

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи

Клейман Лев Александрович

**ПОВЫШЕНИЕ НАДЕЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ  
НА ОСНОВЕ ДИНАМИЧЕСКОГО РАСПРЕДЕЛЕНИЯ  
ДИАГНОСТИЧЕСКИХ ЗАДАЧ**

2.3.2 – Вычислительные системы и их элементы

Диссертация  
на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук, доцент  
Фрейман Владимир Исаакович

Пермь – 2022

## Оглавление

<b>Введение .....</b>	<b>4</b>
<b>1. Анализ методов контроля и диагностики вычислительных систем.....</b>	<b>12</b>
1.1 Анализ структуры вычислительных систем и их элементов .....	12
1.1.1 Классификация элементов вычислительных систем .....	13
1.1.2 Структура элементов вычислительных систем .....	15
1.1.3 Особенности элементов вычислительных систем (различные условия эксплуатации – автономность) .....	20
1.2 Способы обеспечения надёжности вычислительных систем.....	23
1.2.1 ГОСТ 27.002-2015 Надёжность в технике (ССНТ). Термины и определения .....	23
1.2.2 Способы обеспечения составляющих надёжности вычислительных систем.....	24
1.3 Диагностика и контроль функционирования вычислительных систем как основной способ обеспечения надёжности на этапе эксплуатации .....	27
1.3.1 Классификация систем диагностирования.....	27
1.3.2 Условия применения систем диагностирования .....	29
1.3.3 Подходы к проведению диагностирования вычислительных систем .	32
1.4 Математическая постановка задачи диссертационного исследования .....	36
1.5 Выводы по главе.....	38
<b>2. Разработка моделей встроенной системы диагностирования .....</b>	<b>40</b>
2.1 Анализ существующих моделей системы тестового диагностирования и её элементов .....	40
2.2 Разработка модели встроенной системы тестового диагностирования .....	42
2.3 Разработка модели элемента встроенной системы тестового диагностирования.....	49

2.4 Выводы по главе.....	51
<b>3. Методы определения характеристик моделей элементов встроенной системы диагностирования.....</b>	<b>53</b>
3.1 Метод определения важности критериев работоспособности элемента вычислительной системы.....	53
3.2 Метод определения показателя работоспособности элемента вычислительной системы.....	58
3.3 Выводы по главе.....	64
<b>4. Разработка алгоритма распределения диагностической нагрузки между элементами встроенной системы диагностирования.....</b>	<b>66</b>
4.1 Первый этап. Расчёт численных значений моделей системы и её элементов.....	68
4.2 Второй этап. Определение параметров диагностирования.....	69
4.3 Третий этап. Расчёт значений модели элемента системы диагностирования. Распределение нагрузки. Диагностирование.....	71
4.4 Оценка эффективности разработанного алгоритма.....	72
4.5 Выводы по главе.....	79
<b>5. Апробация и внедрение разработанных моделей, методов и алгоритма в состав программно-аппаратного комплекса встроенной системы диагностирования.....</b>	<b>82</b>
5.1 Внедрение в программно-аппаратный комплекс «Безопасный город».....	82
5.2 Выводы по главе.....	88
<b>Заключение.....</b>	<b>92</b>
<b>Список использованных источников.....</b>	<b>96</b>
<b>Приложение А.....</b>	<b>110</b>
<b>Приложение Б.....</b>	<b>114</b>

## Введение

**Актуальность темы исследования** обусловлена тем, что влияние вычислительных систем на показатели эффективности технологических и организационных процессов в различных сферах экономики и производства постоянно усиливается. Для диверсификации применения вычислительных систем, в том числе и в объектах критической инфраструктуры, необходима активизация политики импортозамещения. Одним из обязательных условий этого является выполнение ими повышенных требований к качеству и надежности функционирования. В результате расширяются возможности для более широкого и эффективного использования отечественного конкурентоспособного технического, информационного, программного обеспечения вычислительных систем и их элементов.

Современные вычислительные системы являются сложными (содержат большое количество компонентов и связей между ними), гетерогенными и мультивендорными. Их основными элементами являются высокопроизводительные процессорные устройства, специализированных интегральных схем (СБИС), устройств с перестраиваемой логикой (ПЛИС) и т.п. Они описываются большим количеством различных характеристик и параметров, каждый из которых влияет как на их собственные показатели надежности, так и на интегральные показатели всей вычислительной системы. Одним из наиболее эффективных инструментов обеспечения заданных высоких требований к надежности и качеству функционирования является математический аппарат и методы технической диагностики. Особенно важным и эффективным является их применение в специфических условиях эксплуатации вычислительных систем – автономная работа элементов, чувствительность к перегрузкам, распределённость и т.д.

Эффективность применения вычислительных систем во многом обуславливает обеспечение заданного качества функционирования тех

процессов, информационно-технической платформой которых они являются. При этом элементы вычислительных систем должны сами отвечать повышенным требованиям к своим качественным и эксплуатационным характеристикам. Для этого необходимо разрабатывать и применять многофункциональные и высокопроизводительные системы диагностирования. Как следствие, возникает *противоречие* – усложнение объектов диагностирования и самой системы диагностики приводит к увеличению объёмов передачи и сложности обработки диагностической (служебной) информации [32], что снижает эффективность взаимодействия элементов вычислительной системы между собой, приводит к их перегрузке, снижает время автономной работы, ухудшает другие показатели. Поэтому актуальной задачей становится разработка и реализация методов диагностирования вычислительных систем и их элементов, не снижающих, а улучшающих их качественные и эксплуатационные характеристики.

**Степень научной разработанности проблемы.** Подходы к решению проблемы качественного функционирования систем и их элементов широко представлены в работах отечественных и зарубежных ученых в области технической диагностики и теории надежности. Особенно можно выделить монографии, научные статьи и публикации Пархоменко П.П., Каравая М.Ф., Согомоняна Е.С., Кона Е.Л., Тюрина С.Ф., Сагунова В.И., Ломакиной Л.С., Гурова С.В., Шишмарева В.Ю., Половко А.М., Verbruggen H., Tzafestas S., Zanni E., Jung D. При этом сохраняют актуальность проблемы разработки и исследования новых диагностических моделей вычислительных систем и их элементов для специфических условий эксплуатации таких систем, учитывающих также особенности различных этапов диагностирования.

Вопросами улучшения показателей надежности вычислительных систем и их элементов путем реализации для них систем диагностирования занимались такие ученые, как Ведешенков В.А., Аминев Д.А., Димитриев Ю.К., Nakimi

S.L., Yang C.L., Masson G.M., Kim W., Braun J.E., Li X., Dowdeswell B. В работах этих ученых рассматриваются различные методы повышения надежности вычислительных систем, основным из которых является использование методов и инструментов встроенного контроля. В этих работах описываются методы и алгоритмы, позволяющие проводить тестирование элементов вычислительных систем с обнаружением и поиском неисправностей, прогнозированием изменений технического состояния. Большинство работ по данной теме рассматривают ошибки диагностирования и проблемы в канале передаче данных как причину для реконфигурации системы диагностирования. Это делает актуальной задачу принятия в расчёт фактора технического состояния элементов как причины для проведения реконфигурации встроенной системы тестового диагностирования.

В работах отечественных ученых Подиновского В.В., Нелюбина А.П., Потапова М.А., Фреймана В.И., а также зарубежных исследователей Edwards W., Barron F.H., Figuera J., Greko S., Ehr Gott M., рассматриваются методики принятия решений в многокритериальных задачах. Развитием направления их научных исследований является разработка и анализ методики определения характеристик работоспособности элементов вычислительных систем.

**Цель и задачи исследования.** Целью исследования является улучшение эксплуатационно-технических показателей вычислительных систем и их элементов на основе эффективного применения системы встроенного диагностирования. Для достижения поставленной цели в диссертационной работе поставлены и решены следующие задачи:

1. Аналитический обзор и анализ моделей, методов и инструментов встроенного диагностирования в вычислительных системах различных конфигураций, технологий и условий эксплуатации, анализ их недостатков, обоснование актуальности проводимых исследований.

2. Разработка диагностической модели элементов вычислительных систем и математической модели встроенной системы диагностирования.

3. Создание методов определения основных характеристик разработанных математических и диагностических моделей.

4. Разработка метода динамического распределения служебной (диагностической) нагрузки между элементами встроенной системы диагностирования.

5. Апробация разработанных моделей и методов, реализованных в структуре информационного и программного обеспечения управляющих и коммуникационных элементов вычислительных систем.

**Объект исследования** – вычислительная система и её элементы, используемые в различных условиях эксплуатации и режимах диагностирования.

**Предметом исследования** являются модели и методы встроенного диагностирования на основе динамического распределения служебной нагрузки между элементами вычислительных систем.

**Научная новизна** заключается в разработанных моделях встроенной системы диагностирования и элементов вычислительных систем, методах расчёта основных показателей этих моделей, а также методе динамического распределения диагностических задач. Новизна научных результатов диссертационного исследования состоит в том, что:

1. Разработаны *модели системы диагностирования и элементов вычислительных систем*. Они *отличаются* от существующих тем, что в них основным фактором для реконфигурации системы диагностирования предложено считать техническое состояние элементов системы. Это *позволяет* учитывать показатели работоспособности элемента при распределении диагностических задач на различных этапах диагностирования.

2. Разработан *метод принятия решений в задаче диагностики элементов вычислительных систем*, позволяющий определить значения *весовых коэффициентов* для принятия решения о динамическом распределении диагностических задач. Он *отличается* от существующих тем, что значительно снижает эффекты компенсации, возникающие при использовании наиболее часто применимых алгебраических и количественных критериев. Это *позволило* повысить точность (объективность) расчёта относительной важности критериев в системе принятия решения.

Разработан *метод определения показателя надёжности элемента вычислительной системы*, исходя из текущих значений критериев работоспособности и значений их важности (весовых коэффициентов). Он *отличается* от существующих тем, что предлагает способ снижения влияния эффектов компенсации при расчёте показателя надёжности элемента вычислительной системы, на основе разработанной геометрической модели описания технического состояния. Это *позволило* учесть множество критериев работоспособности с различными значениями важности; с большей точностью определять показатели разработанных моделей; более эффективно распределять диагностическую информацию и повысить надёжность вычислительных систем.

3. Разработан *метод, позволяющий обоснованно и эффективно распределять диагностическую информацию*. Он *отличается* от существующих тем, что позволяет динамически изменять роли элементов системы встроенного диагностирования, а также учитывать техническое состояние элементов для прогнозирования их возможного состояния на следующем этапе проверки. Это *позволило* увеличить время работоспособности системы и уменьшить время её восстановления.

**Теоретическая значимость работы** заключается в создании моделей, методов и алгоритма работы встроенной системы диагностирования,

позволяющей повысить качественные и эксплуатационные характеристики вычислительных систем и её элементов. Методы, разработанные для расчёта основных характеристик моделей, могут быть использованы для решения других масштабных задач, в которых необходимо определить важность учитываемых критериев или принимать решения в зависимости от технического состояния элемента или системы.

**Практическая значимость работы** состоит в том, что предложенный инструментарий в виде моделей, методов, алгоритмов реализован и внедрен в составе информационного и программного обеспечения инструментария встроенного диагностирования вычислительных систем. По основным показателям отмечено, что внедрение результатов работы позволило увеличить коэффициент использования вычислительной системы в среднем на 8,3% (акт ЗАО «ИВС - Сети»). Также кластеризация отказов позволила уменьшить время восстановления системы на 5% (акт ООО «ПроИнфоСервис»). Результаты работы внедрены в учебный процесс кафедры «Автоматика и телемеханика» (акт ФГАОУ ВО «ПНИПУ»).

**Методология и методы исследования** базируются на математическом аппарате и методах системного анализа, теории вероятности и математической статистики, теории надежности, технической диагностики, аналитического и имитационного моделирования.

**Основные положения, выносимые на защиту:**

1. Математическая модель встроенной системы диагностирования и диагностическая модель элемента вычислительной системы (п. 3).
2. Методы определения основных характеристик математической модели встроенной системы диагностирования и диагностической модели элементов вычислительной системы (п. 3).
3. Метод динамического распределения диагностических задач в вычислительных системах (п. 3).

4. Результаты апробации и внедрения разработанных моделей, методов и алгоритма в состав программно-аппаратного комплекса встроенной системы диагностирования (п. 4).

Основные положения соответствуют пункту 3 «Разработка научных подходов, методов, алгоритмов и программ, обеспечивающих надежность, сбое- и отказоустойчивость, контроль и диагностику функционирования вычислительных систем и их элементов» и пункту 4 «Теоретический анализ и экспериментальное исследование функционирования вычислительных систем и их элементов в нормальных и экстремальных условиях с целью улучшения их технико-экономических и эксплуатационных характеристик» паспорта научной специальности 2.3.2 «Вычислительные системы и их элементы».

**Достоверность и обоснованность результатов.** Полученные в диссертационной работе результаты не противоречат теоретическим положениям, известным из научных публикаций отечественных и зарубежных исследователей, и подтверждаются результатами апробации и внедрения предложенных в диссертации моделей, методов и алгоритма в реальные вычислительные системы.

**Апробация результатов работы.** Основные результаты исследований, выполненных в диссертационной работе, докладывались и обсуждались на научно-технических семинарах, международных и Всероссийских конференциях (8), в т. ч. XIII Всероссийское совещание по проблемам управления (г. Москва, ИПУ РАН, 17-20 июня 2019); Конференция российских молодых исследователей в области электротехники и электроники (2021-2022 гг.), Автоматизированные системы управления и информационные технологии (г. Пермь, 2018-2021 г.г.); Энергетика. Инновационные направления в энергетике. CALS-технологии в энергетике (г. Пермь, 2018-2021 г.г.). Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-90001.

**Публикации.** Основные результаты изложены в 16 публикациях, из них 3 статьи в ведущих рецензируемых научных изданиях; 1 статья в журнале, индексируемом в МБЦ Web of Science, 3 публикации в трудах международной конференции, индексируемых в МБЦ Scopus, 2 свидетельства о государственной регистрации программ для ЭВМ.

**Структура и объем работы диссертации.** Диссертационная работа состоит из введения, пяти глав, заключения, списка использованной литературы. Работа содержит 109 страниц основного текста, 6 таблиц, 19 рисунков. Список использованной литературы включает 107 наименований.

## **Глава 1. Анализ методов контроля и диагностики вычислительных систем**

### **1.1 Анализ структуры вычислительных систем и их элементов**

Вычислительная система (ВС) – программно-аппаратный комплекс контроля и/или управления реальным объектом. Её основными задачами являются хранение информации, полученной от источников данных (ИД), представление этой информации пользователям системы для принятия решений, а также управление её элементами с целью настройки поведения объекта, которым она управляет. [14, 42, 51] Особенности таких систем являются: работа в режиме реального времени [73]; большая ответственность, и, следовательно, и высокие требования, предъявляемые к надежности и безопасности функционирования; необходимость в непрерывном функционировании. Стоит отметить, что ВС, при отсутствии оператора должна уметь обрабатывать нештатные ситуации и уведомлять о их происшествии необходимых лиц [48, 57].

Вычислительную систему можно разделить на 2 логические подсистемы: подсистема автоматизации и управления (АиУ) и подсистему сбора передачи и распределения информации (СПРИ). Основное назначение первой подсистемы – обеспечить возможность управления элементами системы с целью корректировки поведения управляемого объекта, предоставление информации о его прошлом, настоящем и возможном будущем. Назначение подсистемы СПРИ обеспечить целостную и своевременную доставку информации от одного элемента вычислительной системы до другого, пока она не будет записана в хранилище и представлена по требованию операторов или самой системы.

Составляющей частью любой системы являются её элементы. [41, 84] Любая система – набор элементов, выполняющих разные задачи, имеющие различную физическую реализацию и различный набор параметров.

Необходимо произвести классификацию элементов ВС, выделить схожие и различные параметры функционирования, определить их особенности.

### **1.1.1 Классификация элементов вычислительных систем**

Элементы, по выполняемым функциям в системе можно разделить на 3 группы:

- Управляющие;
- Связующие;
- Конечные.

Управляющие элементы ВС необходимы для формирования команд, корректирующих работу остальных элементов. Данные элементы отвечают за отображение текущего состояния элементов системы, уведомление о неисправностях. Хранение, отображение и анализ архивных данных, прогнозирование будущего состояния элементов системы являются необходимыми функциональными возможностями управляющих элементов. Важность таких элементов подразумевает повышенные требования к надёжности элементов (периодические отказы, недостаточность вычислительных ресурсов и т.д.) и к их безопасности. Получение несанкционированного доступа к управляющим элементам ставит под угрозу функционирование всей системы.

Связующие элементы – это элементы, отвечающие за передачу информации между всеми элементами системы. В настоящий момент связующими элементами являются коммутаторы, маршрутизаторы, различные вышки связи, соединяющие кабели. Сетевые платы, установленные внутри конечных и управляющих устройств также являются частью связующих элементов системы. Очевидно, что надёжность этих элементов не менее важна. При отсутствии связи или невозможности доставить информацию нужному получателю функционирование системы нарушается.

Конечные элементы – основные функционирующие элементы системы. Они являются точкой взаимодействия системы с внешним миром, получают информацию из внешней среды, преобразуют её и передают управляющим элементам. Получив управляющие последовательности команд от управляющих элементов, эти элементы адаптируют своё функциональное поведение (частота опроса среды, частота вещания, угол поворота и т.п.), тем самым корректируя своё взаимодействие с внешним миром.

Исходя из описанного функционала, основными элементами в системе являются всё же конечные устройства. Без них система не может функционировать. Так например, если вдруг пропадёт связь (отказ связующих элементов) или окажутся недоступны управляющие элементы (отказ управляющих элементов), то конечные элементы какое – то ограниченное время могут работать на текущих настройках и накапливать получаемые данные в свою память. Эти данные впоследствии могут быть получены, переданы и обработаны. Но в случае отказа конечных устройств, все остальные устройства хоть и будут функционировать, но с малой пользой. Управлять управляющим элементам становится нечем, передавать информацию остаётся только между управляющими устройствами, что не является прямым функционалом системы – только служебным. Можно сделать вывод о том, что повышение надежности функционирования в первую очередь требуется именно для конечных элементов.

Элементы вычислительной системы можно классифицировать по способу физической реализации:

- Монолитные;
- Распределённые;

По способам передачи информации элементы можно разделить на:

- Проводные;

- Беспроводные.

Основным достоинством проводных элементов, а также их основным недостатком является возможность прокладки помехоустойчивой физической среды передачи информации. Достоинства помехоустойчивой физической среды очевидны, а из недостатков можно отметить большие затраты на создание такой физической среды в местах дислокации конечных устройств вычислительной системы. Методы беспроводной передачи информации [22] не имеют подобных недостатков, но, с другой стороны, должна быть обеспечена целостность передаваемых данных (методами теории информации), а также защита от несанкционированного доступа, если управляющие или полезные данные системы не являются открытыми [66, 104].

### **1.1.2 Структура элементов вычислительных систем**

Рассмотрение структуры элементов ВС необходимо начать с определения видов структуры элементов. Существует два вида структуры:

- Логическая;
- Физическая.

Логическая и физическая структуры элемента, безусловно, зависят от выполняемых им функций. Логическую структуру проще всего описать с помощью терминологии модулей. Элементы ВС могут иметь в своём составе следующий перечень модулей:

- Модуль горизонтального взаимодействия;
- Модуль вертикального взаимодействия;
- Модуль хранения и обработки информации;
- Модуль диагностики.

Модуль горизонтального взаимодействия отвечает за взаимодействие с другими элементами ВС, находящимся на одном уровне с данным. На определённых уровнях системы горизонтальное взаимодействие между элементами может как отсутствовать, так и присутствовать. Данное взаимодействие является неотъемлемой частью механизма резервирования в системе. При наличии такого взаимодействия элементы могут принимать на себя роль другого элемента того же уровня, тем самым сохраняя возможность функционирования системы, пока другой элемент неисправен. Горизонтальное взаимодействие необходимо, когда в перечень задач элемента входит производство синхронизации полученных/имеющихся данных. Данный функционал необходим в особенности управляющим элементам, так как они могут воздействовать на одни и те же элементы. Так, если необходимо сконфигурировать элемент на основе имеющихся данных, то несколько устройств управления должны обладать одинаковой информацией об управляемом элементе для его корректной настройки. Примером такой ситуации может служить управление положением уличной камеры дорожного движения. Управление может осуществляться операторами с различных устройств (несколько удалённых устройств управления или несколько локальных), основываясь на данных о положении камеры, получаемых с неё или видео/фото потока. Так, имея различные данные на двух устройствах управления, смена конфигурации камеры может привести к последствиям, которые не предполагались ни первым управляющим устройством, ни вторым.

Следующим модулем является модуль вертикального взаимодействия. Этот модуль особенно важен, так как отвечает за непосредственное взаимодействие устройств 1-го уровня с внешней системой (получение / отправка данных), а также за управляющее взаимодействие, доступное для этого элемента. Вертикальное взаимодействие необходимо для функционирования системы, т.к. именно оно позволяет доводить данные до

нужного уровня, на котором будет произведена их обработка, отображение, анализ. После анализа, с помощью модулей вертикального взаимодействия отдельных устройств будет отправлена управляющая последовательность, которая должна дойти до конечных устройств для проведения корректировки их взаимодействия с внешней системой или друг другом. При возникновении неисправности в модулях вертикального взаимодействия, возможность корректного функционирования системы ставится под угрозу. Исходя из этого необходимо, чтобы для вертикального взаимодействия была выбрана физическая реализация (устройства, протоколы, механизмы резервирования), устойчивая к возникновению ошибок / неисправностей. Для примера вертикального взаимодействия можно рассмотреть работу датчиков возникновения пожара на нефтедобывающем предприятии. Датчики постоянно находятся во взаимодействии с внешней системой (воздухом) и при возникновении признаков пожара должны отправить данные об этом через модуль вертикального взаимодействия. При возникновении неисправностей в данном модуле – использование системы становится бесполезным, а последствия катастрофическими. Управляющие устройства с помощью данного модуля могут посылать корректирующие сигналы для управления положением датчика, чувствительностью и т.п. Так, например, если при ложном срабатывании датчика тратится очень много ресурсов, то корректировка чувствительности очень важна и будет произведена через модуль вертикального взаимодействия.

Модуль хранения и обработки информации также может присутствовать в логической структуре элемента ВС. Его функциями является обеспечение возможности хранения информации (полученной из внешней системы или другого элемента) для дальнейшей передачи через вертикальное взаимодействие, а также производство частичной обработки этой информации (фильтрация, агрегация). Как уже отмечалось выше, данный модуль может

обеспечить возможность сохранения информации, которая ещё не была передана в связи с различными проблемами, возникающими в физической реализации модуля горизонтального взаимодействия. Эта информация может быть передана немного позже, что обеспечит целостность данных и позволит более грамотно системе выполнять ее непосредственные задачи. Есть системы, в которых хранение получаемой информации не имеет смысла. Так, например, целью любой сигнализации является не хранение информации о совершённом взломе, а наиболее быстрое уведомление заинтересованных лиц. Хранение этой информации может быть полезно не только для исполнения функций системы, но и для последующего анализа неисправности, если место её возникновения неизвестно. Если сигнализация сработала, но из-за проблем в модуле вертикального взаимодействия не был отправлен рабочий сигнал, то факт срабатывания может подтвердить хранение данных в самом датчике. Это укажет на то, что проблема была именно в модуле вертикального взаимодействия. Если же данного модуля (хранения и обработки) нет, то локализовать неисправность становится сложнее. Будет не очевидно, неисправен ли датчик или модуль вертикального взаимодействия.

Модуль диагностики необходим элементам, функциональные возможности которых важны, а сбои должны быть оперативно обнаружены и устранены. Данный модуль может выполнять различные функции в зависимости от типа системы диагностирования, которая используется в ВС. Так, например, при использовании централизованной системы диагностирования, функционал модуля ограничивается проведением функциональной и тестовой диагностики по полученным от контроллера управляющим сигналам. В случае использования встроенной системы диагностирования [80], функционал модуля заключается ещё и в формировании и отправке управляющей последовательности для другого элемента, сборе

результатов тестирования. В таком случае каждый элемент может являться контроллером диагностики для другого элемента.

