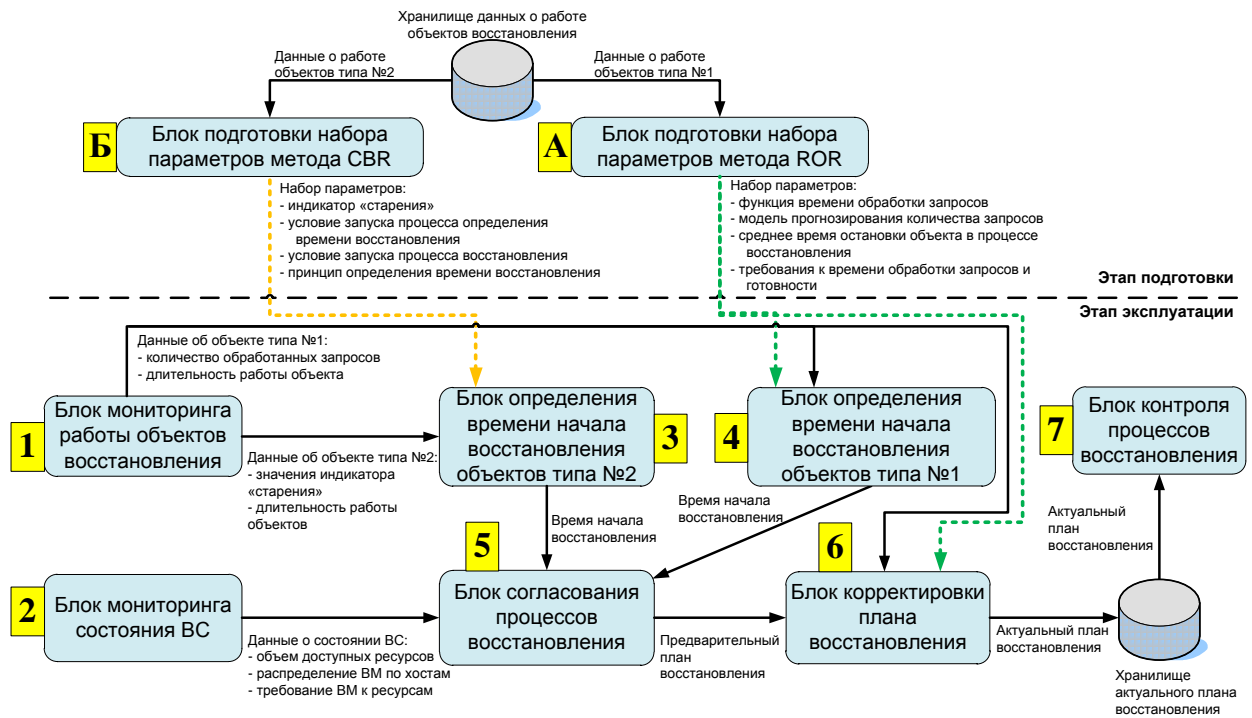


Рисунок 5. Общая схема взаимодействия компонентов методики

В разработанной методике выделены два этапа:

- **Подготовительный этап.** На данном этапе выполняется подготовка комплексной методики в соответствии с этапами подготовки, установленными для каждого из методов, входящих в её состав.
- **Этап эксплуатации.** На данном этапе выполняется мониторинг объектов восстановления, сбор информации о состоянии ВС, подготовка плана восстановления, поддержание его в актуальном состоянии и его реализация.

Управления процессами восстановления осуществляется в одном из двух режимов:

1. **Режим мониторинга.** В данном режиме выполняется мониторинг работы объектов восстановления и определение времени начала восстановления, а также мониторинг состояния ВС. В режиме мониторинга может выполняться обновление плана восстановления при возникновении одного из следующих событий:
  - a. добавлен или удален объект восстановления;
  - b. произошло изменение состояния ресурсов хостов, распределения виртуальных машин по хостам или их требований к ресурсам;
  - c. произошло изменение времени начала восстановления объекта.
2. **Режим восстановления.** Данный режим активируется после запуска одного из процессов восстановления и действует до тех пор, пока существует хотя бы один активный процесс восстановления. В режиме восстановления действует запрет на операцию обновления плана восстановления. Управление процессами восстановления выполняется в циклическом режиме, а именно, после завершения восстановления объекта выполняется его добавление в план восстановления. После завершения режима восстановления управление процессами восстановления возвращается в режим мониторинга.

