

На правах рукописи



ПЕТУШКОВ Григорий Валерьевич

**ФУНКЦИОНАЛЬНАЯ АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ
СИСТЕМ КЛАСТЕРНОГО ТИПА НА ОСНОВЕ
ИСПОЛНИМЫХ МОДЕЛЕЙ И ПРАВИЛ ПРЕДМЕТНОЙ ОБЛАСТИ**

**Специальность 2.3.2. Вычислительные системы и их элементы
(технические науки)**

А в т о р е ф е р а т
диссертации на соискание ученой степени
кандидата технических наук

Москва 2025

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего образования «МИРЭА – Российский технологический университет» (РТУ МИРЭА).

Научный руководитель – доктор физико-математических наук,
профессор, академик РАН
Сигов Александр Сергеевич

Официальные оппоненты: Кулагин Владимир Петрович,
доктор технических наук, профессор,
ордена Трудового Красного Знамени
федеральное государственное бюджетное
образовательное учреждение высшего
образования «Московский технический
университет связи и информатики»,
профессор кафедры математической
кибернетики и информационных технологий
(г. Москва);

Перепелкин Дмитрий Александрович,
доктор технических наук, профессор,
федеральное государственное бюджетное
образовательное учреждение высшего
образования «Рязанский государственный
радиотехнический университет
им. В. Ф. Уткина», декан факультета
вычислительной техники (г. Рязань)

Ведущая организация – акционерное общество «Научно-
производственное предприятие «Рубин»
(г. Пенза)

Защита диссертации состоится 25 декабря 2025 г. в ___ часов на заседании диссертационного совета 24.2.357.06 в ФГБОУ ВО «Пензенский государственный университет» по адресу: 440026, г. Пенза, ул. Красная, 40.

С диссертацией и авторефератом можно ознакомиться в библиотеке ФГБОУ ВО «Пензенский государственный университет». Диссертация размещена на сайте: https://dissov.pnzgu.ru/ecspertiza/Tekhnicheskie_nauki/petushkov

Автореферат разослан «___» _____ 2025 г.

Ученый секретарь
диссертационного совета



Бождай Александр Сергеевич

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Кластеризация – это одно из самых современных направлений в области создания вычислительных систем. Появление и развитие кластерных вычислительных систем обусловлены прогрессом в области сетевых технологий, повышенным спросом на доступные высокопроизводительные вычисления, растущим внедрением методов искусственного интеллекта и машинного обучения. Кластерные вычислительные системы, как правило, отличаются архитектурной простотой, обусловленной регулярностью и однородностью, разделением аппаратуры, связью между узлами в ограниченной окрестности. Как правило, вычислительный кластер определяется как разновидность параллельной или распределенной вычислительной системы, когда взаимосвязанные компьютеры образуют единый вычислительный ресурс, рассматриваемый пользователями с позиций единого системного образа. Кластерные, грид- и облачные вычисления имеют глубокие эволюционные связи, однако процессы управления и особенности применения различаются.

В настоящее время получает распространение облачная технология HPC-as-a-Service – высокопроизводительные вычисления как сервис, что в существенной степени расширяет границы применимости вычислительных кластеров и делает их ресурсы доступными для многих категорий пользователей.

Существенный вклад в развитие высокопроизводительных вычислительных систем внесли Н. Ф. Бахарева, М. В. Бобырь, О. М. Брехов, В. В. Воеводин, Вл. В. Воеводин, Б. А. Головкин, А. В. Каляев, И. А. Каляев, В. С. Князьков, В. П. Корячко, Б. В. Костров, М. Г. Курносов, А. М. Ларионов, В. К. Левин, Д. А. Перепелкин, А. С. Сигов, А. С. Симонов, В. Н. Тарасов, В. С. Титов, В. В. Топорков, А. Г. Финогеев, В. Г. Хорошевский, J. J. Dongarra, M. J. Flinn, V. A. Forouzan, I. Foster, G. C. Fox, J. L. Hennessy, M. van Steen, A. Tanenbaum.