Под параметрами элементов ВС в данном случае подразумевается некоторый набор значений функциональных и аппаратных характеристик устройства, выполняющего роль элемента в системе – *модель элемента вычислительной системы*. При использовании параметрического описания элемент является черным ящиком, у которого в реальном времени изменяются характеристики. Эти характеристики могут изменяться при простое элемента (отсутствие функциональных задач в системе – режим ожидания), при получении управляющих последовательностей от контроллера или взаимодействия с внешней системой (средой). При параметрическом описании элементов ВС становится не так важна аппаратная реализация элемента - если 2 элемента описываются одной и той же параметрической моделью, то при выполнении одних и тех же функциональных задач эти элементы для системы являются идентичными.

Как было сказано ранее, параметры элементов ВС можно разделить на 2 группы:

- Функциональные;
- Аппаратные.

Под функциональными параметрами понимаются параметры, которые были назначены элементу и существуют только внутри системы. Такие параметры могут играть роль при назначении задач элементу, при интерпретации результатов работы. Если в системе реализовано иерархическое разделение элементов, где один элемент ценится выше другого по определённым признакам (взаимодействует с наиболее критичными «участками» внешней системы, используется более точная аппаратура измерения), то и данные приходящие от этого элемента должны быть

обработаны отлично от других элементов, например, с большей степенью доверия.

Аппаратные параметры элемента ВС – характеристики физической реализации элемента. Их роль в системе очень важна, т.к. на их основе могут определяться функциональные параметры, а также текущее состояние элемента. Данные параметры могут быть как постоянными (число ядер процессора, чувствительность измерительного датчика, тип накопителя памяти), так и изменяющимися в режиме реального времени (загруженность процессора, заряд батареи, загруженность оперативной памяти). Для решения задач мониторинга и диагностики наиболее важными являются динамические аппаратные параметры, так как по ним можно определить работоспособность элемента ВС.

Перечень параметров для каждого элемента может быть собственный, но решать задачи мониторинга и диагностики становится проще, когда в системе существуют параметрические модели её элементов. Конечно, не всегда можно описать все элементы одной моделью, но построить единую модель для элементов, находящихся в системе на одном уровне вполне реальная задача.

### **1.1.3 Особенности элементов вычислительных систем**

В разных ВС элементы подсистем могут иметь как сходства, так и различия.

- Диагностическая модель может описать элементы системы, но для каждой конкретной области автоматизации, задачи которой решает ВС, должна быть учтена специфика данной области, что влияет на состав используемых устройств в различных системах;
- В зависимости от степени распределённости элементов системы имеются различия в выборе связующих элементов. Передача информации на более дальние расстояния требует большей коррекции ошибок и передающих

устройств, а также обеспечение информационного обмена между устройствами одного уровня требует дополнительную настройку элементов подсистемы.

Сходства элементов различных вычислительных систем:

- Несмотря на различные направления использования ВС, элементы подсистем имеют общие черты (функциональное назначение и техническую реализацию), что позволяет построить общую диагностическую модель и модель дефектов;
- Наличие двух, описанных ранее, подсистем, является достаточным и необходимым условием работы ВС, что позволяет представить любую ВС в их виде совокупности.

В зависимости от функциональных задач, решаемых вычислительной системой, элементы системы могут иметь некоторые особенности. Если элементы системы вынуждены постоянно менять своё географическое местоположение, то обеспечить постоянный подвод электричества к ним не представляется возможным. В таком случае элементы системы работают на аккумуляторных батареях, благодаря чему обеспечивается их энергетическая *автономность* [1, 39]. При этом должны быть приняты максимальные меры для обеспечения эффективного (экономного) использования ресурсов элемента для того, чтобы не вызывать дополнительную разрядку аккумулятора. [61, 67] Примером такой системы может являться система передвижных мобильных фотофиксаторов. Такая система необходима для контроля соблюдения правил дорожного движения на разных участках дорожной сети. Сам фотофиксатор работает от батареи и может менять своё географическое местоположение через различные промежутки времени.

Энергетическая автономность может накладывать ограничения на возможность использования различных методов передачи информации. Для передачи информации агрегатору в системах с автономными элементами могут

использоваться *беспроводные технологии* передачи информации. В системе, описанной выше – это например LTE + GSM.

В основном, элементы ВС построены на основе ПЛИС. Соответственно им присущи все особенности, которые присущи различным типам ПЛИС. К ним можно отнести: наличие/отсутствие энергонезависимой памяти, наличие/отсутствие встроенной системы безопасности, возможность реконфигурации (FPGA).

Наличие энергонезависимой памяти может обеспечить целостность конфигурации для автономных элементов. Например, при разряде аккумулятора в элементах, у которых энергонезависимая память отсутствует необходимо заново производить их конфигурацию, что усложняет их эксплуатацию, а также является уязвимостью и возможностью для обратного инжиниринга.

Безопасность в настоящее время очень важна, так как множество устройств имеют доступ в сеть, что повышает вероятность вредоносного воздействия. Даже когда присутствует энергонезависимая память, то она защищает только конфигурационные данные, но никак не прикладные данные, получаемые или передаваемые во внешнюю систему. Поэтому элементы, которые имеют доступ в сеть, содержат:

- Аппаратную защиту от дифференциального анализа потребляемой мощности (differential power analysis, DPA);
- Недетерминированный генератор случайных последовательностей (Non-deterministic random bit (number) generator, NRBG);
- Аппаратные межсетевые экраны для защиты интегрированного процессора ARM Cortex-M3.

Особенности элементов предполагают различные механизмы повышения их надёжности. Одним из наиболее эффективных методов повышения

надежности элементов ВС является использование современной системы мониторинга и диагностики [34], учитывающей особенности аппаратной реализации элементов системы.

## **1.2 Способы обеспечения надежности вычислительных систем**

### **1.2.1 ГОСТ 27.002-2015 Надежность в технике (ССНТ). Термины и определения**

В любой вычислительной системе - промышленной, государственной или частной рано или поздно встает вопрос об обеспечении надежности [52]. Лучше всего, когда такой вопрос задается на этапе проектирования системы [72], т.к. различные варианты обеспечения надежности, рассматриваемые в момент, когда система ещё не реализована, позволят скорректировать аппаратную или функциональную реализацию механизмов работы системы. Но также возможны варианты обеспечения (повышения) надежности вычислительной системы и её элементов на этапе реализации и эксплуатации системы.

В России стандартизацию методов, показателей и характеристик надежности задает «ГОСТ 27.002-2015 Надежность в технике (ССНТ). Термины и определения» [10]. Этот стандарт устанавливает основные понятия, термины и определения понятий в области надежности. Здесь сама надежность определяется как «Свойство объекта сохранять во времени способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, хранения и транспортирования». Данный стандарт включает в себя и определения свойств надежности а именно: «Надежность является комплексным свойством, которое в зависимости от назначения объекта и условий его применения может включать в себя безотказность, ремонтпригодность, восстанавливаемость, долговечность, сохраняемость, готовность или определенные сочетания этих свойств.»

### 1.2.2 Способы обеспечения составляющих надежности вычислительных систем

Для того, чтобы определить способы обеспечения надежности, необходимо сначала определить, как оценивается надёжность той или иной вычислительной системы. В п 3.6.6 вышеописанного ГОСТа определяется перечень комплексных показателей надежности, а именно:

- **Коэффициент готовности** - вероятность того, что объект окажется в работоспособном состоянии в данный момент времени;
- **Коэффициент неготовности** - вероятность того, что объект окажется в неработоспособном состоянии в данный момент времени;
- **Коэффициент оперативной готовности** - вероятность того, что объект окажется в работоспособном состоянии в данный момент времени и, начиная с этого момента, будет работать безотказно в течение заданного интервала времени;
- **Коэффициент технического использования** - отношение математического ожидания суммарного времени пребывания объекта в работоспособном состоянии за некоторый период эксплуатации к математическому ожиданию суммарного времени пребывания объекта в работоспособном состоянии и простоев, обусловленных техническим обслуживанием и ремонтом за тот же период;
- **Коэффициент сохранения эффективности** - отношение значения показателя эффективности использования объекта по назначению за определенную продолжительность эксплуатации к номинальному значению этого показателя, вычисленному при условии, что отказы объекта в течение того же периода не возникают.

Таким образом, оценка надежности заключается в определении численных значений характеристик надежности рассматриваемого объекта.

На различных этапах использования ВС методы обеспечения её надёжности различаются. На этапе проектирования системы, надёжность обеспечивается путем рационального проектирования системы, а также расчётно – экспериментальными методами обеспечения надёжности. К таким методам можно отнести конструктивные методы повышения надёжности и резервирование.

Резервирование – один из наиболее действенных методов обеспечения надёжности на этапе проектирования системы. Данный метод включает в себя создание избыточности, путем введения в систему дополнительных элементов, которые в случае отказа основных узлов могут принять на себя их роль на время восстановительных работ. Данные элементы не требуются системе для работы в штатном режиме. Резервирование основывается на удалении области состояния изделия от предельно-допустимых значений параметров. Таким образом возникает запас надёжности. Существуют различные способы обеспечения резервирования. Так можно выделить:

- Активное резервирование;
- Пассивное резервирование.

Данные способы резервирования различаются реакцией на появление отказа, соответственно до и после появления отказа. Виды схем резервирования можно разделить на:

- Общее;
- Раздельное;
- Единичное;
- Внутриэлементное;
- Скользящее;
- С избирательной схемой;
- Автономное.

Также избыточность, вводимая в системе может разделяться на:

- С целой кратностью;
- С дробной кратностью.

Резервирование способно значительно повысить надежность вычислительной системы, если система грамотно под это спроектирована. В свою очередь, обеспечение резервирования может быть очень дорогостоящим и сложным процессом. Если ресурсов для обеспечения требуемого уровня резервирования не хватает (финансовых, технических), то необходимо рассматривать другие варианты повышения надежности системы. Обеспечение надежности на этапе проектирования – достаточно сложный процесс, так как для правильного проектирования системы необходимо с достаточной точностью знать условия, в которых эта система будет эксплуатироваться. Такими условиями могут быть степень распределённости элементов системы, число пользователей, число элементов, физическая среда передачи данных и т.п. Не во всех ситуациях можно с достаточной степенью достоверности определить все условия эксплуатации системы.

Таким образом, в различных ситуациях связанных с невозможностью или недостаточностью обеспечения требуемого уровня надежности вычислительной системы на этапе проектирования и реализации одним из наиболее предпочтительных вариантов является повышения надежности на этапе эксплуатации вычислительной системы.

На этапе эксплуатации все действия по повышению надёжности можно разделить на несколько групп:

- Разработка научных методов эксплуатации;
- Сбор, анализ и обобщение опыта эксплуатации;
- Повышение квалификации обслуживающего персонала.

Важным средством повышения надежности, относящимся к п. 2 данного списка является использование диагностики. Системы функционального диагностирования дают возможность контролировать объект в процессе выполнения им рабочих функций и реагировать на отказ в момент его возникновения. Эти системы могут проектироваться и изготавливаться вместе с контролируемым объектом, а могут быть реализованы и добавлены в систему по мере необходимости в процессе эксплуатации.

### **1.3 Диагностика и контроль функционирования вычислительных систем как основной способ обеспечения надежности на этапе эксплуатации**

#### **1.3.1 Классификация систем диагностирования**

Системы технического диагностирования [9] (СиТД) могут быть различны по своему составу, назначению, структуре, конструкции, схемотехническим решениям, месту установки. [45, 47, 88] Они, как правило, классифицируются по ряду признаков, определяющих их назначение, задачи, структуру, состав технических средств: по степени охвата ОД; по принципу взаимодействия между ОД и СТД; по используемым средствам технического диагностирования и контроля; по степени автоматизации ОД [78].

**По степени охвата** СиТД разделяются на локальные и общие. Под локальными понимают СиТД, решающие одну или несколько задач диагностирования – определения ТС, дефекта или прогнозирования. Общими называют СиТД, решающие все задачи диагностирования.

**По принципу взаимодействия** ОД и СТД СиТД подразделяют на системы с функциональным диагнозом и системы с тестовым диагнозом. В первых системах решение задач диагностики осуществляется в процессе функционирования ОД, во вторых - решение задач диагностики

осуществляется в специальном режиме работы ОД путем подачи на него тестовых сигналов.

**По используемым СТД** СиТД можно разделить: на системы с универсальными СТД (например универсальные информационно-измерительные комплексы в комбинации со специальным ПО); системы со специализированными средствами (стенды, имитаторы, специализированные СТД); системы с внешними средствами, в которых СТД и ОД конструктивно отделены друг от друга; системы со встроенными средствами, в которых ОД и СТД конструктивно представляют одно изделие.

**По степени автоматизации** СТД делятся на: автоматические, в которых процесс получения информации о техническом состоянии ОД осуществляется без участия человека; автоматизированные, в которых получение и обработка информации осуществляется с частичным участием человека; неавтоматизированные (ручные), в которых получение и обработка информации осуществляется человеком-оператором.

Аналогичным образом могут классифицироваться также СТД: автоматические, автоматизированные, ручные.

СиТД применительно к ОД должны предупреждать постепенные отказы; выявлять неявные отказы; осуществлять поиск неисправных узлов, блоков, сборочных единиц и локализовать место отказа.

В системе управления качеством диагностико-прогнозирующие процессы подразделяют на три основные временные составляющие:

1. **прогноз** – процесс определения ТС ОД в будущем на конечном временном интервале с заданной достоверностью;
2. **диагноз** – процесс определения технического состояния объекта в данный момент;

3. **генезис** – процесс определения ТС ОД с определенной точностью на заданном в прошлом временном интервале.

Согласно приведенной выше классификации, техническая диагностика может быть соответственно представлена: как непосредственно техническая диагностика; техническая прогностика; техническая генетика (прогнозирование назад или ретропрогноз).

Такое деление имеет место тогда, когда процесс прогнозирования осуществляется как бы раздельно. Но на практике все три процесса представляют собой неразрывное единство, ибо они выражаются в динамике изменения состояния функционирующей системы или объекта.

### 1.3.2 Условия применения систем диагностирования

Для анализа условий применения различных систем технического диагностирования составляется таблица 1:

Таблица 1

#### Функции систем технического диагностирования

Область применения	Системы диагностирования		
	Встроенные	Мобильные	Стационарные
<b>Изготовление</b>			
Выявление дефектов конструкции	+	+	(+)
Контроль функционирования	(+)	-	-
Защита от аварий	(+)	-	-
<b>Эксплуатация</b>			
Адаптация к изменению	(+)	-	-

технологии			
Адаптация к изменению внешней среды	+	-	-
Обнаружение неисправных элементов	+	+	-
Контроль параметров	+	+	-
Регулировка по динамическим параметрам	+	(+)	-
Накопление данных о параметрах и видах отказа	(+)	+	-
Прогнозирование	+	(+)	-
Потребление			
Контроль качества	+	+	-

Примечание. Знак «+» соответствует возможности применения системы; знак «(+）」 - предпочтительному применению; знак «-» - отсутствию применения системы диагностирования.

В зависимости от назначения сложных комплексных технических устройств диагностированию подвергают как основное изделие, включая конструкцию и её составляющие (встроенные системы), так и состояние функционально объединённых комплектующих (внешние системы).

Для количественной и качественной оценки свойств систем технического диагностирования применяют следующие характеристики и показатели качества:

- оперативность - характеризует возможность своевременного и обоснованного выбора управляющих воздействий в процессе функционирования системы с целью учёта изменений в ситуации;

- гибкость - определяет возможность системы перепрограммирования на различные условия и режимы работы;
- мобильность - определяет быстроту перестройки системы с изменением состояния внешней среды;
- живучесть - характеризует возможность временного продолжения функционирования в случае повреждения отдельных деталей и узлов.

К диагностическим системам приемлемы общие принципы системного анализа:

- принцип целеобусловленности создания системы (совокупности технических средств и обслуживающего персонала);
- принцип относительности (совокупность элементов системы, рассматриваемая как часть большей системы);
- принцип управляемости (определения возможности изменения структуры системы и иерархичности её построения);
- принцип модулируемости (обеспечение возможности прогнозирования состояния объекта, диагностирования или развития самой системы).

При разработке систем диагностирования должны решаться задачи изучения объекта, его возможных дефектов и признаков проявления, выбора или построения модели поведения исправного объекта и его неисправных модификаций. [7, 96]

Изучение объектов предусматривает их классификацию по различным признакам, например, по характеру изменения значений параметров и по условиям работы. Анализ характера работы объектов осуществляется построением диагностической модели [2, 70, 102].

Исходя из требований по надежности к конкретной вычислительной системе, а также из множества условий реализации и эксплуатации вычислительной системы, необходимо выбрать подходящую систему технического диагностирования [50, 94, 103]. Исходя из данных, представленных в таблице 1, встроенные системы диагностирования позволяют покрыть большинство областей применения систем диагностирования, поэтому их использование предпочтительнее. Стоит также отметить, что для использования таких систем должны быть выполнены определенные требования на различных этапах жизненного цикла системы, о которых будет сказано в следующих пунктах данной главы диссертации.

### **1.3.3 Подходы к проведению диагностирования вычислительных систем**

Для диагностирования элементов вычислительных систем первой и одной из важнейших задач является математическое описание системы и её элементов – диагностическая модель.

Необходимо отметить, что некоторая часть важных диагностических характеристик скрыта за специализацией конкретной системы. Данные характеристики не могут быть выделены в общую диагностическую модель вычислительной системы, так как сильно завязаны на назначение системы. Тем не менее, достаточно большое число характеристик можно и нужно вынести в общую модель, т.к. они не менее важны, чем специфические и зачастую дают достаточный объём диагностической информации для принятия решений о функционировании элементов системы. Как было сказано ранее, элементы, выделенные на программно-аппаратном уровне, являются объектами мониторинга и диагностики [60, 90, 91]. В частности, их роль выполняют:

1. Первичные преобразователи информации;
2. Устройства локального управления (ПЛК);

3. Коммуникационные устройства как неотъемлемая часть любой ВС [26].

Методы диагностики данных элементов обычно сводятся к следующему подходу – у каждого элемента есть определённый набор параметров, к которым можно получить доступ и таким образом узнать о состоянии элемента на определённый момент времени [30, 33, 49, 54, 101]. К таким параметрам можно отнести: уровень заряда батареи, режим работы (при возможности мультирежимной работы устройства), ошибки транспортной шины контроллера и т.п.

Для первичных преобразователей информации набор чуть более скромнен:

1. Есть связь/нет связи;
2. Статус элемента (исправен, авария, включен, отключен, ожидание и т.п.);
3. Текущая конфигурация элемента.

Эти параметры говорят о состоянии контролируемого элемента и могут сигнализировать о неисправности. Но такой подход - мониторинг параметров и состояний- имеет несколько минусов:

1. Параметры показывают факт неисправности (при установленном агенте), но не дают информацию о причине её возникновения.
2. Подход работает для каждого элемента по отдельности, невозможно комплексно контролировать системы таких элементов (УЛУ ↔ ПиПР).

Следующий метод диагностики – метод тестирования. Суть метода заключается в отправке контрольного сигнала (последовательности сигналов) на контролируемое устройство [65]. Его реакция – ответ на контрольный сигнал

должен быть адекватным, совпадающим с ответом, заложенным на этапе проектирования системы для данного контрольного сигнала.

Данный метод имеет несколько недостатков, а именно:

1. Необходимо наличие большого числа тестовых (контролирующих) сигналов для покрытия функционала устройства.
2. При изменении (расширении) функционала элемента, должны быть обработаны новые ситуации и придуманы новые контрольные сигналы для симуляции поведения элемента в различных состояниях.
3. При возникновении нескольких неисправностей, результат теста может оказаться ложноположительным, что будет давать неверную информацию о наличии неисправностей в элементе.
4. Отсутствие возможности поиска «плавающих» неисправностей, т.к. тест происходит в определённое время в определённых условиях.

Для элементов ВС можно выделить рекомендации по повышению контролепригодности на этапе проектирования:

1. Создать максимально полную модель элемента, основываясь на его функциональных возможностях, реальных (реализованных на данный момент) и теоретических (планируемых).
2. Определить максимально возможный перечень параметров, доступных для получения и передачи агентом.
3. Параметры необходимо сделать максимально простыми и максимально независимыми друг от друга для реализации независимости состояний. Изменение одного параметра минимально влияет на остальные, давая, таким образом, дискретную оценку каждого состояния элемента, что повышает ценность данного параметра и достоверность его значения.

4. Сделать максимально гибкую конфигурацию элемента, таким образом, давая возможность не выключать его из рабочего состояния. То есть вынести наибольшее возможное число важных изменяемых параметров в подсистему конфигурации, чтобы без отключения от системы менять паттерны работы элемента.

5. Создать API для взаимодействия с элементом сторонних инструментов диагностики, а также реализовать возможность работы с элементом по стандартным, общепринятым протоколам для данного типа устройств.

Инструменты контроля и диагностики очень важны, ибо их задачей является точное обнаружение причины неисправности и, возможно, выдача рекомендаций по исправлению обнаруженной неисправности [17, 38].

В зависимости от типа диагностируемого элемента, инструменты можно разделить на несколько категорий:

1. Программные: (Сниффер траффика)
2. Программно-аппаратные (POST плата, измерительные приборы)

У каждого типа инструментов есть как свои достоинства и недостатки. Для более качественной диагностики необходимо иметь возможность комбинировать несколько типов инструментов, но иметь при этом *один* инструмент, способный покрыть наибольший перечень неисправностей [8, 40, 71]. Это даёт несколько преимуществ:

1. Уменьшает время диагностирования, т.к. использование нескольких инструментов требует затрат времени.

2. Возможная противоречивость показаний нескольких разных инструментов, что может ввести в ступор и также увеличить время поиска и устранения неисправностей.

3. Отпадает необходимость поиска универсальных специалистов, способных проводить диагностику с помощью нескольких типов инструментов, что снижает затраты на заработную плату и поиск кадров.

#### **1.4 Математическая постановка задачи диссертационного исследования**

Для формализации задачи повышения надежности вычислительной системы необходимо рассмотреть комплексные показатели надежности вычислительных систем. К ним относятся: коэффициент готовности, коэффициент оперативной готовности, коэффициент технического использования, коэффициент сохранения эффективности). В данной работе рассматриваться будет такой показатель как коэффициент технического использования (1).

$$K_{\text{ТИ}} = \frac{T_p}{T_p + T_{\text{РЕМ}} + T_{\text{ТО}}}, \quad (1)$$

где  $T_p$  – время нахождения объекта в работоспособном состоянии за наблюдаемый интервал,  $T_{\text{РЕМ}}$  – время нахождения объекта в плановом и неплановом ремонте,  $T_{\text{ТО}}$  - время нахождения объекта в плановом и неплановом техническом обслуживании.

В рассматриваемых вычислительных системах, время нахождения системы в работоспособном состоянии напрямую зависит от числа её работоспособных элементов. Для системы устанавливается некоторое пороговое значение  $N_{\text{ПОР}}$  – пороговое значение числа работоспособных элементов, когда система ещё считается работоспособной. Таким образом, величина  $T_p$  в формуле (1) зависит от характера изменения числа работоспособных элементов системы.

$$T_p = f(N_{\text{РАБ}}(t)) = f(\{EF\}, \{K\}, \{\lambda\}, N_{\text{ЭЛ}}, P), \quad (2)$$

где  $N_{РАБ}(t)$  – функция изменения числа работоспособных элементов в вычислительной системе,  $EF$  – показатель надежности элемента вычислительной системы,  $K$  – значение критерия работоспособности элемента вычислительной системы,  $\lambda$  – значение важности критерия работоспособности,  $N_{ЭЛ}$  – число элементов в системе,  $P$  – число необходимых проверок (тестов).