Таким образом, в данной главе была решена следующая поставленная в диссертационной работе задача: разработка общей схемы взаимодействия компонентов методики и политики управления процессами восстановления, обеспечивающие формирование целостного решения. Подготовленное решение отличается от существующих тем, что учитывает особенности технологии виртуальных машин и позволяет улучшить эффективность ВС по двум показателям: среднее время отклика и доля потерянных запросов.

**В четвертой главе** представлена реализация разработанной комплексной методики. Приведены рекомендации по применению данной методики и её отдельных компонентов. Описаны постановка и проведение серии экспериментов по оценке предложенной методики с точки зрения эффективности работы ВС по двум показателям (среднее время отклика и доля потерянных запросов), а также приведены результаты экспериментов.

Основная цель разработки программного комплекса состоит в апробации и подтверждении достоверности полученных научных результатов. Компонентная архитектура комплекса включает главный модель, модуль сбора данных №1, ответственный за мониторинг объектов восстановления, модуль сбора данных №2, ответственный за сбор данных о состоянии ВС, модуль подготовки объектов восстановления, модуль математической обработки, ответственный за формирование плана восстановления и поддержания его в актуальном состоянии, и модуль контроля процессов восстановления. Кроме того, в структуру системы входят специальные агенты, размещаемые на стороне объектов восстановления и осуществляющие сбор данных и реализацию метода восстановления VMS.

После реализации комплексной методики была проведена серия экспериментов, целями которых были:

- Сравнить эффективность работы разработанной методики с существующими решениями, не требующими внесения изменений в исходный код восстанавливаемой программы, по двум критериям – среднее время отклика и доля потерянных запросов.
- Исследовать влияние разработанной методики на работу ВС и предложить рекомендации по её использованию. Для этого была проведена серия экспериментов с различными данными, определяющими интенсивность «старения» и объем ресурсов ВС.
- Подтвердить преимущества разработанного метода планирования процессов восстановления по сравнению с использованием для этой цели классического подхода к определению места размещения виртуальных машин. Была проведена серия модельных экспериментов с варьированием объема ресурсов на хостах.

В рамках сравнительной оценки эффективности была выполнена серия физических экспериментов. Был рассмотрен сложный случай, когда эффекту «старения» ПО подвержены одновременно оба типа объектов. Для проведения экспериментов подготовлены два сценария работы объектов восстановления, отличавшиеся скоростью процесса «старения». Каждый сценарий определяется набором значений параметров работы объектов восстановления, определяющих эффект «старения» ПО. Для объекта типа 1 таковыми параметрами выступают параметры генератора трафика (время между соединениями, количество запросов в соединении, количество связанных запросов, время

между запросами, время между связными запросами), для объекта типа 2 - интервал вызова проблемной функции программы, подверженной эффекту «старения» ПО. Сценарии отличались такими параметрами как время между соединениями и интервалом времени вызова проблемной функции (для второго сценария указанные параметры были уменьшены на 10% в сравнении с первым).

Сравнение проводилось с решениями, которые не требуют внесения изменений в исходный код восстанавливаемых программ:

- Решение 1 представляет собой комбинацию метода перезагрузки ОС для восстановления объектов типа 1 и 2, и метода определения времени начала восстановления на основе мониторинга их работы (Метод Li).
- Решение 2 представляет собой комбинацию метода перезагрузки ОС для восстановления объектов типа 1 и 2, и метода определения времени начала восстановления без мониторинга их работы (Метод Dohi).
- Решение 3 представляет собой комбинацию метода перезапуска программы для восстановления объекта типа 1 и метода перезагрузки ОС для объекта типа 2 с методом определения времени начала восстановления на основе мониторинга их работы (Метод Li).
- Решение 4 основывается на быстром перезапуске виртуальной машины после обнаружения отказа.

Была выполнена подготовка к работе сравниваемых решений и комплексной методики и их тестирование на каждом из сценариев.