Вопросам формализации параллельных и распределенных компьютерных систем посвящены работы О. Л. Бандман, В. А. Вальковского, Н. П. Вашкевича, В. А. Горбатова, В. Н. Дубинина, Д. В. Жевнерчука, В. Е. Котова, В. П. Кулагина, Л. С. Ломакиной, П. П. Макарычева, Д. В. Пащенко, И. Г. Сидоркиной, А. А. Шалыто, P. J. Antsaklis, E. Boerger, Y. Gurevich, M. V. Iordache, R. Janicki, R. M. Keller, D. J. Kuck, A. Pritsker.

Основную **проблемную ситуацию** создает использование вычислительных кластеров для ограниченной предметной области, касающейся в основном научных расчетов, что затрудняет использование кластеров расширенным кругом пользователей.

Задача использования преимуществ кластерных вычислений за счет выбора наиболее подходящей модели организации функциональной архитектуры является **актуальной**, поскольку ее решение позволит осуществлять предметную ориентацию вычислительных кластеров на конкретного пользователя, использующего кластер как конечный продукт, не требующий значительной

модификации его функциональной архитектуры при внедрении. В диссертационной работе предлагается разрешать проблемную ситуацию за счет использования новых методов организации функциональной архитектуры вычислительных систем кластерного типа на основе современной концепции исполнимых моделей и правил предметной области. Рассматриваемые с позиций искусственного интеллекта, основанного на знаниях и правилах, подобные модели представляют собой сценарии действий, содержащие как процедурную, так и декларативную составляющие модели представления знаний. Декларативная составляющая может быть задана при помощи исчисления предикатов первого порядка, а процедурная, или императивная, – алгебраическими моделями алгоритмов и частичными конечными автоматами. В качестве интегрирующих компонент предлагается использовать известные в международной практике событийно-ориентированный подход (*EDA – Event-Driven Architecture*), основанный на событиях, происходящих при работе приложений и управляющих программ в вычислительном кластере, а также метод разработки систем, основанный на использовании правил предметной области (*DDD – Domain-Driven Design*).

Объектом исследования является функциональная архитектура вычислительных систем кластерного типа.

Предметом исследования являются исполнимые модели, методы и алгоритмы, определяющие функциональную архитектуру вычислительных систем кластерного типа на основе правил и знаний о предметной области.

Целью исследования является повышение эффективности и отказоустойчивости вычислительных систем кластерного типа за счет совершенствования процессов управления на основе новых моделей, методов и алгоритмов, учета требований и свойств предметной области.

Для достижения поставленной цели необходимо решить следующие основные **задачи**:

- провести анализ современного состояния, специфики использования, целей и задач совершенствования функциональной архитектуры вычислительных систем кластерного типа;
- определить подходы к разработке функциональной архитектуры, основанной на исполнимых моделях, знаниях и правилах предметной области, назначении вычислительных кластеров;
- разработать новые исполнимые модели и методы совершенствования вычислительных систем кластерного типа, определяющие их функциональную архитектуру, масштабируемость, основанные на анализе событий, происходящих при работе приложений и управляющих программ;
- разработать логико-вероятностные и исполнимые логико-алгебраические модели на основе методологического подхода к макро- и микроанализу функциональной архитектуры вычислительных систем кластерного типа;
- разработать сетевые автоматные, вероятностные автоматные и логико-алгебраические исполнимые модели функционирования вычислительных систем кластерного типа как основы для создания управляющих программ,

иллюстрирующих работу вычислительных кластеров при последовательно-параллельной и параллельно-конвейерной обработке заданий;

– разработать метод и модель функциональной организации масштабируемых вычислительных систем кластерного типа с повышенной отказоустойчивостью.

Методы исследования. При решении поставленных задач использованы методы математической логики, теории графов, алгебры алгоритмов, теории автоматов и сетей Петри. При проведении экспериментов и проверке предложенных решений использованы методы статистического моделирования.

Научная новизна исследования заключается в следующем:

1. На основе проведенного анализа современного состояния и развития архитектуры вычислительных систем предложена концепция организации функциональной архитектуры вычислительных систем кластерного типа, *отличающаяся* базированием на исполнимых моделях, знаниях и правилах предметной области и *позволяющая* осуществлять предметную ориентацию вычислительных кластеров на конкретного пользователя, использующего кластер как конечный продукт.