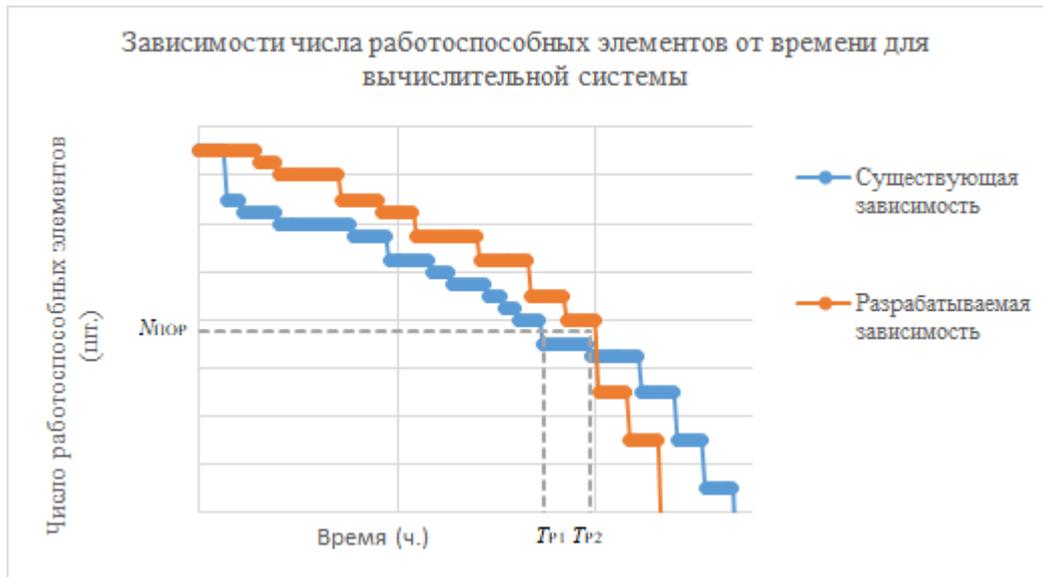


Рисунок 1. Примеры зависимостей числа работоспособных элементов вычислительной системы от времени,  $N_{ПОР}$  – пороговое значение числа работоспособных элементов системы, при котором вся система считается работоспособной,  $T_{P1}$  – время нахождения системы в работоспособном состоянии по существующей зависимости,  $T_{P2}$  – время нахождения системы в работоспособном состоянии по разрабатываемой зависимости.

На рис. 1 показано, как различная характеристика изменения числа работоспособных элементов оказывает влияние на величину  $T_P$ . Таким образом, изменяя зависимость числа работоспособных элементов, можно увеличить время между отказами, что, в свою очередь, позволит увеличить такую характеристику надежности системы, как коэффициент технического использования.

Для решения поставленной задачи необходимо разработать методику эффективного перераспределения диагностической нагрузки между элементами

встроенной системы диагностирования (в такой системе элементы вычислительной системы выполняют роль агентов диагностирования) в зависимости от технического состояния самих элементов. Методика включает в себя модель встроенной системы тестового диагностирования, модель элемента системы, а также алгоритм перераспределения диагностической нагрузки и методы расчёта основных показателей разработанных моделей.

### **1.5 Выводы по главе**

В первой главе были рассмотрены существующие методы контроля и диагностики вычислительных систем. Для этого был проведен анализ структуры вычислительных систем и их элементов, выделены особенности их функционирования. Данный анализ показал, что эффективность применения вычислительных систем во многом обуславливает обеспечение заданного качества функционирования тех процессов, информационно-технической платформой которых они являются. При этом элементы вычислительных систем должны сами отвечать повышенным требованиям к своим качественным и эксплуатационным характеристикам. Для этого необходимо разрабатывать и применять многофункциональные и высокопроизводительные системы диагностирования. Как следствие возникает *противоречие* – усложнение объектов диагностирования и самой системы диагностики приводит к увеличению объёмов передачи и сложности обработки диагностической (служебной) информации, что снижает эффективность взаимодействия элементов вычислительной системы между собой, приводит к их перегрузке, снижает время автономной работы, ухудшает другие показатели. Поэтому актуальной задачей становится разработка и реализация методов диагностирования вычислительных систем и их элементов, не снижающих, а улучшающих их качественные и эксплуатационные характеристики. Такие методы позволяют повысить некоторые показатели надежности вычислительных систем как коэффициент готовности (увеличение наработки

между отказами) и коэффициент технического использования (увеличения времени работоспособности системы и уменьшения времени ремонта). При этом стоит отметить, что значения этих величин можно повысить путем прогнозирования состояния вычислительной систем и её элементов, тем самым не допустив возникновения некоторых отказов.

## **Глава 2. Разработка моделей встроенной системы диагностирования**

### **2.1 Анализ существующих моделей систем тестового диагностирования и их элементов**

Вопросами встроенных систем диагностирования с возможностью самодиагностирования в разное время занимались сотрудники ИПУ РАН им. В.А. Трапезникова: Ведешенков В. А. [4, 5, 6], Аминев Д. А. [15], сотрудник Московского института электроники и математики НИУ ВШЭ - Димитриев Ю. К. [13], Сагунов В.И., Ломакина Л.С. [38], Гуров С.В. [11, 36] и другие. Также, подобными вопросами занимались и наши зарубежные коллеги, такие как S. L. Nakimi (Университет Калифорнии) [54, 82], С.-L. Yang и G.M. Masson [106] (Университет Джонса Хопкинса), Woohyun Kim, James E. Braun [77] (Herrick Laboratory, Purdue University, Mechanical Engineering, West Lafayette, IN, United States), Xudong Li [84] (National Space Science Center, Beijing, China), Barry Dowdeswell, Roopak Sinha, Stephen G.MacDonell [58] (Auckland University of Technology, Auckland, New Zealand).

В частности, в [4] предложен метод (модель Барси-Грандони-Маэстрини) самодиагностирования модулей и линий связи цифровых систем с реконфигурацией, допускающий устойчивые кратные отказы ограниченного числа компонентов. В основу метода [4] положен принцип расширяющихся областей.

В [15] рассмотрены существующие методы самодиагностирования и описана возможность их применения в рамках диагностирования распределенной радиотехнической системы наблюдения. Предложен подход к реконфигурации системы в зависимости от результатов диагностирования.

В [77] рассматривается системное диагностирование по модели Препарата – Метца - Чена в присутствии множественных неисправностей, анализируются условия, при которых состояние каждого модуля определяется

только по исходам тестирования физически связанных с ним модулей (условия самоопределимости). Введен класс локально ( $tr/t$ ) – диагностируемых систем, где  $t$  – кратность неисправностей,  $tr$  – для которых обеспечивается определение состояния всех модулей системы.

В работе [52] рассмотрена адаптивная встроенная диагностическая система. В её основе лежит минимизация числа тестов для определения безошибочно работающих элементов системы, с целью увеличения числа диагностирующих агентов. (Метод расширяющихся областей).

В [106] рассмотрена гибридная диагностическая система. Её особенностью является учёт возможных коммуникационных и коммутационных проблем в рамках обмена информацией между системой диагностирования и элементами системы.

В [77] рассматривается подход к проектированию FDD (fault detection and diagnostics) систем, которые обладают потенциалом для повышения энергоэффективности наряду с сокращением затрат на обслуживание. Для достижения этой цели реализуются виртуальные датчики и модели воздействия сбоя, для которых требуются только недорогие измерения, для обнаружения причины сбоя, диагностики серьезности сбоя и оценки снижения производительности.

В работе [84] рассмотрена адаптивная встроенная диагностическая система. В ней активно используется трансферное обучение, за счет разработки интегрированного подхода к диагностике неисправностей с различными типами компонентов. Используются два метода глубокого обучения: сверточная нейронная сеть (CNN) и многослойный персептрон (MLP), для обучения нескольких базовых моделей с набором исходных данных.

В [58] описаны методы выявления и диагностики неисправностей, применяемые в области управления в аэрокосмической, автомобильной и

промышленной сферах. В каждой сфере используют определенные методы для удовлетворения различных диагностических потребностей. Одним из важнейших пробелов является проблема эффективности расходования ресурсов в беспроводных вычислительных системах [23].

Можно отметить, что большинство работ по данной теме рассматривают ошибки диагностирования и проблемы в канале передаче данных как причину для реконфигурации диагностической системы. Это, несомненно, очень важно, но также важно учитывать текущую загруженность и производительность системы и её элементов, ведь процессы диагностирования дополнительно загружают ресурсы элементов системы, что ведёт к увеличению вероятности их износа, отказа и снижению производительности.

## **2.2 Разработка модели встроенной системы тестового диагностирования**

Одним из основных инструментов повышения надежности и отказоустойчивости элементов ВС является использование развитой, отвечающей современным требованиям, системы диагностирования [11, 12, 29, 43]. Системы диагностирования по виду разделяются на внешние и встроенные. Каждый тип используется в соответствии с требованиями надежности, предъявляемыми к системе. К основным преимуществам встроенных систем диагностирования относятся:

1. Отсутствие необходимости использования, как правило, дорогостоящей и сложной в развертывании и администрировании внешней системы диагностирования.
2. Более полная и адекватно характеризующая конкретный элемент ВС управляющая и диагностическая информация.

3. Возможность выявления дефектов конструкции, контроль функционирования, защита от аварий

4. Возможность внедрения на практически любом этапе жизненного цикла вычислительной системы.

В данной работе будет разрабатываться комбинированная система диагностирования. В ней за распределение диагностических функций и функциональный контроль отвечает внешнее устройство (контроллер), а тестовое диагностирование выполняют назначенные для этого встроенные средства в структуре самих элементов ВС [16, 44, 99].

Применение встроенных средств диагностирования занимает часть производительности управляющих модулей (процессоров, контроллеров) элементов ВС. Ресурсы производительности [28], назначенные для выполнения диагностирующих функций элементов (тестеров) расходуются на формирование теста, передачу – прием данных и их последующую обработку. Для того, чтобы избежать быстрого разряда батарей и нарушения функционирования необходимо эффективно распределять диагностические задачи.

На рис. 2 представлена структурная схема разрабатываемой системы диагностирования (СД).

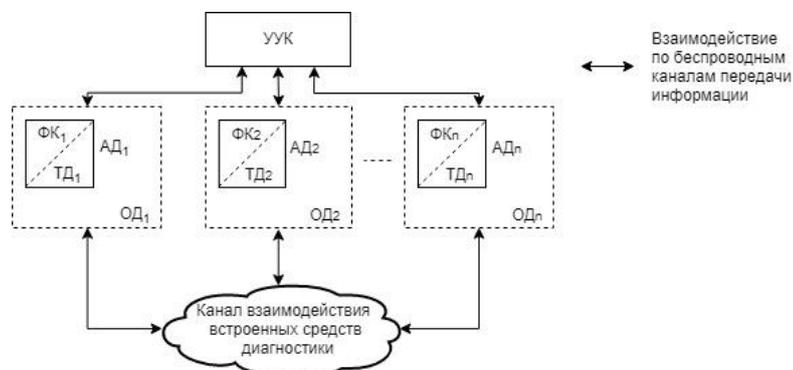


Рисунок 2. Структурная схема разрабатываемой системы встроенного диагностирования

Данная система содержит в себе функциональные элементы (устройства, выполняющие непосредственно функции системы), надёжность функционирования которых необходимо повысить, устройство управления и конфигурирования (УУК) требуется для распределения функциональных задач, сбора результатов выполнения и конфигурирования элементов системы. Функциональные элементы между с собой и УУК общаются по каналам передачи информации. Далее опишем встроенную систему тестового диагностирования [69, 87, 95].

В этой системе предполагается, что каждый функциональный элемент ВС, в зависимости от конфигурации, может являться объектом тестирования либо тестером [53, 83, 99]. Конфигурация каждого элемента задается устройством управления и конфигурирования (УУК), которое также выполняет функциональный контроль (мониторинг). В объекте диагностирования (ОД) реализован агент (А) - программно-аппаратный модуль, который выполняет следующие функции:

1. Тестер - формирование тестов, передача, прием и обработка диагностической информации.
2. Объект тестового диагностирования (ТД) – приём теста и формирование ответа.
3. Объект функционального контроля (ФК) – приём запросов и формирование ответа о своём техническом состоянии (мониторинг).

Таким образом, возложив задачи динамической реконфигурации [25] встроенной системы тестового диагностирования на УУК, дополнительные ресурсы ОД использоваться не будут. Это, в совокупности с самой реконфигурацией позволит повысить надёжность элементов ВС [24, 31].

Рассмотрим модель разрабатываемой встроенной системы диагностирования (рис. 3).

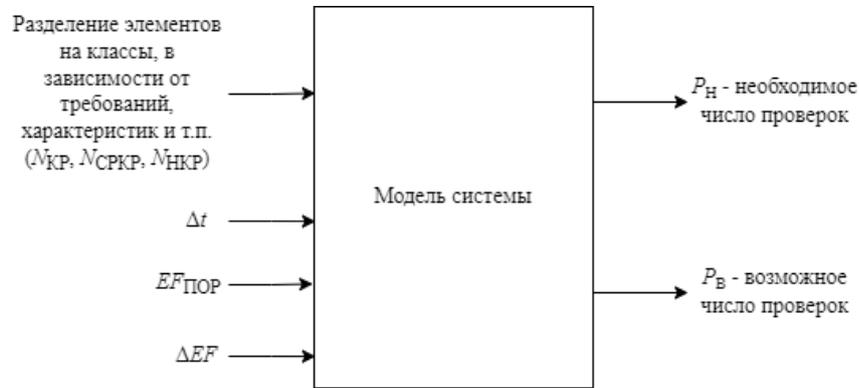


Рисунок 3. Входные и выходные данные модели системы встроенного диагностирования

При разработке модели встроенной системы диагностирования за основу взята диагностическая модель Препарата – Метца – Чена [93, 105], в которой модель представляется в виде графа [64, 92], вершинами которого являются элементы системы, а рёбрами – межэлементные диагностические связи [56, 63]. Это значит, что ребро показывает направление проведения диагностических тестов. В данной модели каждый элемент диагностируется один раз, одним диагностирующим элементом – агентом.

Новизна разрабатываемой методики заключается в учёте характеристик производительности элементов ВС при распределении диагностических функций между этими элементами. Такая система состоит из однотипных элементов, способных диагностировать друг друга одинаковыми по затрачиваемым ресурсам и времени тестами. Можно разделить систему на несколько однотипных блоков [74]. В таком случае, представленная модель будет описывать каждый блок системы по отдельности, но в совокупности представлять такие же диагностические результаты, как если бы она была едина для разнотипных элементов.

Особенности разрабатываемой модели включают в себя:

1. Предлагается разделить элементы на произвольное количество классов, в зависимости от требований, характеристик, предпочтений и т.п. Далее для примера рассмотрим 3 класса элементов:

Высокой критичности – элементы, на которые возложены самые важные и затратные операции в системе, их необходимо диагностировать в первую очередь, и не использовать в качестве диагностического инструмента.

Средней критичности – элементы, которые и диагностируют, и являются диагностируемыми. Половина из этого множества элементов в определённый момент времени диагностирует, когда оставшаяся диагностируется.

Низкой критичности – элементы, которые используются в качестве диагностирующих элементов постоянно, это наименее загруженные элементы системы.

2. Энергетические и вычислительные затраты реконфигурации диагностирования берёт на себя УУК, то есть данный процесс не тратит ресурсы диагностирующих и диагностируемых элементов.

3. Задаются:  $\Delta t$  – диагностический интервал (интервал реконфигурации),  $N_{\text{КР}}$  – число элементов высокой критичности,  $N_{\text{СРКР}}$  – число элементов средней критичности,  $N_{\text{НКР}}$  – число элементов низкой критичности и их перечень,  $EF_{\text{ПОР}}$  – пороговое значение характеристики надежности элемента вычислительной системы, ниже которого элемент считается неработоспособным для системы диагностирования.

4.  $\Delta EF$  – изменение показателя надежности элемента вычислительной системы при проведении тестирования. Принимается за константу для элементов вычислительной системы. Для определения  $\Delta EF$  необходимы предварительные испытания на оборудовании и расчёт по предложенным далее методам. При использовании разработанных моделей встроенной системы

диагностирования и элементов в вычислительной системе с автономными элементами наиболее важным показателем работоспособности при расчёте данной величины необходимо принимать заряд аккумулятора  $Z$ .

На рис. 4 приведён пример диагностического графа для системы с элементами разной критичности для момента времени  $t$  и  $t+\Delta t$ .

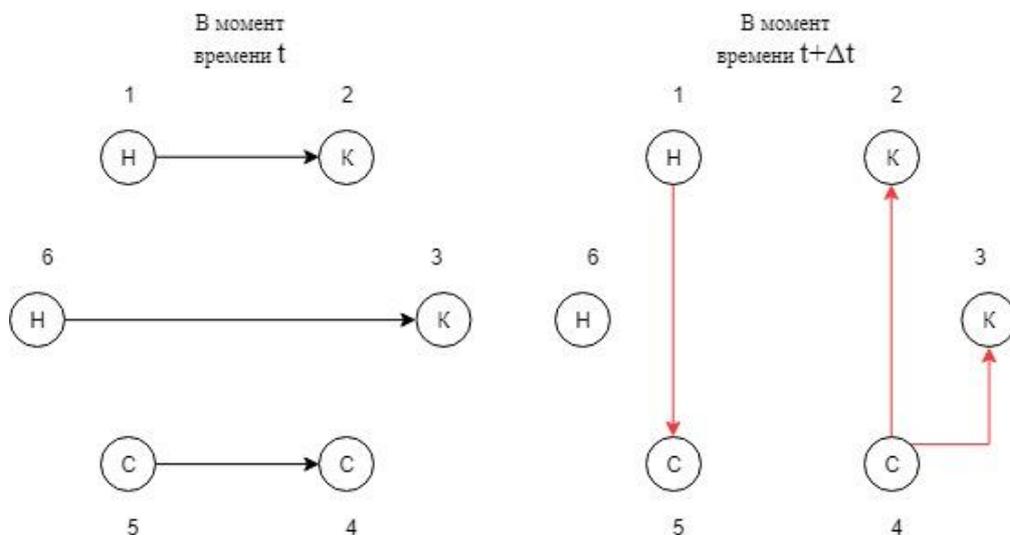


Рисунок 4. Пример диагностического графа по описанной модели с возможностью реконфигурирования в момент времени  $t$  и  $t+\Delta t$ . Н – элементы низкой критичности, С – элементы средней критичности, К – элементы высокой критичности.

На рисунке представлен пример реконфигурации назначения диагностических функций системы, состоящей из 2-х элементов высокой критичности, 2-х элементов средней критичности и 2-х элементов низкой критичности. В момент времени  $t$ , численная величина производительности каждого элемента (расчёт производит УУК, собирая статистику перед каждым циклом тестового диагностирования) позволила определить следующий диагностический граф для системы (левый граф на рис. 4). На нём видно, что необходимо продиагностировать элементы высокой критичности и в момент времени  $t$  с минимальными затратами с этим справятся элементы 1,5,6. В следующий момент времени  $t+\Delta t$ , показатели производительности элементов изменились, как и численная оценка, позволившая УУК перераспределить диагностические функции.

Математическая модель разработанной системы определяется следующими показателями (все величины рассчитываются в начале каждого цикла диагностирования) [81]:

1. Число необходимых проверок перед началом диагностирования:

$$P_H = N_{\text{КР}} + (N_{\text{СРКР}} - \left\lfloor \frac{N_{\text{СРКР}}}{2} \right\rfloor), \quad (3)$$

где  $N_{\text{КР}}$  - число работоспособных элементов высокой критичности,  $N_{\text{СРКР}}$  - число работоспособных элементов средней критичности, а  $\lfloor \rfloor$  - операция округления в меньшую сторону.

2. Максимальное число проверок (возможное), которое может быть проведено в данный диагностический интервал:

$$P_B = \sum_{i=1}^{N_d} P_{B_i}, \quad (4)$$

где  $N_d = N_{\text{НКР}} + \left\lfloor \frac{N_{\text{СРКР}}}{2} \right\rfloor$ , а  $P_{B_i} = \left\lfloor \frac{EF_i - EF_{\text{ПОР}}}{\Delta EF} \right\rfloor$ ,  $N_{\text{НКР}}$  - число работоспособных элементов низкой критичности,  $EF_i$  - показатель надежности  $i$ -го элемента в начале цикла диагностирования (рассчитывается по предложенным далее методам). При значении показателя надежности ниже порогового значения ( $EF_{\text{ПОР}}$ ) элемент больше не является тестером и не учитывается в вышеуказанных характеристиках.

### 2.3 Разработка диагностической модели элемента встроенной системы тестового диагностирования

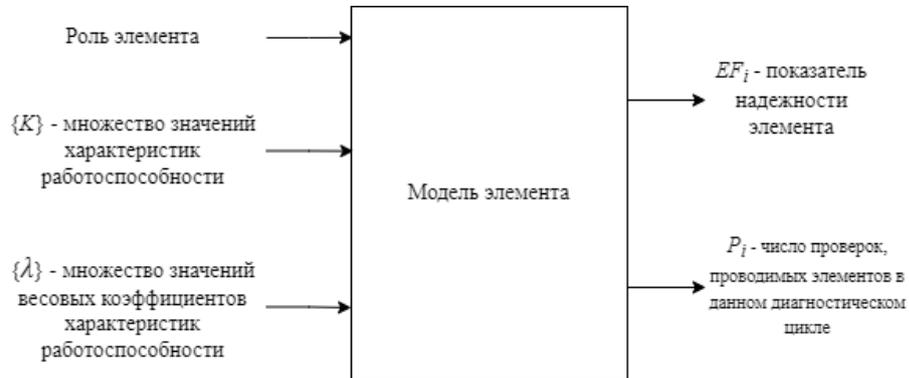


Рисунок 5. Модель элемента вычислительной системы

*Диагностическая модель элемента вычислительной системы:*

1. Показатель надежности элемента. Данный показатель необходим для оценки текущего состояния элемента вычислительной системы и определения возможности выполнения данным элементов функции тестера в системе.

$$EF = f(\{K_j\}, \{\lambda_j\}) \quad (5)$$

где  $K_j$  – значение  $j$ -го критерия работоспособности, а  $\lambda_j$  – весовой коэффициент  $j$ -го критерия работоспособности,  $j \in [1; N]$ ,  $N$  – число критериев работоспособности. В общем случае, показатель надежности – некоторая функция (4), которая зависит от текущих значений характеристик работоспособности элемента и значений их важности. В различных системах, в зависимости от используемого оборудования, условий эксплуатации, степени распределённости и т.п. различные характеристики работоспособности должны быть учтены в интегральном показателе надежности с различной важностью (весом).

1.1. Показатель надежности элемента, учитывающий один показатель работоспособности:

$$EF_i = K_i \quad (6)$$

где  $K_i$  – значение показателя работоспособности  $i$ -го элемента в начале цикла диагностирования,  $K_i$  – нормировано в диапазон  $[0;1]$ , 0 – худшее значение показателя работоспособности, а 1 – наилучшее. Если показателем работоспособности является оставшаяся мощность заряда батареи автономного элемента вычислительной системы (измеряемой в процентах), то  $K_i = Z_i$ ,  $Z_i \in [0;1]$ . В случае такой характеристики, как например загруженность процессора –  $CPU_i$ ,  $K_i = 1 - CPU_i$ ,  $CPU_i \in [0;1]$ .

### 1.2. Показатель надежности элемента (2 и более параметров).

Для того, чтобы определить показатель надежности элемента (определить, способен ли элемент производить тестовое диагностирование над другими элементами) необходимо сначала определить весовые коэффициенты учитываемых показателей работоспособности. Для этого можно воспользоваться разработанным методом из главы 3, или прибегнуть к экспертной оценке. Результатом применения любого из этих методов является некоторое множество весовых коэффициентов показателей работоспособности элемента вычислительной системы.  $M_\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$  – где  $N$  – число показателей работоспособности элемента в данной вычислительной системе.

После получения весовых коэффициентов их необходимо применить для расчёта текущей характеристики производительности. Для данного расчёта был разработан метод, описанный в главе 3. Допустимым вариантом расчёта является метод средневзвешенного значения:

$$EF_i = \sum_{j=1}^N \lambda_j \times K_{ij}, \quad (7)$$

где  $N$  – число показателей работоспособности,  $j$  – порядковый номер показателя работоспособности,  $\lambda_j$  – весовой коэффициент  $j$ -го показателя

работоспособности из множества  $M_\lambda$ ,  $K_{ij}$  – значение  $j$ -го показателя работоспособности  $i$ -го элемента вычислительной системы.

2. Число проверок, которое должен провести каждый элемент:

$$P_i = \frac{EF_i \times P}{\sum_{i=1}^{N_d} EF_i}, \quad (8)$$

где  $P$ – число проверок для данного диагностического цикла, зависит от логического соотношения (3) и (4), описанного в главе 4.