На рисунке 6(a) показаны полученные значения доли потерянных запросов, а на рисунке 6(b) – значения среднего времени отклика (для двух случаев: в ВС существует достаточный объем ресурсов для реализации процессов восстановления (результаты разработанной методики отмечены как «Методика»); 2) в ВС недостаточно ресурсов для реализации процессов восстановления (отмечены как «Методика огр.»).

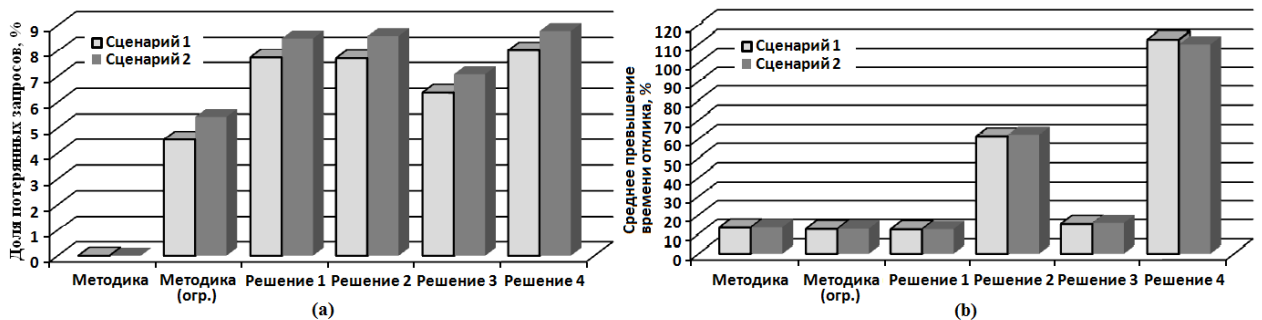


Рисунок 6. Результаты физических экспериментов

Анализ результатов тестирования показал, что разработанная комплексная методика при наличии достаточного объема свободных ресурсов ВС для реализации процессов восстановления предусмотренными в ней методами VMS и VMMR обеспечивает существенно лучшие результаты, а именно полностью исключает потерю запросов и обеспечивает среднее время отклика на уровне лучших результатов среди сравниваемых решений (среднее превышение времени отклика составляет не более 10%). В случае отсутствия достаточного объема ресурсов ВС разработанная методика демонстрирует более чем на 30% лучший результат по показателю доля потерянных запросов, а среднее время отклика на уровне лучших результатов сравниваемых в диссертации решений.

Серия физических экспериментов показала, что на эффективность предложенной методики по показателю количество потерянных запросов существенно влияет объем свободных ресурсов ВС. Для оценки влияния состояния ресурсов на работу методики была проведена серия модельных экспериментов с использованием специально разработанного автором работы программного обеспечения. Модельный эксперимент проводился для двух вариантов разработанной методики, которые отличались подходами к планированию процессов восстановления:

- первый вариант представлял собой комплексную методику как есть, а именно с использованием разработанного метода планирования процессов восстановления RPRR (далее - «Методика 1»);
- во втором варианте планирование процессов восстановления выполнялось на основе классического подхода к размещению виртуальных машин, без учета возможности перераспределения ресурсов (далее - «Методика 2»).

Моделирование выполнялось для 15 хостов. Для усложнения задачи управления процессами восстановления учитывались два типа ресурсов: оперативная память и процессор. Серии опытов отличались максимальным объемом ресурса, добавляемого к базовому уровню ресурса хоста. Опыты в каждой серии отличались объемом добавляемого ресурса, который выбирался в пределах от 0 до максимального добавляемого объема данной серии в соответствии с законом равномерного распределения. Для каждой серии опытов максимальный объем добавляемого ресурса повышался на 5% и рассчитывался по формуле:

$$dR_{i,j} = 5iR'_j / 100, \quad (8)$$

где  $dR_{i,j}$  - максимальный добавляемый объем ресурса  $j$ -ого типа в  $i$ -ой серии;

$R'_j$  - базовый уровень  $j$ -ого типа ресурса.

Результаты моделирования приведены на рисунке 7.