2. Предложен метод разработки функциональной архитектуры вычислительных кластеров, основанный на исполнимых логико-алгебраических операционных моделях, *отличающийся* от известных тем, что он основан на знаниях и правилах предметной области, на назначении вычислительных кластеров и на анализе событий, происходящих при работе приложений и управляющих программ, и *позволяющий* создавать новые исполнимые модели для совершенствования вычислительных систем кластерного типа, определяющие их функциональную архитектуру, масштабируемость и соответствие предметной области.

3. Предложены новые исполнимые модели вычислительных систем кластерного типа на макроуровне, *позволяющие* расширить диапазон их функциональности и применимости: логико-алгебраические операционные и статистические модели «Круглый стол», «Резидент – агенты», «Собрание 1», «Собрание 2», и *отличающиеся* от известных ориентацией на современные диалоговые параллельные взаимодействия при обработке данных в больших социальных сообществах пользователей при помощи масштабируемых приложений, реализуемых кластером.

4. Разработаны новые детализированные автоматные, вероятностные автоматные и логико-алгебраические операционные модели вычислительных систем кластерного типа на микроуровне, *отличающиеся* исполнимым характером и *позволяющие* произвести проектирование приложений на уровне сетевых операций и дать оценку производительности кластера в целом.

5. Предложен метод повышения производительности вычислительных кластеров при выполнении сетевых приложений, *отличающийся* развертыванием операторов распределенного алгоритма на различных узлах кластера

и *позволяющий* при реализации выполнять вычисления, определяемые передачами управления и данных между узлами в параллельно-конвейерном режиме.

6. Предложена исполнимая логико-алгебраическая операционная модель функциональной организации вычислительных систем кластерного типа с повышенной отказоустойчивостью, *основанная* на динамическом резервировании и многократной замене узлов и *позволяющая* при ее реализации обеспечить отказоустойчивость кластера в условиях сильной деградации, обусловленной внутренними и внешними причинами.

Практическая значимость результатов исследования заключается в следующем:

1. Предложенная концепция организации функциональной архитектуры вычислительных систем кластерного типа *позволяет* расширить диапазон их применимости за счет ориентации на предметную область пользователя, использующего вычислительный кластер как конечный продукт с расширяемой функциональностью.

2. Предложенный метод разработки функциональной архитектуры вычислительных кластеров, основанный на учете знаний и правил предметной области, *позволяет* разрабатывать логико-алгебраические операционные модели, исполнительный характер которых предоставляет возможность учета и реализации событий, планируемых при работе создаваемых масштабируемых сетевых приложений и управляющих программ.

3. Предложенные и реализованные новые исполнимые модели вычислительных систем кластерного типа на макроуровне *позволяют* расширить диапазон их функциональности и применимости: логико-алгебраические операционные и статистические модели «Круглый стол», «Резидент – агенты», «Собрание 1», «Собрание 2». Проведенные статистические эксперименты с данными моделями позволяют оценить характеристики производительности при различных уровнях параллельной обработки данных в кластерах, выбрать число узлов кластера и портов коммутатора. Время выполнения статистических моделей при 1000–10 000 прогонах занимает не более 2–25 с, что позволяет оперативно получать характеристики производительности вычислительных кластеров и других вычислительных систем с похожей системной и функциональной архитектурой.

4. Разработанные новые автоматные, вероятностные автоматные и логико-алгебраические операционные модели вычислительных систем кластерного типа на микроуровне *позволяют* учитывать особенности работы приложений на уровне сетевых операций и давать оценку производительности кластера в целом при реализации последовательно-параллельных приложений. Статистические эксперименты продемонстрировали полное соответствие полученных характеристик производительности для базовых примеров закону Амдала.

5. Предложенный метод повышения производительности вычислительных кластеров при выполнении сетевых приложений *позволяет* повысить их

пропускную способность при развертывании операторов распределенного алгоритма на узлах кластера и выполнять вычисления, определяемые передачами управления и данных между узлами в параллельно-конвейерном режиме. Статистические эксперименты продемонстрировали повышение пропускной способности в 2–5 раз.