## 2.4 Выводы по главе

В данной главе были разработаны математическая модель встроенной системы диагностирования и диагностическая модель элемента вычислительной системы. Недостатки существующих моделей приводят к выводу о том, что процесс диагностирования необходимо оптимизировать и считать его частью нагрузки, приходящейся на систему. Если раньше, в виду меньшей распространённости вычислительных систем в жизни общества, данную нагрузку можно было считать бесконечно малой, по сравнению с нагрузкой основного функционала системы, то сейчас игнорировать её невозможно. Таким образом, разработка моделей, методов и алгоритмов оптимизации диагностических процессов является актуальной задачей.

Разработанные модели позволяют унифицировать подход к оценке технического состояния элементов вычислительной системы. Математическая модель встроенной системы диагностирования позволяет не только рассчитать текущие параметры системы, но на основе их сравнения прогнозировать будущее состояние работоспособности вычислительной системы и её элементов.

Математические выражения, определяющие параметры разработанных моделей разработаны таким образом, чтобы обеспечить наиболее простую реализацию программных расчётов. Этим обеспечивается дополнительная эффективность расчётов по разработанным моделям.

### Глава 3. Методы определения характеристик моделей элементов встроеной системы диагностирования

#### 3.1 Метод определения важности критериев работоспособности элемента вычислительной системы

В разрабатываемом методе требуется совместить знания о критериях и их вкладе в результат принятия решений с возможностью усиления вклада дополнительной субъективной экспертной оценкой. Существующие методы, например «Теория важности критериев» Подиновского В.В. [35] или методы, описанные в работах Бойко А.А., Дягтерёва И.С. [3] и наших зарубежных коллег Saaty T.L. [97] позволяют вычислить весовые коэффициенты учитываемых характеристик, но часть этих методов невозможно эффективно автоматизировать, а другую часть применить в рассматриваемой области. Для разработки методики необходимо ввести некоторые определения [21]:

1. Характеристика (системы или элемента) – количественный интегральный показатель системы или элемента.

2. Критерий характеристики – количественный показатель системы или элемента, влияющий определённым (положительным или отрицательным) образом на одну или несколько характеристик.

Необходимо применить условие нормирования, которое говорит о том, что:

$$\sum_{j=1}^{N_{Cr}} \lambda_j = \sum_{j=1}^{N_{Cr}} \left( \sum_{k=1}^{N_{Ch}} U_{jk} \times C_{Chj} \right) \times x = 1 \quad (9)$$

где  $U_{ij}$  – численное значение факта участия критерия характеристики в формировании значения характеристики (0 - при отсутствии влияния на значение характеристики, 1 – при наличии влияния критерия на значение характеристики и равнозначности влияния каждого критерия на характеристики,  $[0;Z]$  – при неравнозначном влиянии критериев на значение характеристик,  $Z$  – максимальное значение относительного влияния критерия),

$C_{chj}$  – значение покрытия характеристики критериями,  $x$  – нормирующий коэффициент.

Если в системе, при принятии решений [62] в условиях многокритериальности необходимо численно определить важность критериев и данным критериям в однозначное соответствие можно поставить численные критерии характеристик системы или элементов, то полученные в результате использования разработанной методики веса можно использовать для определения важности критериев.

На *первом* этапе расчёта по предлагаемой методике необходимо каждому оцениваемому критерию привести в соответствие критерий характеристики элемента или системы. Так, например, в технических системах, если необходимо рассчитать коэффициент важности для такой характеристики как отказоустойчивость кластера, то ей можно поставить в соответствие такой критерий характеристики как: число элементов кластера, число попыток переотправки и т.д.

На *втором* этапе необходимо определить характеристики системы и её элементов, на значение которых оказывают влияние выделенные на первом этапе критерии характеристики.

На *третьем* этапе методики необходимо сформировать таблицу, в которой строками будут являться характеристики системы и элементов, а столбцами – критерии характеристик. С помощью эксперта (можно на этапе проектирования) проставить соответствие влияния критерия характеристики на саму характеристику. С помощью «+» обозначается наличие влияния значения критерия на значение характеристики. Отсутствие «+» означает в свою очередь отсутствие весомого влияния.

Таблица 2

Таблица соответствия после 1-го этапа

	$Cr_1$	$Cr_2$	$Cr_3$	$C_{Ch}$
$Ch_1$	+	+	-	2
$Ch_2$	-	+	-	1
$Ch_3$	-	-	+	1
$C_{Cr}$	1	2	1	4

На данном этапе уже можно посчитать весовые значения критериев характеристик. Для этого необходимо воспользоваться формулой и решить уравнение нормирования (9) относительно коэффициента нормирования:

$$\sum_{j=1}^{N_{Cr}} \lambda_j = \sum_{j=1}^{N_{Cr}} \left( \sum_{k=1}^{N_{Ch}} U_{jk} \times C_{Chj} \right) \times x = (2 \times 1 + 1 \times 0 + 1 \times 0)x + 3x + x = 6x = 1 \quad (10)$$

Соответственно, так как:

$$\lambda_j = \left( \sum_{k=1}^{N_{Ch}} U_{jk} \times C_{Chj} \right) \times x \quad (11)$$

То,

$$\begin{aligned} \lambda_1 &= (2 \times 1 + 1 \times 0 + 1 \times 0) \times \frac{1}{6} = \frac{1}{3}, \\ \lambda_2 &= (2 \times 1 + 1 \times 1 + 1 \times 0) \times \frac{1}{6} = \frac{1}{2}, \\ \lambda_3 &= (2 \times 0 + 1 \times 0 + 1 \times 1) \times \frac{1}{6} = \frac{1}{6}. \end{aligned}$$

На *четвертом* этапе, также при помощи эксперта возможно увеличить относительную степень влияния критерия на характеристику при помощи увеличения числа плюсов в их пересечении.

Таблица 3

Таблица соответствия после 4-го этапа

	$Cr_1$	$Cr_2$	$Cr_3$	$C_{Ch}$
$Ch_1$	++	+		3
$Ch_2$		+++		3
$Ch_3$			+	1
$C_{Cr}$	2	4	1	7

Используя формулу (9), а затем (12) подставляя правильные значения участия критерия в значении характеристики получим обновленные значения весов:

$$\sum_{j=1}^{N_{Cr}} \lambda_j = \sum_{j=1}^{N_{Cr}} \left( \sum_{k=1}^{N_{Ch}} U_{jk} \times C_{Chj} \right) \times x = (3 \times 2 + 3 \times 0 + 1 \times 0) \times x + 10x + x = 17x = 1 \quad (12)$$

$$\lambda_1 = (3 \times 2 + 3 \times 0 + 1 \times 0) \times \frac{1}{17} = \frac{6}{17},$$

$$\lambda_2 = (3 \times 1 + 3 \times 3 + 1 \times 0) \times \frac{1}{17} = \frac{10}{17},$$

$$\lambda_3 = (3 \times 0 + 3 \times 0 + 1 \times 1) \times \frac{1}{17} = \frac{1}{17},$$

Стоит отметить, что данный метод является масштабируемым как на число критериев, так и на число учитываемых характеристик. Данная методика готова к внедрению как на этапе проектирования, так и на этапе эксплуатации. В процессе эксплуатации значения весовых коэффициентов могут быть пересчитаны, что полезно при решении задачи оптимизации распределения нагрузки.

Для решения задачи определения важности критериев работоспособности элементов вычислительной системы, предлагается принять за характеристики системы характеристики надежности, представленные в ГОСТе по надёжности [10], а именно – составляющие характеристики безотказности, долговечности, ремонтпригодности и сохраняемости. Критерии работоспособности элементов вычислительной системы, которые влияют на данные характеристики

выбираются исходя из особенностей вычислительной системы, её элементов и условий эксплуатации.

В данной работе рассматриваются системы с автономным питанием элементов, каждый из которых оснащён процессором и дисковой памятью. В этом случае перечень критериев работоспособности:  $Z$  – заряд батареи,  $CPU$  – загрузка процессора,  $D_{SPACE}$  – оставшаяся дисковая память,  $D_{SPEED}$  – скорость чтения/записи,  $N_{SN}$  – соотношение сигнал/шум в среде передачи информации. Соответственно, пример таблицы для первого этапа при решении задачи определения важности характеристик работоспособности элемента вычислительной системы может быть следующий:

Таблица 4

Пример таблицы связей между характеристиками и их критериями в вычислительной системе

	$Z$	$CPU$	$D_{SPACE}$	$D_{SPEED}$	$N_{SIGNOIZE}$	$COV_{CHAR}$
Безотказность	+	+	+	+	-	4
Долговечность	+	+	+	-	-	3
Ремонтопригодность	+	+	-	+	+	4
Сохраняемость	+	-	+	-	+	3
$COV_{CR}$	4	3	3	2	2	14

Проведя расчёт и уточнение степени влияния критерия характеристики на характеристику получим весовые коэффициенты, которые можно использовать для решения задачи определения показатели надежности элемента и последующего перераспределения диагностической нагрузки.

### 3.2 Метод определения показателя работоспособности элемента вычислительной системы

Существующие методы определения интегрального показателя при решении многокритериальных задач, описанные в работах [46, 59, 85, 89] имеют некоторые недостатки, которые не позволяют использовать их для решения поставленной задачи. К ним относятся: несогласованность экспертной оценки [37, 76], эффекты компенсации, сложность автоматизации. Таким образом, разработка метода определения показателя работоспособности элемента вычислительной системы, который бы испытывал меньше эффекты компенсации в критичных ситуациях и метода, который мог быть легко автоматизирован, является актуальной задачей исследования.

Необходимо представить элемент вычислительной системы с текущими значениями рабочих характеристик в виде конуса единичной высоты и единичного максимального радиуса. В цилиндрических координатах по оси  $Z$  наносится вес характеристики, а значение характеристики определенного веса задается с помощью радиуса  $r$  в радиальной плоскости, параллельной основанию, на высоте, равной значению мощности (Рис. 6).

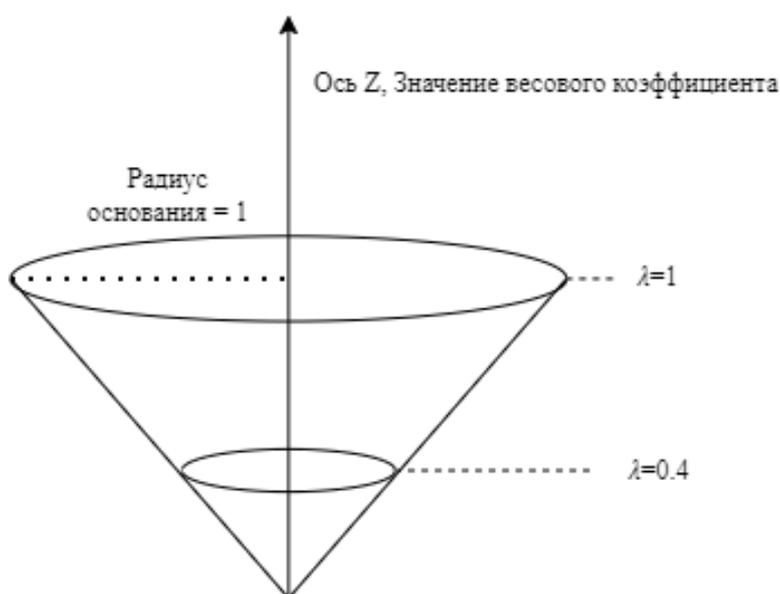


Рисунок 6. Описание представления элемента вычислительной системы

Эта модель позволяет:

- Учитывать разную мощность характеристик, располагая их на желаемой высоте по оси  $Z$ .
- Одинаково учитывать характеристики с равными весовыми коэффициентами, разместив их в модели на одинаковой высоте и в одной радиальной плоскости, параллельной основанию, на высоте, численно равной весу характеристики. В этом случае площадь поперечного сечения равномерно распределяется между характеристиками с одной мощностью.

Чтобы рассчитать показатель работоспособности элемента, необходимо выполнить следующие шаги:

1. Вычислить объем фигуры, полученной обработкой текущих значений характеристик производительности элемента вычислительной системы. Для этого необходимо описать, как учитываются площади поперечного сечения, образованные характеристиками производительности на каждом уровне. Пример показан на рис. 7.

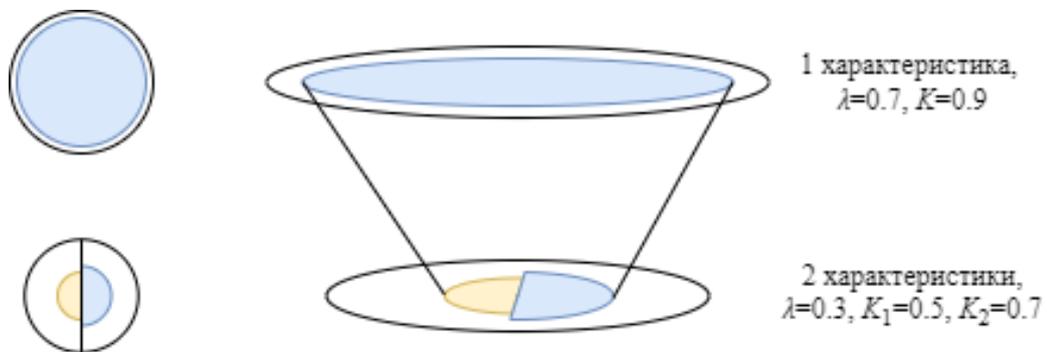


Рисунок 7. Пример 2-х уровневой модели элемента, чей объем должен быть рассчитан для определения показателя работоспособности

Чтобы определить необходимую величину, нужно выполнить несколько шагов:

1. Рассмотреть два ближайших уровня мощности и посчитать сегменты интегрирования для объема между двумя этими уровнями. Формула (13) показывает, как посчитать набор углов для одного уровня:

$$M_{ang}(l) = \left\{ \frac{2 \times \pi \times n}{N(l)} \right\}, n \in \{0, \dots, N(l)\}, \quad (13)$$

где  $l$  –уровень важности и  $N(l)$  – число характеристик с весовым коэффициентом, равным  $l$ .

Затем должны быть определены все угловые отсечки для данного уровня:

$$M_{ang}(l_1, l_2) = M_{ang}(l_1) \cup (M_{ang}(l_2) - M_{ang}(l_1) \cap M_{ang}(l_2)), \quad (14)$$

$$N_{ang}(l_1, l_2) = |M_{ang}(l_1, l_2)| \quad (15)$$

2. Расчёт объёма сегмента между двумя уровнями

$$V_{segment}(l_1, l_2, s) = \int_{R(l_1, s)}^{R(l_2, s)} \rho d\rho \int_{M_{ang}(l_1)_{s-1}}^{M_{ang}(l_2)_s} d\varphi \int_{\frac{(\rho - R(l_1, s)) \times (\lambda(l_2) - \lambda(l_1))}{R(l_2, s) - R(l_1, s)} + \lambda(l_1)}^{\lambda(l_2)} dz, \quad (16)$$

$$R(l, s) = K(s) \times \sqrt{(1 - (\lambda(l) - 1))^2}, \quad (17)$$

где  $\lambda(l)$  – вес на уровне  $l$  и  $K(s)$  – значение характеристики в сегменте.

3. Просуммировать все объемы между двумя уровнями

$$V(l_1, l_2) = \sum_{s=1}^{N_{ang}(l_1, l_2)} V_{segment}(l_1, l_2, s) \quad (18)$$

4. Посчитать значение объёмов между всеми парами ближайших уровней

$$V_{current} = \sum_{l=2}^{N_{power}} V(l-1, l), \quad (19)$$

где  $N_{power}$  – число различных уровней.

После выполнения этих шагов получена характеристика, которая показывает текущее техническое состояние элемента вычислительной системы. Чтобы определить значение показателя надежности элемента вычислительной системы, нам необходимо рассчитать объем, когда элемент имеет наилучшие значения своих характеристик. Этот объем можно вычислить по тому же

алгоритму. Но есть более простое решение (для вычисления в автоматическом режиме), когда необходимую величину можно вычислить по следующей формуле:

$$V_{\max} = \frac{\pi \times (\lambda(l_{\max}) - \lambda(l_{\min})) \times (R(l_{\max})^2 + R(l_{\min})^2 + R(l_{\max}) \times R(l_{\min}))}{3}, \quad (20)$$

где  $l_{\max}$  and  $l_{\min}$  – уровни с максимальным и минимальным весом.

Для расчета значения показателя работоспособности элемента вычислительной системы необходимо найти соотношение между  $V_{current}$  и  $V_{max}$ :

$$EF = \frac{V_{current}}{V_{max}} \quad (21)$$

Значение  $EF \in [0;1]$  показывает относительное состояние эффективного состояния элемента по отношению к идеальному техническому состоянию элемента.

*Исследование разработанного метода определения показателя надежности элемента вычислительной системы.*

Для сравнения разработанного метода и метода средневзвешенного значения при определении показателя надежности была написана программа моделирования [68, 98, 100] с пользовательским интерфейсом. Программа написана на языке Javascript при помощи фреймворка Ember.js [55]. Основной целью программы является перебор всевозможных вариантов рассматриваемых ситуаций, расчёт показателей надежности элементов вычислительной системы по двум алгоритмам (средневзвешенного и разработанного), представление данных в виде таблиц. Входными параметрами для программы являются перечень учитываемых показателей работоспособности (критериев характеристик надежности) и их весовые коэффициенты, шаг изменения значений критериев и пороговые значения для показателей работоспособности.

Сравнению будет подвергаться влияние эффектов компенсации на результирующий показатель надежности элемента вычислительной системы. Входными параметрами программы являются перечень характеристик, их

мощности и шаг разности значений. После ввода параметров программа вычисляет все возможные значения характеристик с введенным шагом и вычисляет коэффициент полезного действия по двум моделям – разработанной и взвешенной средней.

Критерием эффективности будет оцениваться с помощью значения  $\Delta$ :

$$\Delta = \frac{\sum_{i=1}^N (EF_W - EF_{DEV})}{N} \quad (22)$$

Соответственно, чем больше значение  $\Delta$ , тем больше вероятность, что пороговое значение попадет в интервал между  $EF_W$  и  $EF_{DEV}$  и разработанный алгоритм определит элемент как неработоспособный.

Первая ситуация, которую необходимо исследовать - это ситуация, когда есть одна характеристика с наибольшим значением веса и её текущее значение ниже порогового. Рассмотрение такой ситуации позволит оценить влияние множества менее важных характеристик на итоговый показатель надежности элемента. В этом случае ожидаемым результатом является меньший показатель работоспособности, т.к. значение важнейшей характеристика ниже порогового значения. Эксперименты проводятся для разного количества характеристик (от 2 до 6), с шагом значений = 0,1 и шагом весовой характеристики = 0,05.

Таблица 5

Результаты эксперимента для 1-го случая

<i>Число параметров</i>	<i>Число рассматриваемых ситуаций</i>	<i>Ситуации, в которых разработанный алгоритм дает меньшее значение EF, чем существующий (%)</i>	<i>Средняя <math>\Delta</math></i>
2	33	97.98	0.112
3	363	98.65	0.144
4	3993	99.66	0.171
5	43923	100	0.185
6	483153	100	0.201

В первом столбце указано числа критериев работоспособности, во втором – число возможных ситуаций, удовлетворяющих условиям первого рассматриваемого случая. Третий столбец показывает процент ситуаций, в которых разработанный алгоритм оказался эффективнее существующего (средне - взвешенного). В четвертом столбце указана средняя разница между значениями рассчитанных показателей надежности элементов.

Эти результаты показывают, что с увеличением количества характеристик количество успешных случаев также увеличивается. Аналогичное увеличение наблюдается и в проценте эффективности.

Второй случай – это ситуация, когда все характеристики, кроме одной с наибольшим весом имеют значения ниже порогового. В этой ситуации ожидается, что по разработанному методу элемент вычислительной системы в большей степени определялся как неработоспособный. Результаты моделирования показаны ниже в Таблице 6.

Таблица 6

Результаты эксперимента для 2-го случая

<i>Число параметров</i>	<i>Число рассматриваемых ситуаций</i>	<i>Ситуации, в которых разработанный алгоритм дает меньшее значение EF, чем существующий (%)</i>	<i>Средняя <math>\Delta</math></i>
2	33	100	0.191
3	99	100	0.163
4	297	99.72	0.149
5	891	99.35	0.142
6	2673	98.75	0.136

Эти результаты показывают, что с увеличением количества характеристик количество средних случаев успеха уменьшается, но уменьшается медленно, также как среднее значение разности показателей

надежности, рассчитанных по предложенным методам. Дальнейшее моделирование показало, что при числе параметров более 8 разница между показателями нивелируется и эффективность алгоритмов становится практически одинаковой.

### **3.3 Выводы по главе**

В данной главе были представлены разработанные методы определения важности критериев при решении многокритериальной задачи и метод определения показателя работоспособности элемента вычислительной системы. Первый метод позволяет совместить объективные знания о критериях и их характеристиках для автоматизированного и быстрого расчёта весовых коэффициентов учитываемых характеристик. Данный метод позволяет изменить степень влияния каждого критерия характеристики при помощи экспертов или согласованной экспертной оценки. Это даёт возможность учесть специфику конкретной вычислительной системы или её элемента для более правильного расчёта весовых коэффициентов характеристик. Это важно, так как в высоконагруженных системах любая многокритериальная задача распределения нагрузки (или функциональных задач) требует максимально эффективного как с точки зрения ресурсов, так и с точки зрения времени выполнения задачи распределения между элементами. Чем точнее и корректнее будут посчитаны весовые коэффициенты тем более правильно любая задача распределения будет решена в конкретной системе.

Метод определения показателя работоспособности очень важен для вычислительных систем. Чем точнее будет посчитан показатель работоспособности, тем более правильное и эффективное на данный момент решение о распределении любого рода задач между элементами может быть принято. Существующие методы сильно подвержены эффектам компенсации, что при решении задачи определения технического состояния элемента

вычислительной системы является критичным фактором. Эффекты компенсации могут представить элемент как работоспособный в ситуациях, когда на самом деле множество малозначимых факторов своими показателями «перекрывают» низкое значение важных показателей работоспособности. Эксперт смог бы дифференцировать такие ситуации, но в настоящий момент требуются максимально оперативные и автоматические принятые решения в виду сложности и нагруженности вычислительных систем.

Проведенные эксперименты показали преимущества разработанного алгоритма. Рассмотренные ситуации тестируют разработанный метод на подверженность эффектам компенсации. По сравнению с существующим методом средне – взвешенного, данная разработка показывает лучшие значения.

Можно сделать вывод о том, что методы, разработанные в данной главе, имеют преимущества по сравнению с существующими при решении задачи распределения диагностической нагрузки между элементами вычислительной системы. Преимущества для первого метода заключаются в возможности автоматизации метода расчёта весовых коэффициентов учитываемых характеристик, а также возможности уточнения экспертами степени влияния характеристик на критерии, что позволяет учесть специфику конкретной системы при решении многокритериальной задачи. Преимуществом второго метода является уменьшенная, по сравнению с существующими аналогами, подверженность эффектам компенсации, что даёт, в свою очередь, более эффективное определение состояние элемента и распределение нагрузки между элементами.

#### **Глава 4. Разработка алгоритма распределения диагностической нагрузки между элементами встроенной системы тестового диагностирования**

В силу развития беспроводных технологий [18, 73, 66, 104] широкое распространение получили автономные элементы ВС. Примерами автономных элементов ВС могут являться автономные маршрутизаторы TP-Link TL-MR3040, энергетически автономные продукты компании CISCO (Cisco Aironet 1240), автономные датчики контроля утечек и пожарной безопасности [86] на нефтяных месторождениях [107]. Это актуально и для мобильных устройств [27] и AD-НОС сетей [104], так как в них существует автономное питание элементов.

Для различных вычислительных систем характеристики производительности её элементов могут быть разные. Таким образом, одним из требований к разрабатываемому алгоритму [20, 79] становится масштабируемость в плане учёта любого числа характеристик, влияющих на производительность элементов.

Как известно, снижение *уровня заряда батареи* приводит к нарушению функционированию элементов, что приводит к уменьшению такого показателя надёжности как среднее время безотказной работы (наработка на отказ). В большинстве современных элементах реализованы встроенные средства диагностирования, которые занимают часть *производительности процессорных устройств* и, соответственно, ускоряют разряд батареи. Таким образом, перераспределяя загрузку элементов диагностическими функциями, можно выборочно уменьшать скорость разряда и повысить показатели надёжности (безотказности) элементов ВС.