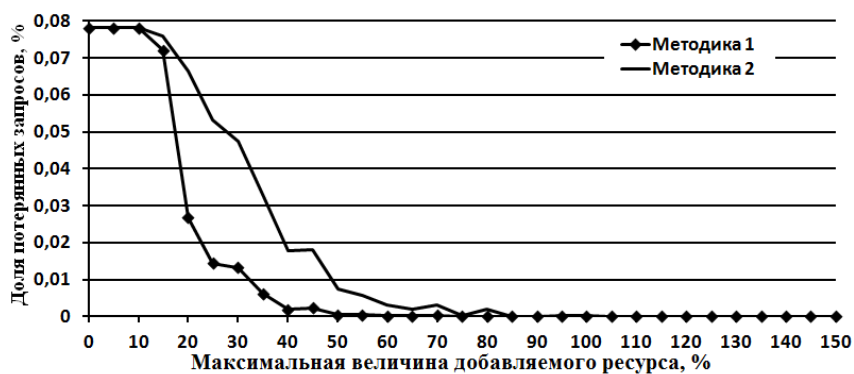


Рисунок 7. Результаты моделирования для различных вариантов методики

Результаты экспериментов показали, что комплексная методика с разработанным методом планирования процессов восстановления RPRR лучше справляется с задачей управления процессами восстановления и обеспечивает снижение доли потерянных запросов в среднем более чем на 60% в условиях дефицита ресурсов ВС по сравнению с использованием в ней классического подхода к размещению виртуальных машин.

В завершении главы был подведен итог проведенным экспериментам:

- Разработанная методика при достаточном объеме свободных ресурсов ВС для

реализации процессов восстановления предусмотренными в методике методами VMS и VMMR обеспечивает существенно лучшие результаты, а именно исключает потерю запросов и обеспечивает среднее время отклика на уровне лучших результатов среди сравниваемых решений (среднее превышение времени отклика составляет не более 10%).

- Разработанная методика при отсутствии достаточного объема ресурсов ВС для реализации процессов восстановления предусмотренными в методике методами VMS и VMMR обеспечивает снижение доли потерянных запросов более чем на 30% и среднее время отклика на уровне лучших результатов сравниваемых в диссертации решений.
- Используемый в разработанной методике метод планирования процессов восстановления RPRR учитывает возможность перераспределения ресурсов ВС, тем самым обеспечивая снижение доли потерянных запросов в среднем более чем на 60% по сравнению с использованием классического подхода к размещению виртуальных машин. Таким образом, нужно отметить, что наличие достаточного объема свободных ресурсов является условием высоких результатов разработанной комплексной методики.

### ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В процессе исследований, выполненных в диссертационной работе, получены следующие результаты:

1. Разработан набор методов восстановления рабочего состояния программы для платформы виртуализации (метод VMMR) и сервера (метод VMS), которые обладают следующими преимуществами перед существующими методами восстановления рабочего состояния программы:
  - обеспечивают восстановление вне зависимости от источника эффекта «старения» ПО;
  - не приводят к прерыванию обслуживания пользователей в процессе восстановления;
  - не требуют модификации исходного кода восстанавливаемой программы.
2. Разработан набор методов определения времени начала восстановления:
  - метод CBR основан на мониторинге характеристик выполнения целевой программы и/или ОС и отличается от существующих методов тем, что ориентирован на определение времени начала восстановления платформы виртуализации и обеспечивает учет характера изменения условий работы платформы виртуализации;
  - метод ROR основан на мониторинге объема работы, выполненной целевой программы, и отличается от существующих методов тем, что ориентирован на определение времени начала восстановления сервера и обеспечивает учет требований к эффективности его работы по двум показателям – время обработки запросов и коэффициент готовности.
3. Разработан метод планирования процессов восстановления RPRR, который обладает следующими преимуществами перед существующими методами:
  - обеспечивает согласование процессов восстановления различных программ;
  - учитывает три показателя, характеризующих эффективность процессов восстановления: количество виртуальных машин, активность которых сохраняется в процессе восстановления, своевременность и длительность