Для выполнения основных функций УУК разработан новый алгоритм реконфигурирования встроенной подсистемы тестового диагностирования. Данный алгоритм позволяет одновременно с решением задачи распределения

диагностической нагрузки решать некоторые задачи мониторинга. Заложенные в нём методы расчёта показателей моделей системы и элемента, а также проверка некоторых условий решают задачу прогнозирования возникновения неисправностей и их предотвращения. Данные преимущества, а также возникающая кластеризация отказов положительно влияет не только на время работоспособности системы из формулы (1), но и на время ремонта.

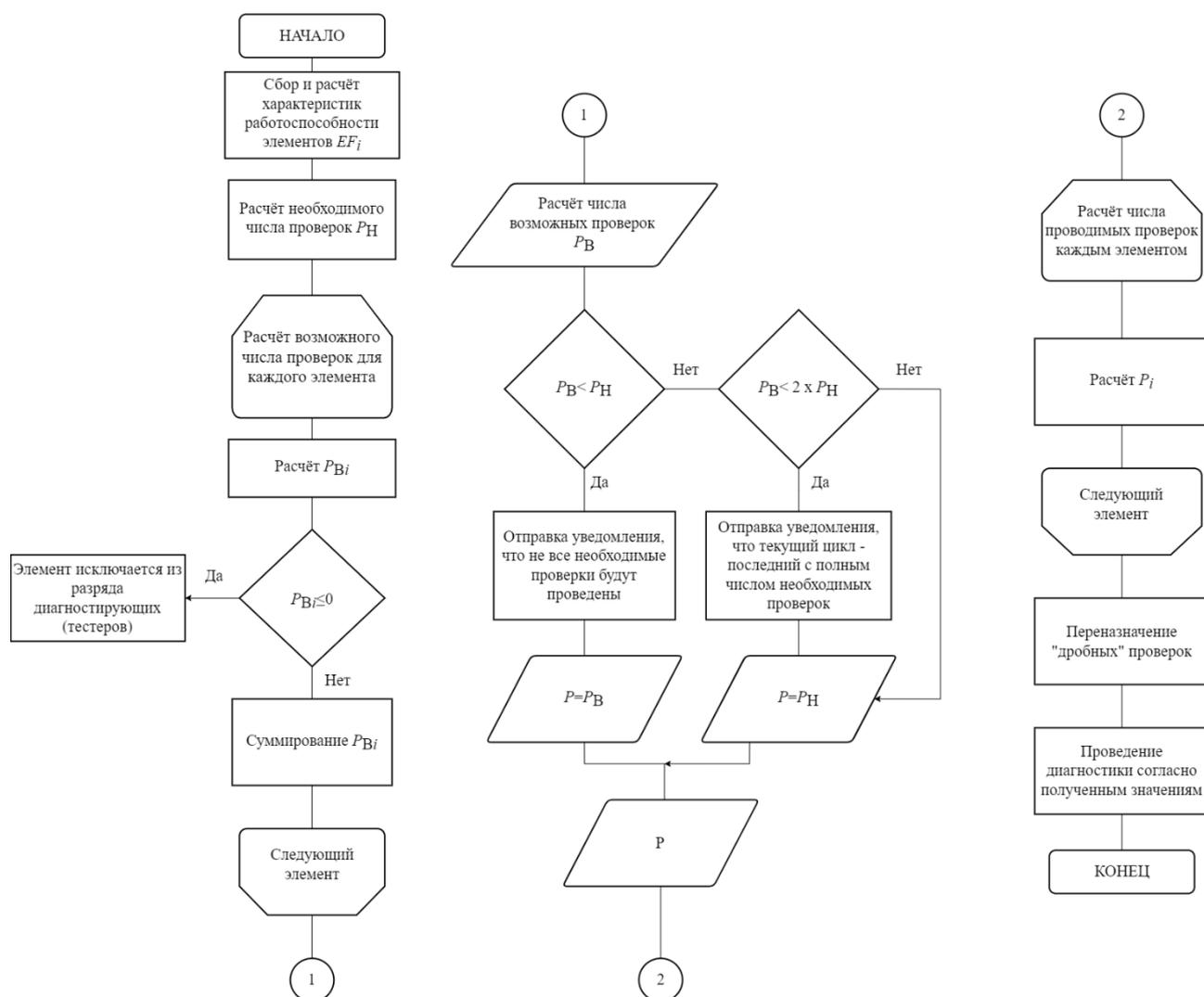


Рисунок 8. Схема разработанного алгоритма реконфигурации

#### 4.1 Первый этап. Расчёт численных значений моделей системы и её элементов

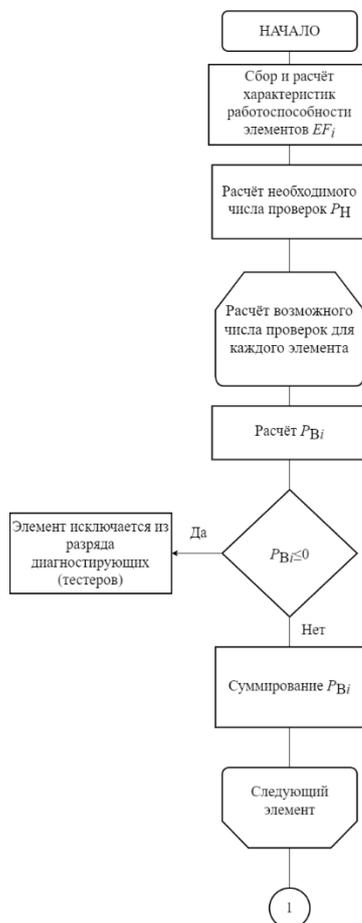


Рисунок 9. Первый этап работы алгоритма

В начале первого этапа происходит расчёт основных показателей по разработанным моделям системы и элементов.

1. Опрос и сбор текущих значений учитываемых характеристик работоспособности элементов системы.

2. Расчёт значений  $EF$  для каждого элемента и расчёт необходимого числа проверок для системы. В рамках этого шага происходит первый этап мониторинга и оповещения о состоянии элементов вышедших из строя исходя из значений  $EF$  (ниже или около порогового значения).

3. В рамках следующего шага происходит расчёт возможного числа проводимых проверок для каждого элемента. На данном этапе работает

мониторинг исключения элемента из разряда диагностирующих, а также уведомление соответствующих систем / служб / сотрудников.

4. Расчёт суммарного значения возможных проверок, которые могут быть проведены системой.

На первом этапе алгоритма рассчитаны основные значения разработанных моделей, т.е. получено значение нагрузки (числа возможных и необходимых проверок), которая будет распределяться между элементами вычислительной системы, и получена информация о элементах и возможности / невозможности проведения этими элементами диагностических проверок.

#### 4.2 Второй этап. Определение параметров диагностирования

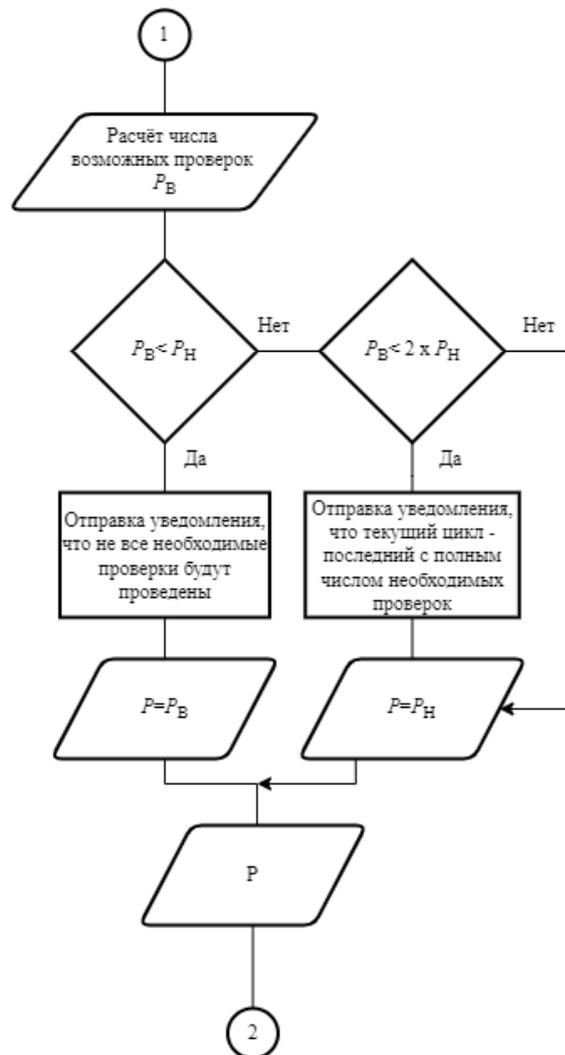


Рисунок 10. Схема второго этапа разработанного алгоритма

Основной задачей второго этапа является определение окончательного числа проверок, которые будут проведены в рамках текущего диагностического цикла, а также уведомления об уменьшении проводимого числа проверок в рамках диагностирования.

Если,  $EF_i \leq EF_{\text{ПОР}}$ , то элемент исключается из разряда диагностируемых с соответствующим оповещением АСТ. Также, если:

$$P_B < P_H,$$

то  $P = P_B$ .

В данном случае происходит отправка уведомления АСТ о том, что не все необходимые проверки будут выполнены в текущий момент времени. Необходимые действия в данном случае – проведение регламентных работ по повышению показателя надежности элементов с низким значением данного показателя.

Также, если:

$$P_B < 2 \times P_H,$$

то, происходит отправка уведомления АСТ о том, что текущий диагностический цикл – крайний цикл с полным числом необходимых проверок.

Если первые два условия не выполнены, тогда:

$$P = P_B$$

Этот этап очень важен, так как в его рамках происходит определение реальной нагрузки производимой встроенной системой диагностирования. Уведомления, генерируемые на данном этапе, позволяют не только предупредить о уже возникшей неисправности, но и предотвратить её возникновение путем учета изменения числа возможных проверок, а также предрасчёта аналогичного шага следующего диагностического цикла.

### 4.3 Третий этап. Расчёт значений модели элемента системы диагностирования. Распределение нагрузки. Диагностирование



Рисунок 11. Схема 3-го этапа разработанного алгоритма

В рамках третьего этапа происходит непосредственно распределение нагрузки (диагностических задач) между элементами вычислительной системы.

Для начала по формуле (8) производится расчёт числа проверок, которые будут проведены каждым элементом, исходя из значения его работоспособности (надежности). В силу того, что значения округляются до целых, то остается некоторое нераспределенное число проверок, которые необходимо распределить между элементами, участвующими в диагностике. Этот процесс происходит итеративно следующим образом:

1. Элементы сортируются исходя из значения их коэффициента работоспособности по убыванию.

2. Начиная с первого элемента рассчитывается, сколько дополнительных проверок может взять на себя данный элемент.

3. Элементу назначается максимальное значение дополнительных проверок.

Стоит отметить, что все дополнительные проверки обязательно будут распределены, так как на втором этапе конечное число проверок меньше или равно возможному числу проверок.

После этого УУК направляет каждому диагностирующему элементу информацию о том сколько и какие элементы он должен диагностировать. Элементы принимают эту информацию и производят диагностику согласно рассчитанным данным.

На этом цикл диагностики заканчивается. Встроенная система диагностирования ожидает начала нового этапа диагностирования для перерасчёта значений и перераспределении нагрузки.

#### **4.4 Оценка эффективности разработанного алгоритма**

С целью проверки корректности разработанного алгоритма было разработано ПО, симулирующее работу разработанного алгоритма и показывающее сравнение с «неразумным» алгоритмом, когда вся диагностическая нагрузка возложена на один элемент до момента его выхода из строя. Назовём такой алгоритм – «1→всех». Его суть кратко заключается в том, что один элемент назначается тестирующим, пока не выйдет из строя. Затем следующий элемент принимает на себя его обязанности и работает до выхода из строя. Данная программа была написана на языке Javascript (для кроссплатформенности), с помощью фреймворка EmberJS.

Ниже приведены результаты работы алгоритма. При вводе входных параметров программа генерирует введённое число элементов, присваивая им случайные начальные значения заряда батареи и загрузки процессора. От этой отправной точки работают сравниваемые алгоритмы. Пример сгенерированного начального состояния представлен на рис. 12.



Рисунок 12. Пример начального состояния

В левой части рисунка мы видим панель для ввода параметров системы, необходимых для работы разработанного алгоритма. Справа мы видим визуализацию сгенерированных элементов вычислительной системы по введённым параметрам. Внутри каждого элемента 2 столбца – левый отвечает за уровень заряда элемента, правый за текущую загрузку CPU. Перед каждым циклом диагностирования значение CPU заново генерируется случайно, значение мощности заряда батареи уменьшается в зависимости от числа проведенных диагностических тестов.

Результатом работы программы является график, наглядно демонстрирующий преимущества разработанного алгоритма. В качестве оцениваемой характеристики было выбрано число работоспособных элементов в каждый момент времени.

На рис. 13 представлен один из примеров расчёта. Как можно увидеть по графику, число работоспособных элементов при использовании разработанного алгоритма выше в каждый момент времени работы системы.

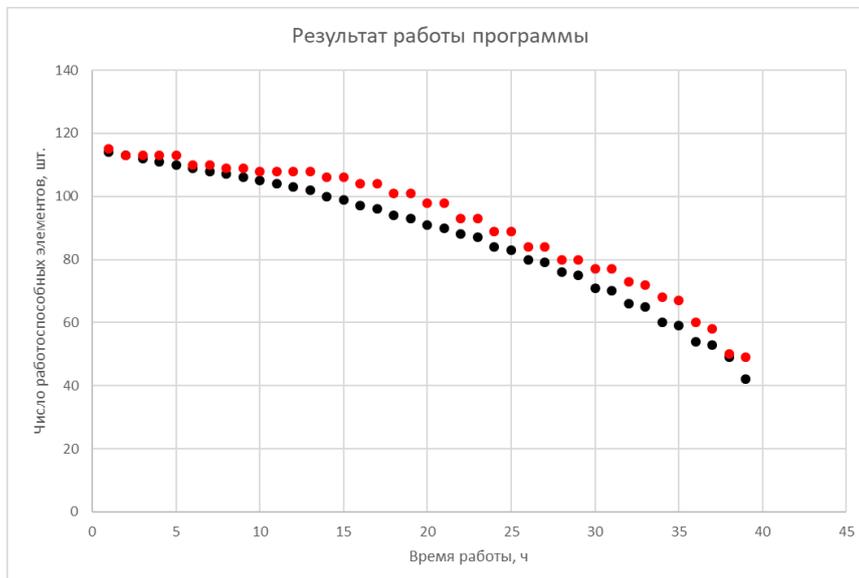


Рисунок 13. Пример работы программы. (Красным – разрабатываемый алгоритм, Чёрным – алгоритм 1→всех).

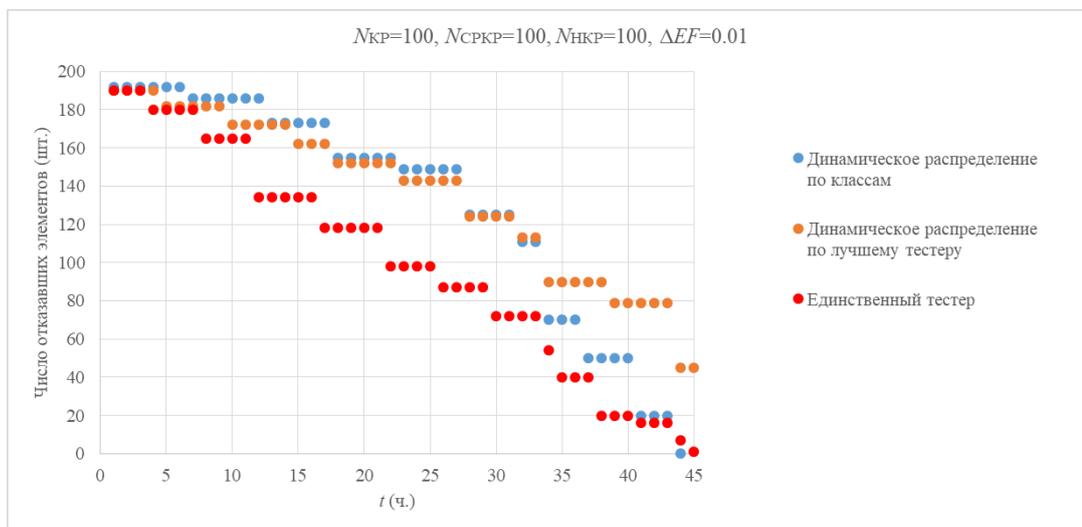
Для оценки эффективности разработанных моделей и алгоритма был разработан интегральный критерий оценки эффективности методик диагностирования. Он позволит оценить, как изменились параметры, от которых зависит коэффициент использования (1) в результате применения разработанных моделей и алгоритма. Отказом системы в данном случае будем считать ситуацию, когда число работоспособных элементов становится меньше или равно пороговому значению  $N_{\text{ПОР}}$ . Хотя система и продолжает функционировать, но возникновение такой неисправности вынуждает в скором времени производить ремонтные работы по ее восстановлению.

Число работоспособных элементов:

$$N_p(t) = \sum_{i=1}^{N_{\text{HKP}} + N_{\text{СРKP}}} N_{p_i}(t),$$

$$N_{p_i}(t) = \begin{cases} 0, & EF_i(t) \leq EF_{\text{ПОР}} \\ 1, & EF_i(t) > EF_{\text{ПОР}} \end{cases},$$

Для оценки разработанного алгоритма, был рассмотрен другой алгоритм, когда выбирается единственный тестирующий элемент, который выполняет свои функции до возникновения отказа (алгоритм единственного тестера). Также разработанный специально для данного сравнения алгоритм, тот же единственный тестер выбирается каждый раз перед началом диагностического цикла исходя из наибольшего значения показателя надежности (алгоритм лучшего тестера). Было разработано ПО, позволяющее произвести симуляцию работы методик диагностирования и оценить их при помощи интегрального показателя. Для определения показателя надежности использовалась одна невосстанавливаемая характеристика для автономных элементов вычислительной системы – мощность заряда батареи. Соответственно  $EF=Z$  и  $\Delta EF=\Delta Z$ . На каждом этапе для каждого элемента генерируется случайная нагрузка ( $\Delta EF_{РАБ}$ ). Результаты экспериментов представлены на рис. 14.



a

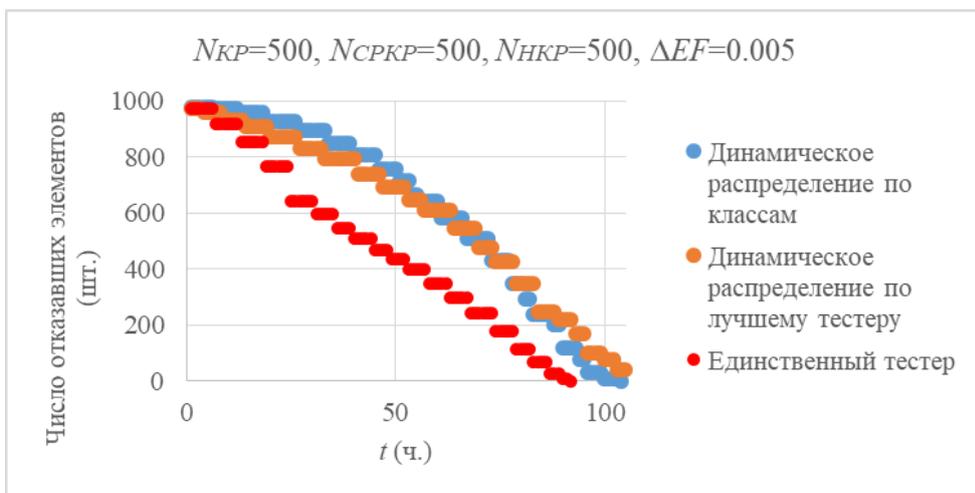
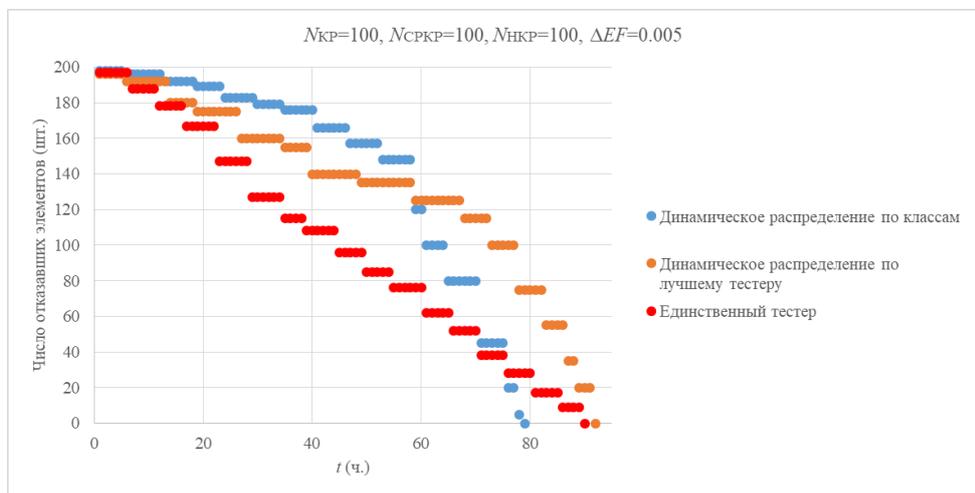
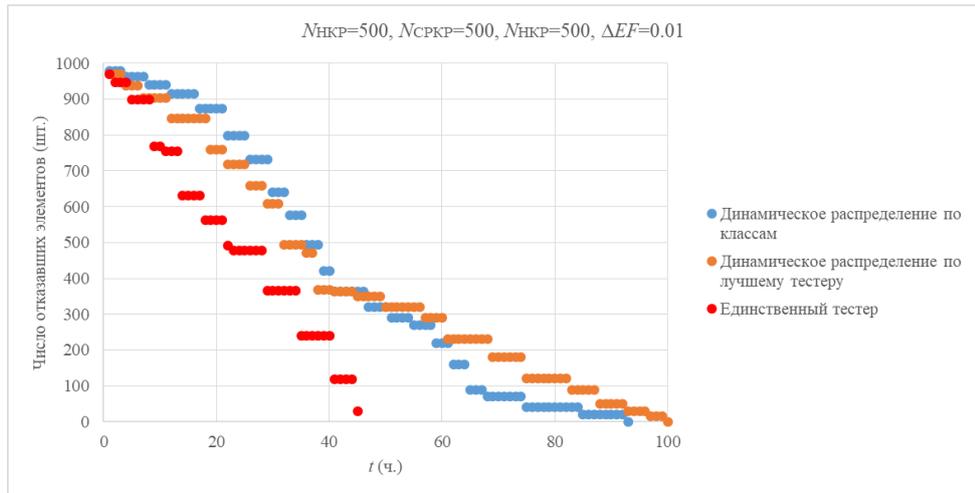


Рисунок 14. Сравнение разработанной и существующих методик.

Результаты, полученные при помощи теоретических расчётов и программного моделирования [19] не противоречат друг другу. Соответственно были сделаны следующие выводы:

#### *Масштабируемость результатов*

- Изменение числа элементов (как в большую, так и в меньшую сторону) сохранило лучшие показатели разработанного алгоритма по сравнению с существующими.
- Изменение сложности проводимых диагностических тестов не оказало отрицательного влияния на эффективность алгоритма (различные  $\Delta EF$ ).

#### *Количественные характеристики эффективности алгоритма*

- По графикам видно, что характеристика изменения числа работоспособных элементов изменилась таким образом, что на первом этапе число отказавших элементов невелико. Это позволяет увеличить время работы системы  $T_p$ , т.к. момент, когда число работоспособных элементов становится меньше порогового значения сдвигается во времени.
- Выделен этап работы алгоритма, когда скорость уменьшения числа работоспособных элементов увеличивается. Данный эффект позволяет в распределенных системах (элементы находятся на достаточном удалении друг от друга) произвести кластеризацию отказов. На начальном этапе работы алгоритма число отказов будет небольшим. В определенной временной точке, когда большая часть элементов находится в состоянии, близком к состоянию отказа, число отказов увеличивается. Соответственно, произвести ремонтно-восстановительные работы можно за один выезд всему кластеру элементов вычислительной системы. Таким образом, оказывается положительное влияние на такую характеристику коэффициента использования, как время ремонта –  $T_{РЕМ}$ .

Дополнительно проанализируем получившиеся результаты. Кажется, что в точке пересечения разработанного метода и метода «по лучшему тестеру» данный метод становится более эффективным. На самом деле, этой точке на графике соответствует абсолютно разное состояние элементов вычислительных систем. При распределении диагностической нагрузки в данной точке все элементы уже близки к пороговому значению коэффициента работоспособности, что означает - элементы близки к отказу. В свою очередь при динамическом распределении по лучшему тестеру состояние элементов может быть различным акак близким к пороговому значению, так и достаточно работоспособным. Таким образом, сравнивать алгоритмы после прохождения данной точки не имеет большого смысла.

Стоит отметить, что для систем, в которых отказом является прекращения работы малого числа элементов алгоритм динамического распределения по лучшему тестеру не подходит. Так, например, потеря пожароопасной зоны или участка дороги при использовании камер не должна остаться без внимания. Методы резервирования, безусловно, должны помогать в таких случаях, но в результате использования динамического распределения диагностической нагрузки по классам можно увеличить срок безотказной работы всех элементов и уменьшить число возникающих отказов.

Наилучшие условия для использования разработанного алгоритма – при количестве элементов превышающем 100 (каждого типа). С увеличением количества элементов мы видим увеличение количества работающих элементов. В системе со 100 элементами каждого типа среднее увеличение количества рабочих элементов составляет 6% по сравнению с алгоритмом «по лучшему тестеру» и 35% по сравнению с алгоритмом «по единственному тестеру». В системе с 500 элементами - 7% и 45% соответственно. С увеличением стоимости тестовой мощности разработанный алгоритм увеличивает количество работающих элементов на 2%, в то время как другие

алгоритмы не показывают такой же результат. Это показывает, что алгоритм хорошо работает и в системах с высоконагруженными элементами.

#### **4.5 Выводы по главе**

В данной главе был разработан алгоритм динамического распределения диагностической нагрузки. Необходимость в создании данного алгоритма была обусловлена тем, что всё более часто стали появляться распределённые системы с автономными элементами. Эти системы характеризуются тем, что их элементы находятся на достаточно большом расстоянии друг от друга и от системы управления. Такие элементы обычно объединяются в кластеры и им назначается устройство локального управления. Из-за необходимости постоянной смены локации элементов вычислительных систем по отдельности или кластеров, в которых они состоят становится невозможным подводить к ним постоянное электропитание. Такие элементы называются автономными и характеризуются тем, что питают их переносные источники питания.

Мультивендорность и гетерогенность вычислительных систем также является фактором для развития алгоритмов встроенного диагностирования. Элементы вычислительной системы разных производителей для встроенной системы диагностирования должны быть представлены как «черный ящик» в виде входных и выходных параметров, которые можно собрать, обработать и, возможно, присвоить данному элементу.

Цель разработанного алгоритма – увеличить число работоспособных элементов (значение коэффициента работоспособности должно быть больше порогового) путем обоснованного перераспределения диагностической нагрузки. Принцип обоснованности заключается в том, что если техническое состояние элемента в данный момент хуже, чем у других, то есть если он является диагностирующим, то ему должно достаться меньше проверок для проведения диагностики.

Основой для алгоритма будут данные, полученные путем расчёта по разработанным моделям из главы 2. Эти данные позволяют оценить техническое состояние элементов вычислительной системы, определить необходимое и возможное число проверок исходя из которого проводится дополнительный мониторинг состояния системы. Если число возможных или необходимых проверок уменьшилось, то это говорит о значительном снижении работоспособности конкретных элементов системы. Данный алгоритм способен прогнозировать состояние системы при следующем диагностическом цикле и предупреждать о нём. Так, если число возможных проверок будет менее чем в 2 раза больше чем необходимое, то сформируется уведомление о последнем полном диагностическом цикле.

Для моделирования работы алгоритма, а также оценки его эффективности было разработано ПО на языке Javascript (Ember.js). Программа состоит из 2 частей. Первая часть позволяет сгенерировать и визуализировать вычислительную систему по выходным параметрам. Вторая часть производит непосредственно расчёт по разработанному и сравниваемым алгоритмам и отображает их в виде графика или таблицы.

Результаты моделирования показали некоторые преимущества, которые удалось получить по сравнению с существующими алгоритмами. Так, например, удалось изменить характеристику числа работоспособных элементов таким образом, что число работоспособных элементов от начала работы системы до определенной точки выше, чем у существующих аналогов. Эти результаты позволяют предполагать, что использования алгоритма в реальной системе увеличит коэффициент технического использования.

По результатам моделирования удалось обнаружить и использовать эффект кластеризации отказов. В виду обоснованного распределения диагностической нагрузки элементы большим числом подходят к пороговому

значению коэффициента работоспособности. Это означает, что проведение регламентных или ремонтных работ может быть выполнено в рамках одного выезда, что уменьшит как время выполнения этих работ, так и их стоимость.

Таким образом, данные, полученные в результате разработки алгоритма, разработки ПО и моделирования работы алгоритма дают основания полагать, что разработанный алгоритм действительно позволяет повысить эффективность встроенной системы диагностирования, увеличивая время работоспособности элементов вычислительной системы и путем кластеризации отказов снизить время и стоимость проведения технических ремонтных работ.

## **Глава 5. Апробация и внедрение разработанных моделей, методов и алгоритма в состав программно-аппаратного комплекса встроенной системы диагностирования**

### **5.1 Внедрение результатов в программно – аппаратный комплекс «Безопасный город»**

Основные результаты диссертационной работы реализованы в составе элементов вычислительных систем и внедрены в программно-аппаратный комплекс «Безопасный город».

Система «Безопасный город» является программно-аппаратным комплексом, основной целью которого является сбор, обработка, хранение данных, поступающих с различных городских источников информации. Такими источниками являются стационарные, мобильные и передвижные комплексы дорожных камер, светофоры, информация о вызовах службы 112, системы фиксации городских парковок, данные о ДТП.

Данная система реализует программно аппаратный комплекс, который предоставляет возможность операторам в режиме реального времени следить за состоянием различных комплексов фотофиксации по г. Перми и Пермскому краю. В функционал системы входит:

- Отображение и уведомление о состоянии технических средств фото- и видеофиксации на территории г. Перми и Пермского края в режиме реального времени;
- Реализация мониторинга траффика дорожного движения;
- Аналитика траффика дорожного движения (выявление отклонения показателей от нормативных);
- Аналитика мест положения и маршрутов транспортных средств;

- Интеграция различных поставщиков информации (передвижные комплексы, парковки);
- Отображение оперативной информации на карте;
- Агрегация вызовов в службу 112;
- Информация об ограничениях дорожного движения и ДТП;
- Выявление мест концентрации ДТП для перекрестков и перегонов.

Главный экран интерфейса системы представлен на рис. 15.

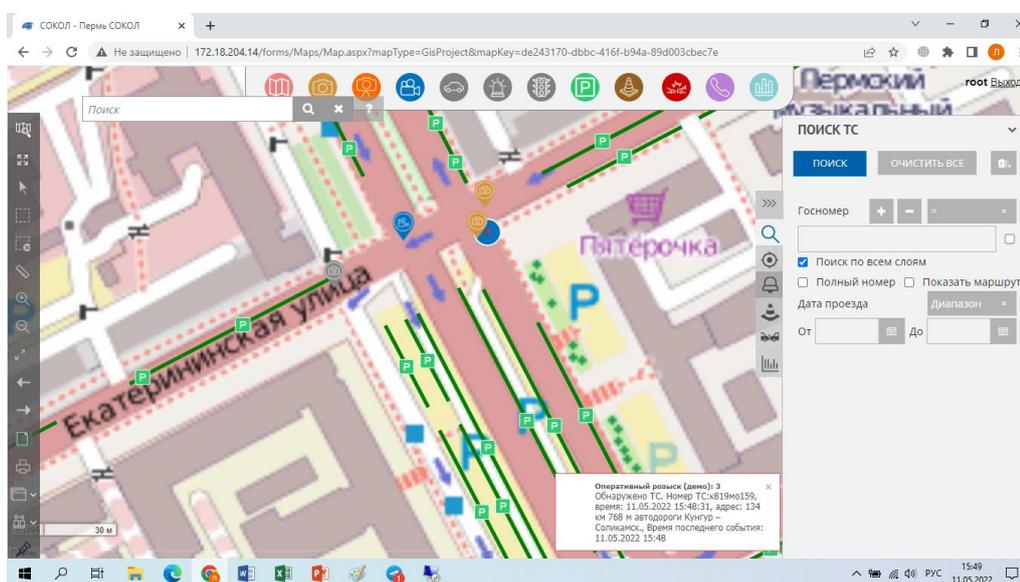


Рисунок 15. Главный экран интерфейса центра обработки и отображения данных системы «Безопасный город»

Разработанные в данной работе модели, методы и алгоритм были внедрены в подсистему ВС «Безопасный город», отвечающую за сбор, обработку, хранение данных, поступающих со стационарных, мобильных и передвижных комплексов фиксации нарушений правил дорожного движения. Данная подсистема отвечает и за мониторинг и диагностику самих комплексов фиксации в т. ч. обрабатывает и представляет оперативные данные об их техническом состоянии операторам системы «Безопасный город». Схема подсистемы представлена на рис. 16.

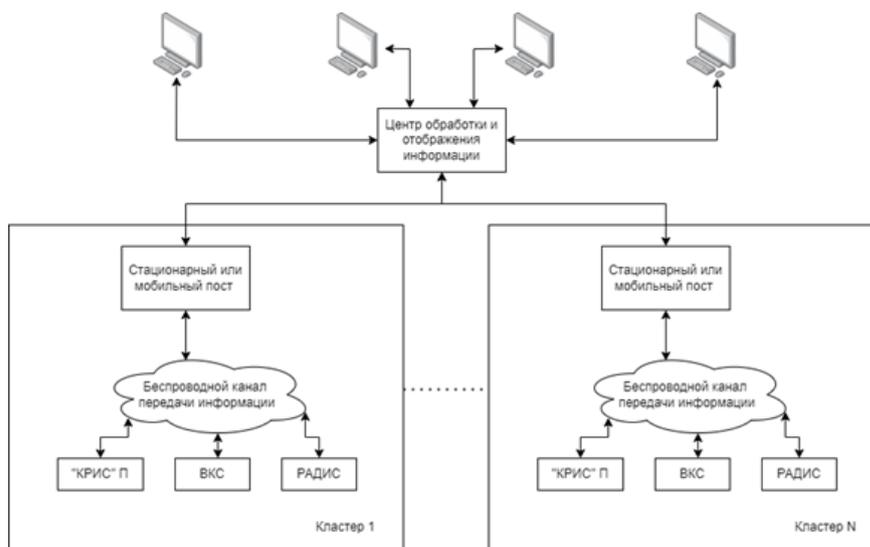


Рисунок 16. Схема подсистемы мобильных и передвижных комплексов фиксации нарушений ПДД «Безопасный город»

Основными элементами вычислительной системы в данном случае являются комплексы КРИС-П, ВКС (СКАТ) и РАДИС. КРИС-П - фоторадарный передвижной комплекс является оперативно-техническим средством автоматического контроля скоростного режима и предназначен для фото- и видеофиксации нарушений ПДД. Комплекс устанавливается сбоку от контролируемого участка дороги. Имеется возможность передачи данных и кадров на Мобильный пост по радиоканалу и/или в Центр обработки данных (ЦОД\*) с помощью флеш-накопителя. Фоторадарный датчик оборудован программно-аппаратными средствами для автоматического распознавания государственных регистрационных знаков (ГРЗ) ТС и проверки их по различным федеральным и региональным базам. СКАТ-П - предусматривает установку оборудования на обочине контролируемого участка дороги. Передвижной комплекс устанавливается на треногу и подключается к автономному источнику питания, который обеспечивает до 24 часов работы комплекса. Настройка комплекса производится с помощью портативного компьютера. Информация передается дистанционно или предоставляется в центр обработки данных на защищенном внешнем носителе.



Рисунок 17. Мобильные комплексы фото- и видеофиксации *a* - СКАТ – П,  
*б* – КРИС-П.

В данной подсистеме было реализовано встроенное диагностирование внутри кластера по алгоритму единственного тестера. Определена некоторая очередность выполнения функции тестера для каждого элемента. При этом тестером в рамках очереди может стать устройство, текущие технические характеристики которого не являются лучшими для выполнения диагностических операций. Это приводило к ухудшению эксплуатационно-технических характеристик системы.

С целью повышения эффективности проведения процедур диагностики, а также улучшения эксплуатационно-технических показателей вычислительной системы разработанные модели, методы и алгоритм были адаптированы и внедрены в состав программного обеспечения самих комплексов фиксации, постов, отвечающих за управление кластерами комплексов фотофиксации, а

также в программную реализацию системы, агрегирующей и отображающей полученные данные на пультах операторов системы.

Для комплексов фотофиксации были разработана и внедрена модель, а также были добавлены обработчики, позволяющие получать данные о проведении необходимого числа диагностических проверок и тестируемых устройствах от внешнего контроллера. В программном комплексе поста был реализован функционал получения и расчёта данных согласно модели системы встроенного диагностирования, а также реализован алгоритм перераспределения диагностической нагрузки между элементами вычислительной системы.

На рис. 18 представлены характеристики изменения числа работоспособных элементов до и после внедрения результатов диссертационной работы.

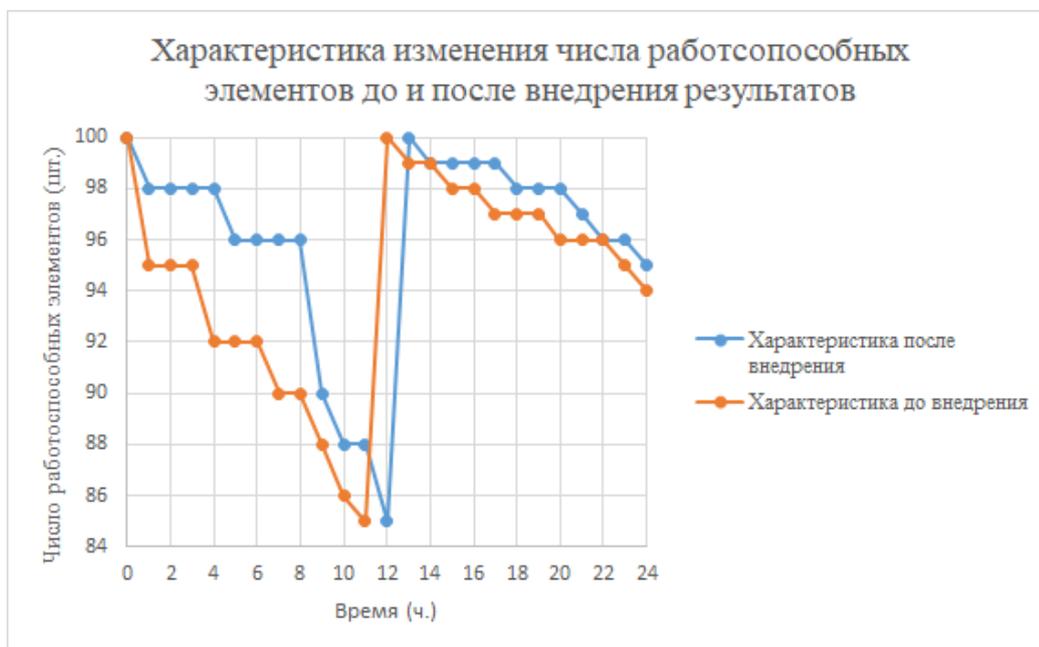


Рисунок 18. Характеристики изменения числа работоспособных элементов до и после внедрения результатов диссертационной работы.

Исследуя графики можно сделать несколько выводов:

- Разработанные модели, методы и алгоритм позволили увеличить время нахождения системы в работоспособном состоянии в наблюдаемом периоде с 11 до 12 часов (8,3%), что позволяет увеличить коэффициент технического использования системы.
- Резкие изменения числа работоспособных элементов в интервале 0-1ч и 8-9ч (соответственно 08:00-09:00 и 17:00-18:00) объясняется увеличением дорожного траффика, а также увеличением числа регистрируемых фотофактов, что увеличивает рабочую нагрузку на элементы системы.

Результаты, полученные в рамках моделирования, и практические результаты, полученные в реальной системе, не противоречат друг другу. Для доказательства рассмотрим данные, полученные по результатам моделирования и данные реальной системы и оценим их по критерию Пирсона (хи-квадрат).

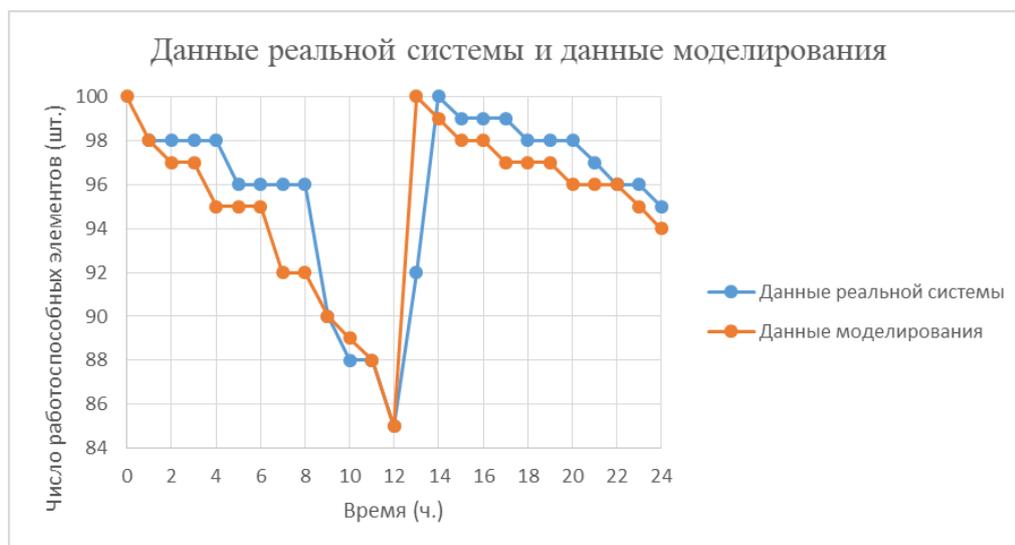


Рисунок 19. Сравнение результатов моделирования и данных реальной системы.

Критерий Пирсона, или критерий  $\chi^2$  (Хи-квадрат) применяют для проверки гипотезы о соответствии эмпирического распределения предполагаемому теоретическому распределению  $F(x)$  при большом объеме выборки ( $n \geq 100$ ). Критерий применим для любых видов функции  $F(x)$ , даже при неизвестных значениях их параметров, что обычно имеет место при

анализе результатов механических испытаний. В этом заключается его универсальность.

Использование критерия  $\chi^2$  предусматривает разбиение размаха варьирования выборки на интервалы и определения числа наблюдений (частоты) для каждого из интервалов. Для удобства оценок параметров распределения интервалы выбирают одинаковой длины. Число интервалов зависит от объема выборки.

Недостатком критерия согласия Пирсона является потеря части первоначальной информации, связанная с необходимостью группировки результатов наблюдений в интервалы и объединения отдельных интервалов с малым числом наблюдений. В связи с этим рекомендуется дополнять проверку соответствия распределений по критерию  $\chi^2$  другими критериями. Особенно это необходимо при сравнительно малом объеме выборки ( $n \approx 100$ ).

Согласно критерию Пирсона, 2 данные характеристики со степенью достоверности 0.95 согласуются с вероятностью 0.97.

Внедрение результатов диссертационной работы позволило увеличить коэффициент технического использования системы, путем изменения поведения характеристики числа работоспособных элементов. Система уведомлений, заложенная в алгоритме, позволила в некоторых случаях предотвратить возникновение отказов путём своевременного проведения ремонтных и регламентных работ.

## **5.2 Выводы по главе**

В 5-й главе диссертации представлены результаты апробации и внедрения полученных результатов научной работы в реальную вычислительную систему. Для внедрения была выбрана вычислительная система «Безопасный город». Основной функционал данной системы состоит в

управлении, мониторинге, диагностике, а также агрегации полученной информации от устройств находящихся на дорогах г. Перми и Пермского края и обеспечивающих безопасность дорожного движения. В этой системе присутствуют различные аналитические модули, позволяющие анализировать полученные данные и предоставлять операторам или другим лицам информацию касательно местоположения или маршрутов транспортных средств, текущую дорожную обстановку, а также производить аналитику архивных данных, на основе которых принимаются решения об изменении участков дорожной сети, дорожных условий и т.п.

Внедрение производилось в подсистему, отвечающую за работу со стационарными мобильными и передвижными устройствами фото- и видеофиксации. Данная подсистема была выбрана по нескольким причинам, а именно: наличие встроенной системы диагностирования, большое количество автономных элементов (передвижные и мобильные комплексы фиксации используют переносные аккумуляторы в качестве источников питания), разделение на кластеры согласно геопозиции, наличие устройства локального управления в каждом кластере а также наличие резервирования областей наблюдения. Последний пункт важен, так как позволяет определить некоторый ненулевой порог неработоспособных элементов вычислительной системы. Данный параметр позволит произвести более длительные опыты по внедрению разработанных моделей, методов и алгоритма диагностирования и оценить возникающие положительные эффекты.

Данная подсистема состоит из нескольких уровней, каждый из которых потребовал доработок. На первом уровне находятся сами элементы вычислительной системы (ПиПР). Ими являются программно - аппаратные комплексы КРИС-П, ВКС (СКАТ-П), РАДИС и некоторые другие. В данном случае был доработан программный комплекс этих устройств в части общения

с мобильным постом и возможности принятия конфигурации встроенного диагностирования и работы именно по ней.

Был доработан программный комплекс мобильного поста – устройства локального управления кластером первичных преобразователей. В рамках этих доработок был реализован сбор и расчёт показателей разработанных моделей, соответственно расчёт и распределение числа проверок между агентами диагностирования, встроенных в первичные преобразователи. Также доработан модуль оповещений, отвечающий за генерацию и отправку в ЦОиОД сообщений мониторинга.

В систему обработки и отображения потребовалось внести некоторые изменения. Так, были реализованы оперативные уведомления получаемые в реальном времени операторами о текущем или возможном будущем статусе работоспособности элементов вычислительной системы и всей системы в целом, а также отображение их на карте для определения оптимального маршрута проведения ремонтно-восстановительных работ.

Наблюдения за изменениями после внедрения полученных результатов позволило определить численные показатели эффективности, а именно:

- Увеличение периода работоспособности вычислительной системы в среднем на 8,3% (примерно 1 час) (Акт внедрения ГК «ИВС»).
- Кластеризация отказов позволила строить более оптимальные маршруты проведения ремонтно-восстановительных работ, за счёт чего уменьшено время восстановления системы на 5% (примерно 1,5ч) (Акт внедрения ООО «ПроИнфоСервис»).
- Результаты, полученные в рамках научной работы, были внедрены в учебный процесс кафедры АТ ФГАОУ ВО «Пермский Национальный Исследовательский Политехнический Университет».

При сравнении результатов до и после внедрения а также результатов моделирования и реальной характеристики, при помощи математического аппарата теории вероятности было доказано, что полученные результаты с достоверностью 0,95 сходятся с вероятностью 0,97. Это подтверждает правильность теоретических расчётов.

Таким образом, результаты, полученные в диссертации нашли непосредственное практическое применение в важной системе, обеспечивающей сохранность безопасности, здоровья и жизни граждан Пермского края.

## Заключение

Представленная диссертационная работа посвящена решению важной научно-технической проблемы – улучшение эксплуатационно-технических показателей вычислительных систем и их элементов на основе эффективного применения системы встроенного диагностирования. В диссертационной работе поставлены и решены следующие задачи исследования.

1. Разработаны математические *модели встроенной системы тестового диагностирования и элемента вычислительной системы*. Использование данных моделей позволило представить элемент вычислительной системы в виде черного ящика с набором характеристик. Это, в свою очередь, позволит использовать разработанную модель встроенной системы диагностирования с достаточной степенью точности описания элементов в любой области применения вычислительных систем.