- процесса восстановления;
- учитывает возможность перераспределения ресурсов ВС при реализации процессов восстановления рабочего состояния.
4. Разработана общая схема взаимодействия компонентов методики и политика управления процессами восстановления, обеспечивающие формирование целостного решения, которое отличается от существующих решений тем, что учитывает особенности технологии виртуальных машин и позволяет улучшить эффективность ВС по двум показателям: среднее время отклика и доля потерянных запросов.
  5. Предложена схема программного комплекса, реализующего разработанную методику, и выполнена её программная реализация.
  6. Проведённая серия экспериментов показала, что разработанная комплексная методика превосходит по выбранным показателям (среднее время отклика и доля потерянных запросов) сравниваемые в диссертации решения. При наличии достаточного объема свободных ресурсов ВС для реализации процессов восстановления предусмотренными в разработанной методике методами VMS и VMMR она позволяет полностью исключить потерю запросов и обеспечить среднее время отклика на уровне лучших результатов среди сравниваемых решений (среднее превышение времени отклика составляет не более 10%). В случае отсутствия достаточного объема ресурсов ВС разработанная методика демонстрирует более чем на 30% лучший результат по показателю доля потерянных запросов, а среднее время отклика на уровне лучших результатов сравниваемых в диссертации решений.

#### **Основные результаты диссертации опубликованы в работах:**

В изданиях из перечня ВАК:

1. Удовиченко, А.О. Проблема «старения» программного обеспечения и пути её решения / А.О. Удовиченко // Информатизация и связь. - 2012. - №1. - С.17–20.
2. Удовиченко, А.О. "Метод определения времени восстановления рабочего состояния приложения с учетом требований к его эффективности" / В.П. Соловьёв, А.О. Удовиченко // Информатизация и связь. - 2012. - №2. - С.61–66.
3. Удовиченко, А.О. Метод планирования размещения группы виртуальных машин с перераспределением ресурсов / В.П. Соловьёв, А.О. Удовиченко // Программные продукты и системы. - 2012. - №1. - С.134–138.
4. Удовиченко, А.О. Методы восстановления рабочего состояния приложения / А.О. Удовиченко // Программные продукты и системы. - 2012. - №2. - С.113–117.
5. Удовиченко, А.О. Комплексная методика борьбы с эффектом «старения» ПО / Н.Н. Пуцко, А.О. Удовиченко // Программная инженерия. - 2012. - №4. - С.13–18.
6. Удовиченко, А.О. Метод определения времени восстановления приложения, учитывающий условия его работы / А.О. Удовиченко // Информационные системы и технологии. - 2012. - №6. - С.5-15.

В других изданиях:

7. Удовиченко, А.О. Разработка метода восстановления рабочего состояния серверного программного обеспечения на основе технологии виртуальных машин / А.О. Удовиченко // Труды VII Международная научно-практическая конференция «TRANS-MECH-ART-CHEM». – М.: МИИТ, 2010. - С.368-370.
8. Удовиченко, А.О. Эффект накопления ошибок программного обеспечения и методы восстановления его рабочего состояния / А.О. Удовиченко // Тезисы докладов XVII Всероссийская межвузовская научно-практическая конференция студентов и

- аспирантов «Микроэлектроника и информатика». - Зеленоград: МИЭТ, 2010. - С.183.
9. Удовиченко, А.О. Комплексный подход к решению проблемы «старения» программного обеспечения / А.О. Удовиченко // Перспективы развития информационных технологий: сборник материалов VII Международной научно-практической конференции. - Новосибирск: НГТУ, 2012. - С.281-286.
  10. Удовиченко, А.О. Актуальные вопросы борьбы с эффектом «старения» программного обеспечения / А.О. Удовиченко // Современные информационные технологии: тенденции и перспективы развития: сборник материалов XIX научной конференции. - Ростов-на-Дону: ЮФУ, 2012. - С.286-287.
  11. Удовиченко, А.О. Эффективное решение проблемы «старения» программного обеспечения / А.О. Удовиченко // Труды научно-практической конференции Неделя науки - 2012 «Наука МИИТа – транспорту». - М.: МИИТ, 2012.

Удовиченко Антон Олегович

РАЗРАБОТКА КОМПЛЕКСНОЙ МЕТОДИКИ СНИЖЕНИЯ ВЛИЯНИЯ ЭФФЕКТА  
«СТАРЕНИЯ» ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА РАБОТУ  
МНОГОМАШИНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ, ПОСТРОЕННОЙ НА  
ОСНОВЕ ТЕХНОЛОГИИ ВИРТУАЛЬНЫХ МАШИН

Специальность - 05.13.15

Вычислительные машины, комплексы и компьютерные сети