Разработанные модели обеспечивают унифицированный подход к описанию элементов вычислительной системы позволяют объединять элементы в кластеры, разделять по уровням важности. Это предоставляет дополнительные рычаги настройки встроенной системы диагностирования для удовлетворения требований по надежности вычислительной системы на этапе эксплуатации.

Математические величины, описывающие разработанные модели заданы выражениями, позволяющие достаточно легко автоматизировать их расчёт на различных ресурсных мощностях.

Показатели рассчитанных моделей позволили реализовать логику алгоритма динамического перераспределения диагностической нагрузки, что в последствии повысило коэффициент технического использования вычислительной системы.

2. Для расчёта основных показателей разработанных моделей были разработаны два новых метода – метод принятия решений в задачах диагностики, позволяющий определить весовые коэффициенты учитываемых характеристик работоспособности и метод расчёта показателя надёжности элемента вычислительной системы. Первый метод может быть использован при решении различных задач, при правильном определении степени участия каждой характеристики в том показателе, который необходимо повысить. В данной работе метод представляет достаточно удобный инструмент для определения важности учитываемых характеристик работоспособности, который теоретически может быть автоматизирован. Вторым методом также может быть использован при решении различного круга задач, в которых необходимо определять текущее состояние работоспособности элемента вычислительной системы, например, в задачах мониторинга. Проведённый анализ расчётов по разработанному методу показал, что он даёт более правильные оценки в ситуациях, возникающих при распределении диагностической нагрузки.

Применение разработанных методов позволило более точно определить важность критериев и показатели надёжности элементов вычислительных систем. Это, в свою очередь, позволило точно определить техническое состояние элементов вычислительных систем, вовремя исключить их из разряда диагностируемых, чем повысить время работоспособности системы, а затем и коэффициент технического использования.

3. Разработан и апробирован алгоритм распределения диагностической нагрузки между элементами встроенной системы тестового диагностирования. Использование данного алгоритма позволяет не только эффективно распределять диагностическую нагрузку между элементами вычислительной системы, что было показано на примере моделирования при помощи ПО и результатами внедрения в реальную систему, но и с помощью механизма

уведомлений и прогнозирования влиять на такую характеристику надежности вычислительной системы как время восстановления.

Проверка сходимости теоретических и практических результатов подтвердила предположения о правильности теоретических расчётов и возможном положительном эффекте при использовании разработанного алгоритма.

Практическая значимость диссертационного исследования заключается в улучшении качественных и эксплуатационных характеристик вычислительной системы и её элементов, что подтверждается актами внедрения. Полученные результаты исследования использованы в конкретной системе диагностирования, а также могут быть использованы для реализации средств диагностирования вычислительных систем широкого назначения.

Внедрение полученных результатов позволило повысить такой показатель надежности вычислительной системы как коэффициент технического использования. Эта цель была достигнута благодаря повышению времени работоспособности вычислительной системы и уменьшению времени ремонта (восстановления). Первый параметр уменьшился благодаря динамическому и обоснованному перераспределению диагностической нагрузки, вследствие чего изменилась характеристика изменения числа работоспособных элементов вычислительной системы. Данное изменение позволило увеличить число работоспособных элементов в течение определенного интервала времени. Второй параметр изменился также благодаря изменениям в характеристике изменения числа работоспособных элементов. Динамическое распределение произвело эффект выравнивания нагрузки и, в следствии, этого - выравнивание показателя работоспособности элементов вычислительной системы. Благодаря чему у большого числа элементов одновременно наблюдается приближение показателя

работоспособности к пороговому значению. Это позволяет строить более оптимальные маршруты ремонтно-восстановительных работ, а также оптимизировать их наполнение.

Элементы мониторинга и прогнозирования, введенные в алгоритм динамического распределения диагностической нагрузки позволили предотвратить появление некоторого числа отказов, что повышает время работоспособности вычислительной системы, а значит и коэффициента технического использования.

## Список использованных источников

1. Абакумов, М.И. Обзор современных средств заряда и разряда аккумуляторных батарей / М.И. Абакумов, А.В. Савченко // Актуальные проблемы гуманитарных и естественных наук. – 2015. – №7-4. – С. 31-33.
2. Богданов, Ю.Ю. О двух диагностических моделях распределенных цифровых систем / Ю.Ю. Богданов // Автоматика и телемеханика. – 1986. – №8. – С. 127-134.
3. Бойко, А.А. Метод оценки весовых коэффициентов элементов организационно-технических систем / А.А. Бойко, И.С. Дягтерёв // Системы управления, связи и безопасности. – 2018. – №2. – С. 245 – 266.
4. Ведешенков, В.А. Об организации самодиагностируемых цифровых систем / В.А. Ведешенков // Автоматика и телемеханика. – 1983. – №7. – С. 133-144.
5. Ведешенков, В.А. О двух методах дешифрации результатов диагностирования цифровых систем / В.А. Ведешенков, А.М. Нестеров // Электронное моделирование. – 1981. – №2. – С. 53-58
6. Ведешенков, В.А. Самодиагностирование цифровых систем с реконфигурацией / В.А. Ведешенков // Проблемы управления. – 2003. – №4. – С. 39–51.
7. Горелов, О.И. Анализ степени диагностируемости открытых вычислительных сетей / О.И. Горелов, Е.В. Плотников // Автоматика и Вычислительная Техника. – 1993. – №6. – С. 61-66.
8. Горелов, О.И. Представление открытых вычислительных сетей моделью самодиагностируемой системы с распределенным диагностическим ядром / О.И. Горелов, Е.В. Плотников // Автоматика и Вычислительная Техника. – 1983. – №4. – С. 64-70.
9. ГОСТ 20911-89. Техническая диагностика. Термины и определения. – М.: Издательство стандартов, 1989. – 13 с.

10. ГОСТ 27.002-2015. Надежность в технике. Основные понятия. Термины и определения. – М.: Издательство стандартов, 2015. – 28 с.
11. Гуров, С.В. Надежность восстанавливаемых резервированных систем с последствием отказов / С.В. Гуров, Л.В. Уткин // Автоматика и телемеханика. – 2017. – №1. – С. 137-151.
12. Дианов, В.Н. Диагностика и надежность автоматических систем / В.Н. Дианов. – М.: МГИУ, 2004. – 160 с.
13. Дмитриев, Ю.К. Диагностируемость вычислительных систем с несимметричными оценками / Ю.К. Дмитриев // Автоматика и телемеханика. – 1985. – №12. – С. 106-112.
14. Жвеляя, Л.Р. Характеристики информационных систем управления / Л.Р. Жвеляя // Международный студенческий научный вестник. – 2015. – № 3-3. – С. 364-365.
15. Журков, А.П. Анализ возможностей применения подходов самодиагностирования к распределенной радиотехнической системе наблюдения / А.П. Журков, Д.А. Аминев, П.А. Гусева, С.С. Мирошниченко, П.А. Петросян // Системы управления, связи и безопасности. – 2015. – №4. – С. 114-121.
16. Зрелова, Т.И. Самодиагностирование цифровых систем, содержащих блоки с самопроверяемыми схемами встроенного контроля / Т.И. Зрелова // Автоматика и телемеханика. – 1984. – № 2. – С. 123-133.
17. Каравай, М.Ф. Построение теста для поиска кратных неисправностей комбинационных устройств произвольного базиса / М.Ф. Каравай // Автоматика и телемеханика. – 1973. – №4. – С.164-179.
18. Клейман, Л. А. Система управления точками Wi-fi-доступа / Л.А. Клейман, В.И. Фрейман // Энергетика. Инновационные направления в энергетике. CALS-технологии в энергетике. – 2017. – № 1. – С. 39-45.

19. Клейман, Л.А. Метод моделирования дискретных событий в качестве инструмента диагностики элементов информационно-управляющих систем / Л.А. Клейман, В.И. Фрейман // Автоматизированные системы управления и информационные технологии: Материалы всероссийской научно-технической конференции: в двух томах, Пермь, 30–31 мая 2019 года. – Пермь: Пермский национальный исследовательский политехнический университет. – 2019. – С. 51-56.

20. Клейман, Л.А. Методика повышения надежности элементов информационно-управляющих систем с применением встроенных средств диагностирования / Л.А. Клейман, В.И. Фрейман // Энергетика. Инновационные направления в энергетике. CALS-технологии в энергетике. – 2020. – Т. 1. – С. 63-71.

21. Клейман, Л.А. Методика принятия решений в задаче диагностики элементов информационно-управляющих систем / Л.А. Клейман // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2021. – № 38. – С. 90-109.

22. Клейман, Л.А. Мониторинг и диагностика элементов беспроводных информационно-управляющих систем / Л.А. Клейман, В.И. Фрейман, А.А. Южаков // Вестник Поволжского государственного технологического университета. Серия: Радиотехнические и инфокоммуникационные системы. – 2018. – № 4(40). – С. 58-73.

23. Клейман, Л.А. Повышение надежности устройств беспроводных систем управления на основе метода анализа тепловых карт / Л.А. Клейман, В.И. Фрейман // XIII Всероссийское совещание по проблемам управления ВСПУ-2019: Сборник трудов XIII Всероссийского совещания по проблемам управления ВСПУ-2019, Москва, 17–20 июня 2019 года / Институт проблем

управления им. В.А. Трапезникова РАН. – Москва: Институт проблем управления им. В.А. Трапезникова РАН. – 2019. – С. 2866-2873.

24. Клейман, Л.А. Повышение надёжности функционирования элементов информационно-управляющих систем с применением встроенных средств диагностирования / Л.А. Клейман, Е.Л. Кон, В.И. Фрейман, А.А. Южаков // Вестник Поволжского государственного технологического университета. Серия: Радиотехнические и инфокоммуникационные системы. – 2019. – № 3(43). – С. 29-40.

25. Клейман, Л.А. Реконфигурирование встроенной системы тестового диагностирования элементов информационно-управляющих систем / Л.А. Клейман, В.И. Фрейман // Инновационные технологии: теория, инструменты, практика. – 2019. – Т. 1. – С. 13-18.

26. Клейман, Л.А. Система мониторинга и диагностики коммуникационных элементов беспроводных систем управления / Л.А. Клейман, В.И. Фрейман // Инновационные технологии: теория, инструменты, практика. – 2018. – Т. 1. – С. 384-389.

27. Клейман, Л.А. Система управления и мониторинга точек Wi-fi доступа / Л. А. Клейман, В. И. Фрейман // Автоматизированные системы управления и информационные технологии: Материалы всероссийской научно-технической конференции. В 2-х томах, Пермь, 17 мая 2018 года. – Пермь: Пермский национальный исследовательский политехнический университет. – 2018. – С. 256-260.

28. Климанова, Е.Ю. Оценка производительности вычислительных систем / Е.Ю. Климанова, А.Р. Субханкулова, Б.В. Зеленко, О.Ю. Леонтьева // Вестник Казанского технологического университета. – 2015. – №24. – С. 102-105.

29. Литиков И.П. Тестово-функциональное диагностирование цифровых устройств и систем / И.П. Литиков, Е.С. Согомоян // Автоматика и телемеханика. – 1985. – №3. – С. 111-121.

30. Микеладзе, М.А. Развитие основных моделей самодиагностирования сложных технических систем / М.А. Микеладзе // Автоматика и телемеханика. – 1995. – №5. – С. 3-18.

31. Михайлов, В.С. Оценка вероятности безотказной работы по результатам испытаний, не давших отказы / В.С. Михайлов // Надежность и качество сложных систем. – 2017. – №2 (18). – С. 56-60.

32. Новиков, Д.А. Кибернетика: Навигатор. История кибернетики, современное состояние, перспективы развития / Д.А. Новиков. – М.: ЛЕНАНД, 2016. – 160с.

33. Панкин, А.М. Алгоритмы систем диагностирования новых энергетических объектов / А.М. Панкин, Н.В. Коровкин // Научно-технические ведомости СПбГПУ. – 2016. – №4 (254). – С. 98-105.

34. Пархоменко, П.П. Техническая диагностика / П.П. Пархоменко. – М.: Наука, 1972. – 368 с.

35. Подиновский, В.В. Введение в теорию важности критериев в многокритериальных задачах принятия решений / В.В. Подиновский. – М.: Физматлит, 2007. – 64 с.

36. Половко, А.М. Основы теории надежности / А.М. Половко, С.В. Гуров. – ВHV. – 2008. – 704 с.

37. Рупосов, В.Л. Методы определения количества экспертов / В.Л. Рупосов // Вестник ИрГТУ. – 2015. – №3. – С. 1-7.

38. Сагунов, В.И. Контролепригодность структурно связанных систем / В.И. Сагунов, Л.С. Ломакина. – М.: Энергоатомиздат. – 1990. – 112 с.

39. Ситников, А.В. Аккумуляторные батареи носимых электронных устройств / А.В Ситников, С.И. Масленникова // Радиостроение. – 2017. – №5. – С. 52-72.

40. Федоров, И.И. Модель самодиагностирования для распределенных отказоустойчивых систем с деградацией структуры / И.И. Федоров // Автоматика и телемеханика. – 1990. – №1. – с. 136-144.

41. Фрейман, В.И. К вопросу о проектировании и реализации элементов и устройств распределенных информационно-управляющих систем / В.И. Фрейман // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. – 2019. – №30. – С. 28-49.

42. Хлебенских, Л.В. Автоматизация производства в современном мире / Л.В. Хлебенских, М.А. Зубкова, Т.Ю. Саукова // Молодой ученый. – 2017. – № 16 (150). – С. 308-311.

43. Шишмарев, В.Ю. Надежность технических систем / В.Ю. Шишмарев. – М.: Академия, 2010. – 304 с.

44. Шурпаев, Ю.М. Самодиагностирование распределенной системы вычислительных элементов с шинной топологией методом декомпозиции / Ю.М. Шурпаев // Автоматика и телемеханика. – 1995. – №12. – С. 144-151.

45. Balakrishnan, M. An Analysis of a Reliability Model for Repairable Fault-Tolerant Systems / M. Balakrishnan, C.S Raghavendra // IEEE Transactions on Computers. 1993. T. 42. No. 3. P. 327-339.

46. Barbazza, A. Consensus Modeling in Multiple Criteria Multi-expert Real Options-Based Valuation of Patents / A. Barbazza, M. Collan, M. Fedrizzi // Intelligent Systems. Advances in Intelligent Systems and Computing. 2014. Vol. 322. P. 269-278.

47. Barsi, F. A Theory of Diagnosability of Digital Systems / F. Barsi, F. Grandoni, P. Maestrini // IEEE Transactions on Computers. 1976. Vol. 25. No.6. P. 585-593.

48. Baxter, G. Socio-technical systems: From design methods to systems engineering / G. Baxter, I. Sommerville // *Interacting with Computers*. 2011. Vol. 23. No. 1. P. 4-17.
49. Blough, D.M. The Broadcast Comparison Model for On-Line Fault Diagnosis in Multicomputer Systems: Theory and Implementation / D.M. Blough, H.W Brown // *IEEE Transactions on Computers*. Vol. 48. No. 5. P. 470-493.
50. Blough, D.M. Complexity of Fault Diagnosis in Comparison Models / D.M. Blough, A. Pelc // *IEEE Transactions on Computers*. Vol. 41. No.3. P. 318-324.
51. Chadeev, V. M. Automation of Autonomous Largescale Production Systems / V. M. Chadeev, N. I. Aristova // *Management of large – scale system development (MLSD): 12th International Conference, Moscow, 01–03 October 2019: proceedings.* – Los Alamitos: IEEE. 2019. P. 1–4.
52. Chien, C. Design and Analysis of Adaptive Iterative Learning Control for Iteration-varying Nonlinear Systems / C. Chien, Y. Wang, F. Lian // *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi*. 2018. P. 469-474.
53. Chu, S.-C. t-TDA-Diagnosable Systems / S.-C. Chu, J.R. Armstrong // *IEEE Transactions on Computers*. 1989. Vol. 38. No. 6. P. 914-920.
54. Chwa, K.Y. On Fault Identification in Diagnosable Systems / K.Y. Chwa, S.L. Hakimi // *IEEE Transactions on Computers*. 1981. Vol. 30. No.6. P.414-422.
55. Curie, D.H. Analysis on Web Frameworks / D.H. Curie, J. Jaison, J. Yadav, J.R. Fiona // *Journal of Physics: Conference Series*. 2019. No. 1362. P. 1-6.
56. De, N. A New Method to test System Diagnosability / N. De, A. Sengupta, S. Dyopadhyay, P.K. Strimany // *Information Sciences*. 1980. Vol. 22. No. 2. P.131-138.
57. Djuraev, R.X. Analysis Of The Relationship Between The Indicators Of Controllability And Reliability Characteristics Of Data Transmission Systems / R.X. Djuraev, S.Y. Djabbarov, T.Q. Toshtemirov // *Information Science and*

Communications Technologies (ICISCT) : International Conference, Tashkent, 04–06 November 2019: proceedings. – Los Alamitos: IEEE. 2019. P. 1–4.

58. Dowdeswell, B. Finding faults: A scoping study of fault diagnostics for Industrial Cyber–Physical Systems / B. Dowdeswell, R. Sinha, S.G. MacDonell // *Journal of Systems and Software*. 2020. Vol. 168. P. 1-16.

59. Dragana, C. An approach for weighted average consensus in event detection / C. Dragana, G. Stamatescu, V. Mihai, D. Popescu // *25th Mediterranean Conference on Control and Automation (MED)*. 2017. P. 1100-1105.

60. Edwards, W. Smarts and Smarter: Improved Simple Methods for Multiattribute Utility Measurement / W. Edwards, F.H. Barron // *Organizational Behavior and Human Decision Processes*. 1994. Vol. 60. No. 3. P. 306-325.

61. Fatullah, M. A. Analysis of Discharge Rate and Ambient Temperature Effects on Lead Acid Battery Capacity / M. A. Fatullah, A. Rahardjo, F. Husnayain // *Innovative Research and Development (ICIRD) : 2nd IEEE International Conference, Jakarta, 28–29 June 2019 : proceedings*. – Los Alamitos: IEEE. 2019. P. 1-5.

62. Fei, L. An ELECTRE-Based Multiple Criteria Decision Making Method for Supplier Selection Using Dempster-Shafer Theory / L. Fei, J. Xia, Y. Feng, L. Liu // *IEEE Access*. 2019. Vol. 7. P. 84701-84716.

63. Friedman, A. D. System-Level Fault Diagnosis / A.D. Friedman, L. Simoncini // *Computer*. 1980. Vol.13. No.3. P.47-53.

64. Fujiwara, H. Connection Assignments for Probabilistically Diagnosable Systems / H. Fujiwara, K. Kinoshita // *IEEE Transactions on Computers*. 1978. Vol. 27. No. 3. P. 280-283.

65. Fujiwara, H. Some Existence theorems for probabilistically Diagnosable systems / H. Fujiwara, K. Kinoshita // *IEEE Transactions on Computers*. 1978. Vol. 27. No. 4. P.379-384.

66. Gao, Y. Autonomous Wi-Fi Relay Placement With Mobile Robots / Y. Gao, H. Chen, Y. Li, C. Lyu, Y. Liu // IEEE/ASME Transactions on Mechatronics. 2017. Vol. 22, No. 6. P. 2532-2542.

67. Gordievsky, E. Development of Mobile Power Complex Model on Renewable Energy Sources for Autonomous Electrical Supply of Russian Far Eastern Region / E. Gordievsky, E. Sirotkin, A. Miroshnichenko // Electrical Power Engineering (UralCon) : International Ural Conference, Chelyabinsk, 01–03 October 2019: proceedings. – Los Alamitos: IEEE. 2019. P. 148–153.

68. Günal, M.M. Discrete event simulation for performance modelling in health care: a review of the literature / M.M. Günal, M. Pidd // Journal of Simulation. No. 4 (2010). P. 42-51.

69. Harmat, L. A New Model For Self-Testing and Self-Diagnosing Multi-Microprocessor Systems / L. Harmat // Proc. Int. Symp. On Fault-Tolerant Computing (FTCS-11). 1981. P. 170-172.

70. Hiramoto, Y. A Built-In Self-Diagnostic Mechanism for Delay Faults Based on Self-Generation of Expected Signatures / Y. Hiramoto, S. Ohtake, H. Takahashi // Asian Test Symposium (ATS): 28th IEEE, Kolkata, 10-13 December 2019: proceedings. – Los Alamitos: IEEE. 2019. P. 31-36.

71. Houankpo, H.G. Mathematical Model for Reliability Analysis of a Heterogeneous Redundant Data Transmission System / H.G. Houankpo, Houankpo, D.V. Kozyrev, E. Nibasumba // Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT): The 12th International Congress, Brno, 01–03 October 2020: proceedings. – Los Alamitos: IEEE. 2020. P. 189–194.

72. Indrajit, R. E. Information Technology Strategic Plan development methodology: Governing from the perspectives of enterprise architecture / R.R. Indrajit // 2017 Second International Conference on Informatics and Computing (ICIC). 2017. P. 1-4.

73. Karimireddy, T. Optimization of Real-Time Transmission Reliability on Wireless Industrial Automation Networks / T. Karimireddy, S. Zhang // Automation and Computing (ICAC): 24th International Conference, Newcastle upon Tyne, 06-07 September 2018: proceedings. – Los Alamitos: IEEE. 2018. P. 1-6.

74. Karunanithi, S. Analysis of digital systems using a new measure of system of system diagnosis / S. Karunanithi S., A.D. Friedman // IEEE Transactions on Computers. 1979. Vol. 28. P.121-133.

75. Kavalero, M.V. Adaptive Q-routing with Respect to Energy Consumption Model / M.V. Kavalero, L.A. Kleiman, V.I. Freyman // Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering Conference (ElConRus). P. 22-27.

76. Kim, B. Visualizing set concordance with permutation matrices and fan diagrams / B. Kim, B. Lee, J. Seo // Interacting with Computers. 2007. Vol. 19. No. 5-6. P. 630-643.

77. Kim, W. Development, implementation, and evaluation of a fault detection and diagnostics system based on integrated virtual sensors and fault impact models / W. Kim, J.E. Braun // Energy and Buildings. 2020. Vol. 228. P. 1-27.

78. Kime, C.R. An Analysis Model for Digital Systems Diagnosis / C.R. Kime // IEEE Transactions on Computers. 1970. Vol. 19. No.11. P. 1063-1073.

79. Kleiman, L. A. The Method of Dynamic Distribution of the Diagnostic Load between Information and Control Systems Elements / L.A. Kleiman, V.I. Freyman // Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering Conference (ElConRus). 2021. P. 952-955.

80. Kleiman, L.A. Improving the functioning reliability of the information management system elements, using built-in diagnostic tools / L.A. Kleiman, V.I. Freyman // Radio Electronics, Computer Science, Control. 2021. №1. P. 158–171.

81. Kleiman, L.A. Mathematical Model for Determining the Reliability Indicators of Computing System Elements / L.A. Kleiman, V.I. Freyman //

Proceedings of the 2022 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering Conference (ElConRus). 2022. P. 693-696.

82. Kreutzer, S.E. System-Level Fault Diagnosis: A survey / S.E. Kreutzer, S.L. Hakimi // *Microprocessing and Microprogramming*. 1987. Vol. 20. No. 4,5. P. 323-331.

83. Lee, J.-K. A Characterization of t/s-Diagnosability and Sequential t-Diagnosability in Designs / J.-K. Lee, J.T. Butler // *IEEE transaction on Computers*. 1990. Vol. 39. No.10. P. 1298-1304.

84. Li, X. Fault diagnostics between different type of components: A transfer learning approach / X. Li, Y. Hu, M. Li, J. Zheng // *Applied Soft Computing*. 2020. Vol 86. P. 1-24.

85. Luthi, J. Mean value analysis for computer systems with variabilities in workload / J. Luthi, S. Majumdar, G. Haring // *Proceedings of IEEE International Computer Performance and Dependability Symposium*. 1996. P. 32-41.

86. Maddukuri, S.V.P.K. A low cost sensor based autonomous and semi-autonomous fire-fighting squad robot / S.V.P.K. Maddukuri, U.K. Renduchintala, A. Visvakumar, C.S. Pang, K. Mittapally // *2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*. Patna. 2016. P. 279-283.

87. Maeng, J. Comparison Connection Assignment for Self-Diagnosis of Multiprocessor Systems / J. Maeng, M.A. Malek // *Proc. Int. Symp. On Fault-Tolerant Computing(FTCS-11)*. 1981. P.173-175.

88. Manusov, V. Z. Diagnostics of Technical State of Modern Transformer Equipment Using the Analytic Hierarchy Process / V. Z. Manusov, D. V. Orlov, V. V. Frolova // *Environment and Electrical Engineering and Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe): IEEE International Conference, Palermo, 12-15 June 2018: proceedings*. – Los Alamitos: IEEE. 2018. P. 1-6.

89. Miao, Z. Weighted Iterative Truncated Mean Filter / Z. Miao, J. Maeng, M.A. Malek // IEEE Transactions on Signal Processing. 2013. Vol. 61. No. 16. P. 4149-4160.
90. Moore, C.L. Monitoring High Performance Computing Systems for the End User / C.L. Moore, P.S. Khalsa, T.A. Yilk // Cluster Computing: 2015 IEEE International Conference, Chicago, 08–11 September 2015: proceedings. – Los Alamitos: IEEE. 2015. P. 714-716.
91. Pelc, A. Optimal Fault Diagnosis in Comparison Models / A. Pelc // IEEE Transactions on Computers. 1992. Vol. 41. P. 779-786.
92. Pelc, A. Unidirected Graph Models for System-level Fault Diagnosis / A. Pelc // IEEE Transactions on Computers 1991. Vol. 40. No. 11. P.1271-1276.
93. Preparata, F. P. On the Connection Assignment Problem of diagnosable systems / F.P. Preparata, G. Metze, R.T. Chein // IEEE Transactions on Electronic Computers. 1967. Vol. 16. P. 848-854.
94. Rangarajan, S. Diagnosing Arbitrarily Connected Parallel Computers with High Probability / S. Rangarajan, D. Fussell // IEEE Transactions on Computers. 1992. Vol. 41. No. 5. P. 606-615.
95. Russel, J. D. On the Diagnosability of Digital Systems / J.D. Russel, C.R. Kime // Proc. Int. Symp. On Fault-Tolerant Computing. 1973. P.139-144.
96. Russel, J. D. System Fault Diagnosis: Closure and Diagnosability with Repair / J.D. Russel, C.R. Kime // IEEE Transactions on Computers. 1975. Vol. 24. No.11. P.1078-1089.
97. Saaty, T. L. Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors - The Analytic Hierarchy/Network Process / T.L. Saaty // RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics): journal. 2008. Vol. 102. No. 2. P. 251 – 318.

98. Semini, M. Applications of discrete-event simulation to support manufacturing logistics decision-making: a survey / M. Semini, H. Fauske, J.O. Strandhagen // Proceedings of the 2006 Winter Simulation Conference, IEEE. 2006. P. 1946-1953.
99. Sengupta, A. On the Diagnosability of a General Model of System with Three-Valued Test Outcomes / A. Sengupta, A. Sen // IEEE Transactions on Computers. 1986. Vol. 35. No.2. P.170-173.
100. Sharma, P. Discrete-Event Simulation / P. Sharma // International journal of scientific & technology research. 2015. Vol. 4. No. 04. P. 136-140.
101. Somani, A.K. On Diagnosability of Large Fault Sets in Regular Topology-Based Computer Systems / A.K. Somani, O. Peleg // IEEE Transactions on Computers. 1996. Vol. 45, No.8. P. 892-903.
102. Sun, G. Fault diagnosis for rotating machinery based on artificial immune algorithm and evidence theory / G. Sun, Q. Hu, Q. Zhang // Control and Decision (CCDC): 27th International Conference, Qingdao, 23–25 May 2015: proceedings. – Los Alamitos: IEEE. 2015. P. 2975–2979.
103. Tang, Q.-Y. Diagnosis of Parallel Computers with Arbitrary Connectivity / Q.-Y Tang, X. Song // IEEE Transaction on Computers. 1999. Vol. 48. P. 757-761.
104. Vegda, H. Secure and Efficient Approach to Prevent Ad Hoc Network Attacks Using Intrusion Detection System / H. Vegda, N. Modi // 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India. 2018. P. 129-133.
105. Wang, S. The  $g$ -Good-Neighbor Diagnosability of Bubble-Sort Graphs under Preparata, Metze, and Chien's (PMC) Model and Maeng and Malek's (MM)\* Model / S. Wang, Z. Wang // Information. 2019. No, 10:21. P. 1-14.
106. Yang, C.-L. Hybrid Fault Diagnosability with Unreliable Communication Links / C.-L. Yang, G.M. Masson // IEEE transactions on Computers. 1988. Vol. 37. No. 2. P. 175-181.

107. Zhang, J. Fault diagnosis of coal-mine-gas charging sensor networks using iterative learning-control algorithm / J. Zhang, K. Huang // Physical Communication. 2020. P. 1-14.

## Приложение А

### Копии актов внедрения результатов диссертационной работы



#### ЗАО «ИВС-СЕТИ»

Юридический адрес: Луначарского, 3а, ул., Пермь, 614990  
 Почтовый адрес: Луначарского, 3а, ул., Пермь, 614990  
 тел. (342) 219-65-00, 238-52-00, факс: (342) 219-65-10  
 e-mail: [info@ivs-corp.ru](mailto:info@ivs-corp.ru)  
 ОКПО: 12062785, ОГРН: 1025900507676 ИНН/КПП: 5902138574/590201001  
[www.ivs-corp.ru](http://www.ivs-corp.ru)



УТВЕРЖДАЮ  
 Генеральный директор ЗАО «ИВС-Сети»  
 / С. В. Кирюшкин /

3 июня 2022г.

#### АКТ

**о внедрении научных результатов,  
 полученных Клейманом Львом Александровичем  
 при выполнении диссертационной работы  
 на соискание ученой степени кандидата технических наук**

Комиссия в составе:

Председатель: Генеральный директор ЗАО «ИВС-Сети», Кирюшкин Сергей Владимирович

Члены комиссии:

Руководитель отдела «ИБ», Марусин Андрей Игоревич

Ведущий специалист отдела «ИБ», Коробейников Андрей Анатольевич

составила настоящий акт о том, что научные результаты, полученные Клейманом Львом Александровичем в диссертационной работе, внедрены в информационном и программном обеспечении модуля «Стационарные, мобильные и передвижные комплексы фото- и видеофиксации» системы «Безопасный город». В частности, были применены разработанные математическая модель встроенной системы диагностирования и диагностическая модель элемента вычислительной системы, методы расчёта основных показателей разработанных моделей, а также алгоритм перераспределения диагностических задач.

1. В ПО мобильных и передвижных комплексов фото- и видео- фиксации были внесены изменения, позволяющие передавать значения необходимых критериев работоспособности для расчёта характеристик разработанных моделей. Также изменения коснулись функционала диагностирования. Программный код данных функциональных модулей стал способен принимать и выполнять управляющие команды с внешнего устройства, включающие в себя число проверок и перечень диагностируемых данным устройством функциональных модулей. Основные параметры внедрения:

- объём контролируемого оборудования: 100 элементов 2-х типов оборудования;

- рассматриваемый период 24 ч.;

- время работоспособности (время с начала отсчёта, прошедшее до момента, когда более 15 % устройств имели показатель работоспособности ниже порогового значения) увеличилось на 1 час, что составляет примерно 8,3 %.

2. В ПО мобильного поста были внесены изменения, позволяющие реализовать расчёты математической модели встроенной системы диагностирования, а также диагностических моделей элементов вычислительной системы. Также изменения коснулись подсистемы генерации и отправки уведомлений в центр обработки и отображения данных. Были добавлены новые триггеры, вызываемые событиями, возникающими при расчёте значений моделей. Таким образом, за 24 ч. удалось предотвратить 8 отказов элементов системы.

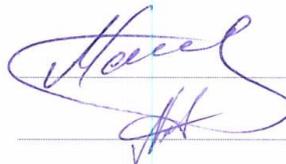
За время эксплуатации программное обеспечение функционировало устойчиво, без сбоев, обеспечивая стабильную работу всех компонентов модуля системы.

Председатель комиссии:



/С. В. Кирюшкин/

Члены комиссии:


/А.И. Марусин/

/А.А. Коробейников/

« 3 » июня 2022



УТВЕРЖДАЮ  
Директор  
ООО «ПроИнфоСервис»

«30» мая 2022 г. / Шлыков А.В. /

«30» мая 2022 г.

**АКТ**

**о внедрении результатов научной работы,  
Клеймана Льва Александровича  
представленной на соискание ученой степени  
кандидата технических наук**

Комиссия в составе:

Председатель: директор по разработке информационных систем Вильдеман А.В.

Члены комиссии:

ведущий менеджер проекта Буланов А.Ю.,

инженер-программист 1-й категории Турунцев К.А.,

составила настоящий акт о том, что в компании внедрено разработанное Клейманом Л.А. информационное и программное обеспечение «Модуль «Построение оптимальных маршрутов для проведения ремонтно-восстановительных работ фото- и видеофиксаторов», обеспечивающих создание путевых листов проведения ремонтно-восстановительных работ передвижных и мобильных фото- и видеофиксаторов в системе обработки и отображения информации, входящей в состав программно-аппаратного комплекса «Безопасный город». В составе этого ПО был реализован разработанный алгоритм динамического распределения диагностических задач, а именно процедуры формирования и обработки уведомлений о техническом состоянии системы.

Данный модуль был доработан для возможности учета передаваемой с мобильного поста информации о состоянии кластера вычислительной системы, в который входят устройства КРИС-П и СКАТ-П. Учет этих данных, а также данных о работе алгоритма динамического распределения диагностической нагрузки, позволил выделить этап работы системы, когда возникновение отказов подвергается кластеризации (смещению в более узкий временной интервал). Основываясь на данной информации и произведенных доработках ПО, модуль построения оптимальных маршрутов проведения ремонтно-восстановительных работ смог строить более эффективные как по стоимости (уменьшение количества выездов), так и по времени (оптимальный маршрут) маршруты.

По результатам доработок и внедрения сделаны следующие выводы: за рассматриваемый период (72 ч.) по причине получения информации о кластеризации отказов время восстановления уменьшилось в среднем на 1,5 часа (улучшение примерно на 5 % относительно характеристики до внедрения). Количество необходимых выездов ремонтно-восстановительной бригады за рассматриваемый интервал времени уменьшилось на 4 (примерно 10 % от характеристики до внедрения).

Полученные результаты позволяют сделать вывод о том, что эффект от внедрения можно считать положительными. Данные доработки рекомендованы к тиражированию на большее число учитываемых функциональных модулей.

Председатель комиссии:

Члены комиссии:

«30» мая 2022 г. / Вильдеман А.В. /

«30» мая 2022 г. / Буланов А.Ю. /

«30» мая 2022 г. / Турунцев К.А. /

УТВЕРЖДАЮ  
Проректор по образовательной  
деятельности  
Пермского национального  
исследовательского  
политехнического университета  
доктор технических наук, доцент



 / А.Б. Петроченков /  
« 17 » июня 2022 г.

**АКТ**  
**о внедрении научных результатов,**  
**полученных Клейманом Львом Александровичем**  
**при выполнении диссертационной работы**  
**на соискание ученой степени кандидата технических наук**  
**«Повышение надежности вычислительных систем**  
**на основе динамического распределения диагностических задач»**

Комиссия в составе:

Председатель: д.т.н., профессор Тюрин С.Ф.

Члены комиссии:

д.т.н., профессор Хижняков Ю.Н.,

к.т.н., доцент Шабуров А.С.,

составила настоящий акт о том, что основные теоретические положения и практические результаты диссертационной работы Клеймана Л.А. использованы в образовательном процессе, реализуемом для бакалавров направлений подготовки 11.03.02 «Инфокоммуникационные технологии и системы связи» и 27.03.04 «Управление в технических системах». Предложенные модели, методы и алгоритм динамического распределения диагностических задач нашли применение в дисциплинах «Надежность, техническая диагностика и информационная безопасность инфокоммуникационных и информационно-управляющих систем», «Информатика в приложении к отрасли», «Научно-исследовательская работа». Результаты диссертационного исследования применены в рамках практических занятий в виде методик расчётов надежности и реализации алгоритмов на языках программирования HTML, CSS, JavaScript.

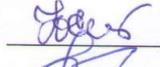
Эффект от внедрения результатов диссертационной работы заключается в повышении уровня освоения профессиональных компетенций и их компонентов (знаний, умений, владений) в области диагностики и контроля вычислительных систем и их элементов, что соответствует требованиям Федеральных государственных образовательных стандартов высшего образования.

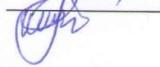
Результаты внедрения результатов диссертационной работы обсуждались на заседании кафедры «Автоматика и телемеханика» 25.04.2022 г., протокол № 10.

Председатель комиссии

Члены комиссии:

 / Тюрин С.Ф. /

 / Хижняков Ю.Н. /

 / Шабуров А.С. /

« 16 » мая 2022 г.

## Приложение Б

### Листинг разработанной программы моделирования

```

reconfigurationAlgorithm(computingObjects){
  let middleWorkersCount = (computingObjects.middleCriticalObjects.data.length
    - Math.floor(computingObjects.middleCriticalObjects.data.length / 2));
  let iterationNumber = 0;
  let tStart = parseInt(this.tInterval);
  let graphData = [];
  let deltaPInt = parseFloat(this.deltaP);
  let pPorogInt = parseInt(this.pPorog);
  let currentIterationMiddleWorkers = this.getCurrentIterationMiddleWorkers(computingObjects,
middleWorkersCount, iterationNumber).slice(0);
  let notCriticalChecks = computingObjects.notCriticalObjects.data.reduce((accumulator, currentValue) => {
    return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
  }, 0);
  let middleCriticalChecks = currentIterationMiddleWorkers.reduce((accumulator, currentValue) => {
    return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
  }, 0);
  let checksNeededCount = computingObjects.criticalObjects.data.length +
(computingObjects.middleCriticalObjects.data.length - currentIterationMiddleWorkers.length);
  let availableChecks = notCriticalChecks + middleCriticalChecks;
  while (checksNeededCount <= availableChecks) {
    let currentIterationWorkers = [].concat(currentIterationMiddleWorkers,
computingObjects.notCriticalObjects.data.slice(0));
    let commonKoff = 0;
    currentIterationWorkers = currentIterationWorkers.filter((worker) => {
      let availableWorkerChecks = Math.floor((worker.currentPowerLevel - pPorogInt) / deltaPInt);
      return availableWorkerChecks > 0;
    });
    currentIterationWorkers.forEach((worker) => {
      worker.currentIterationKoff = (worker.currentPowerLevel + (100 - worker.currentCPULevel)) / (2 * 100);
      commonKoff += worker.currentIterationKoff;
    });
    currentIterationWorkers.sort((w1, w2) => {
      if (w1.currentIterationKoff > w2.currentIterationKoff) {
        return -1;
      }
      else if (w1.currentIterationKoff < w2.currentIterationKoff) {
        return 1;
      }
      return 0;
    });
    let checksToDistribute = checksNeededCount;
    let undistributedChecks = 0;
    currentIterationWorkers.forEach((worker) => {
      let currentChecks = Math.ceil(checksNeededCount * worker.currentIterationKoff / commonKoff);
      let availableWorkerChecks = Math.floor((worker.currentPowerLevel - pPorogInt) / deltaPInt);
      if (currentChecks > availableWorkerChecks) {
        undistributedChecks += (currentChecks - availableWorkerChecks);
        if (availableWorkerChecks > checksToDistribute) {
          worker.currentChecks = (checksToDistribute > 0) ? checksToDistribute : 0;
        }
      }
      else {
        worker.currentChecks = (checksToDistribute > 0) ? availableWorkerChecks : 0;
      }
    });
  }
}

```

```

    }
    else {
      if (currentChecks > checksToDistribute) {
        worker.currentChecks = (checksToDistribute > 0) ? checksToDistribute : 0;
      }
      else {
        worker.currentChecks = (checksToDistribute > 0) ? currentChecks : 0;
      }
    }
    checksToDistribute -= (checksToDistribute === 0) ? 0 : currentChecks;
    worker.currentPowerLevel -= (worker.currentChecks * deltaPInt);
  });
  if (undistributedChecks > 0) {
    let workersForUndistribute = currentIterationWorkers.filter((worker) => {
      let availableWorkerChecks = Math.floor((worker.currentPowerLevel - pPorogInt) / deltaPInt);
      return availableWorkerChecks > 0;
    });
    workersForUndistribute.sort((w1, w2) => {
      let w1availableWorkerChecks = Math.floor((w1.currentPowerLevel - pPorogInt) / deltaPInt);
      let w2availableWorkerChecks = Math.floor((w2.currentPowerLevel - pPorogInt) / deltaPInt);
      if (w1availableWorkerChecks > w2availableWorkerChecks) {
        return -1;
      }
      else if (w1availableWorkerChecks < w2availableWorkerChecks) {
        return 1;
      }
      return 0;
    });
    workersForUndistribute.forEach((worker) => {
      let availableWorkerChecks = Math.floor((worker.currentPowerLevel - pPorogInt) / deltaPInt);
      let addedChecks = (availableWorkerChecks > undistributedChecks) ? undistributedChecks :
availableWorkerChecks;
      worker.currentChecks = (worker.currentChecks) ? worker.currentChecks + addedChecks : addedChecks;
      undistributedChecks -= addedChecks;
    });
  }
  let workingCount = 0;
  Object.keys(computingObjects).forEach((groupName)=>{
    if(groupName !== 'criticalObjects'){
      workingCount += computingObjects[groupName].data.filter((object)=>{
        return object.currentPowerLevel > (pPorogInt + deltaPInt);
      }).length;
    }
  });
  graphData.push({
    "x" : tStart,
    "y" : workingCount,
  });
  tStart += parseInt(this.tInterval);
  iterationNumber++;
  currentIterationMiddleWorkers = this.getCurrentIterationMiddleWorkers(computingObjects,
middleWorkersCount, iterationNumber).slice(0);
  notCriticalChecks = computingObjects.notCriticalObjects.data.reduce((accumulator, currentValue) => {
    return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
  }, 0);
  middleCriticalChecks = currentIterationMiddleWorkers.reduce((accumulator, currentValue) => {
    return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
  }, 0);

```

```

        checksNeededCount = computingObjects.criticalObjects.data.length +
        (computingObjects.middleCriticalObjects.data.length - currentIterationMiddleWorkers.length);
        availableChecks = notCriticalChecks + middleCriticalChecks;
    }
    return graphData;
},

oneToAllAlgorithm(computingObjects){
    let middleWorkersCount = (computingObjects.middleCriticalObjects.data.length
        - Math.floor(computingObjects.middleCriticalObjects.data.length / 2));
    let iterationNumber = 0;
    let tStart = parseInt(this.tInterval);
    let graphData = [];
    let deltaPInt = parseFloat(this.deltaP);
    let pPorogInt = parseInt(this.pPorog);
    let currentIterationMiddleWorkers = this.getCurrentIterationMiddleWorkers(computingObjects,
        middleWorkersCount, iterationNumber).slice(0);
    let notCriticalChecks = computingObjects.notCriticalObjects.data.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    let middleCriticalChecks = currentIterationMiddleWorkers.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    let checksNeededCount = computingObjects.criticalObjects.data.length +
        (computingObjects.middleCriticalObjects.data.length - currentIterationMiddleWorkers.length);
    let availableChecks = notCriticalChecks + middleCriticalChecks;
    while(checksNeededCount < availableChecks) {
        let currentIterationWorkers = [].concat(currentIterationMiddleWorkers,
            computingObjects.notCriticalObjects.data.slice(0));
        currentIterationWorkers.sort((w1, w2) => {
            if(w1.currentPowerLevel > w2.currentPowerLevel){
                return -1;
            }
            else if(w1.currentPowerLevel < w2.currentPowerLevel){
                return 1;
            }
            return 0;
        });
        let checksToDistribute = checksNeededCount;
        currentIterationWorkers.forEach((worker)=>{
            let availableWorkerChecks = Math.floor((worker.currentPowerLevel - pPorogInt) / deltaPInt);
            let currentChecks = 0;
            if(checksToDistribute > 0 && availableWorkerChecks > 0){
                currentChecks = (checksToDistribute > availableWorkerChecks) ? availableWorkerChecks :
                checksToDistribute;
            }
            checksToDistribute -= currentChecks;
            worker.currentPowerLevel -= (currentChecks * deltaPInt);
        });
        let workingCount = 0;
        Object.keys(computingObjects).forEach((groupName)=>{
            if(groupName !== 'criticalObjects'){
                workingCount += computingObjects[groupName].data.filter((object)=>{
                    return object.currentPowerLevel > (pPorogInt + deltaPInt);
                }).length;
            }
        });
        graphData.push({
            "x" : tStart,

```

```

        "y" : workingCount,
    });
    tStart += parseInt(this.tInterval);
    iterationNumber++;
    currentIterationMiddleWorkers = this.getCurrentIterationMiddleWorkers(computingObjects,
middleWorkersCount, iterationNumber).slice(0);
    notCriticalChecks = computingObjects.notCriticalObjects.data.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    middleCriticalChecks = currentIterationMiddleWorkers.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    checksNeededCount = computingObjects.criticalObjects.data.length +
(computingObjects.middleCriticalObjects.data.length - currentIterationMiddleWorkers.length);
    availableChecks = notCriticalChecks + middleCriticalChecks;
    }
    return graphData;
},

oneToAllSimpleAlgorithm(computingObjects){
    let middleWorkersCount = (computingObjects.middleCriticalObjects.data.length
        - Math.floor(computingObjects.middleCriticalObjects.data.length / 2));
    let iterationNumber = 0;
    let tStart = parseInt(this.tInterval);
    let graphData = [];
    let deltaPInt = parseFloat(this.deltaP);
    let pPorogInt = parseInt(this.pPorog);
    let currentIterationMiddleWorkers = this.getCurrentIterationMiddleWorkers(computingObjects,
middleWorkersCount, iterationNumber).slice(0);
    let notCriticalChecks = computingObjects.notCriticalObjects.data.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    let middleCriticalChecks = currentIterationMiddleWorkers.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    let checksNeededCount = computingObjects.criticalObjects.data.length +
(computingObjects.middleCriticalObjects.data.length - currentIterationMiddleWorkers.length);
    let availableChecks = notCriticalChecks + middleCriticalChecks;
    while(checksNeededCount < availableChecks) {
        let currentIterationWorkers = [].concat(currentIterationMiddleWorkers,
computingObjects.notCriticalObjects.data.slice(0));
        let checksToDistribute = checksNeededCount;
        currentIterationWorkers.forEach((worker)=>{
            let availableWorkerChecks = Math.floor((worker.currentPowerLevel - pPorogInt) / deltaPInt);
            let currentChecks = 0;
            if(checksToDistribute > 0 && availableWorkerChecks > 0){
                currentChecks = (checksToDistribute > availableWorkerChecks) ? availableWorkerChecks :
checksToDistribute;
            }
            checksToDistribute -= currentChecks;
            worker.currentPowerLevel -= ( currentChecks * deltaPInt);
        });
        let workingCount = 0;
        Object.keys(computingObjects).forEach((groupName)=>{
            if(groupName !== 'criticalObjects'){
                workingCount += computingObjects[groupName].data.filter((object)=>{
                    return object.currentPowerLevel > (pPorogInt + deltaPInt);
                }).length;
            }
        }
    }
}

```

```

    });
    graphData.push({
        "x" : tStart,
        "y" : workingCount,
    });
    tStart += parseInt(this.tInterval);
    iterationNumber++;
    currentIterationMiddleWorkers = this.getCurrentIterationMiddleWorkers(computingObjects,
middleWorkersCount, iterationNumber).slice(0);
    notCriticalChecks = computingObjects.notCriticalObjects.data.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    middleCriticalChecks = currentIterationMiddleWorkers.reduce((accumulator, currentValue) => {
        return accumulator + Math.floor((currentValue.currentPowerLevel - parseInt(this.pPorog)) / deltaPInt);
    }, 0);
    checksNeededCount = computingObjects.criticalObjects.data.length +
(computingObjects.middleCriticalObjects.data.length - currentIterationMiddleWorkers.length);
    availableChecks = notCriticalChecks + middleCriticalChecks;
    }
    return graphData;
},

```