6. Предложенная исполнимая логико-алгебраическая операционная модель функциональной организации вычислительных систем кластерного типа с повышенной отказоустойчивостью, основанная на динамическом резервировании и замене узлов, *позволяет* обеспечить отказоустойчивость кластера в условиях сильной деградации, обусловленной внутренними и внешними причинами. Статистические эксперименты показали, что работоспособность кластера сохраняется при его деградации до одного узла с соответствующим понижением производительности. Результат приведенного анализа базового примера показал, что в условиях сильной, но не стопроцентной, деградации вычислительного кластера было не только обеспечено выполнение разбитого на части задания, но и достигнуто ускорение вычислений в 5,33 раза за счет частичного сохранения параллельного режима. Модель расширена на случай анализа отказоустойчивости систем распределенных вычислений и волонтерских кластеров с большим и нестабильным числом узлов (до 100–150 и более вычислительных узлов).

Внедрение результатов исследования. Результаты диссертационной работы внедрены: в ООО «ПРАЙМГРУП» (г. Красногорск) в ходе разработки центров обработки данных (ЦОД) и выработки системно-технических решений для предприятий топливно-энергетического комплекса; в АО «Концерн «Моринформсистем-Агат» (Корпорация морского приборостроения, г. Москва) при использовании предложенной в диссертации методики оценки надежности кластерной вычислительной системы для ЦОД; в АО «Научно-технический центр «Атлас» (г. Москва) результаты диссертации применены в ходе выполнения опытно-конструкторской работы «Платформа», в том числе внедрены предложенные в диссертации метод и исполнимые логико-алгебраические операционные модели функциональной организации вычислительных систем с повышенной отказоустойчивостью, отличающиеся динамическим резервированием и заменой узлов и позволяющие обеспечить отказоустойчивость кластера в условиях сильной деградации, обусловленной внутренними и внешними причинами. Результаты работы внедрены также в федеральном государственном бюджетном образовательном учреждении высшего образования «МИРЭА – Российский технологический университет» (г. Москва) в рамках преподавания дисциплин «Вычислительные системы реального времени», «Промышленный Интернет вещей», «Архитектура вычислительных средств систем управления». Все виды внедрений подтверждаются актами.

Достоверность и обоснованность результатов работы. Обоснованность и достоверность полученных результатов определяются корректным использованием теории алгоритмов и конечных автоматов, исчисления предика-

тов первого порядка, теории вероятностей и математической статистики, методов компьютерного моделирования, внедрением результатов в учебный процесс вуза и в промышленности.

Соответствие паспорту специальности. Результаты исследований соответствуют п. 1 «Разработка научных основ создания и исследования общих свойств и принципов функционирования вычислительных систем и их элементов»; п. 3 «Разработка научных подходов, методов, алгоритмов и программ, обеспечивающих надежность, сбое- и отказоустойчивость, контроль и диагностику функционирования вычислительных систем и их элементов»; п. 7 «Разработка научных методов и алгоритмов организации параллельной и распределенной обработки информации, многопроцессорных, многоядерных, многомашинных и специальных вычислительных систем» паспорта научной специальности 2.3.2. Вычислительные системы и их элементы.

Основные положения, выносимые на защиту:

1. Концепция предметной ориентации вычислительных кластеров на конкретного пользователя, использующего кластер как конечный продукт.

2. Метод разработки функциональной архитектуры вычислительных кластеров, основанный на знаниях и правилах предметной области, а также на учете событий, происходящих при работе приложений и управляющих программ.

3. Исполнимые логико-алгебраические операционные и статистические модели вычислительных систем кластерного типа на *макроуровне*, позволяющие расширить диапазон их функциональности и применимости: «Круглый стол», «Резидент – агенты», «Собрание 1», «Собрание 2», и отличающиеся от известных ориентацией на диалоговые параллельные взаимодействия при обработке данных и сообщений.

4. Автоматные, вероятностные автоматные и логико-алгебраические операционные модели вычислительных систем кластерного типа на *микроуровне*, позволяющие при реализации учитывать особенности работы приложений на уровне сетевых операций и давать оценку производительности кластера в целом при реализации последовательно-параллельных приложений.

5. Метод повышения производительности вычислительных кластеров при выполнении сетевых приложений на основе определяемых между узлами передач управления и данных в сетевом параллельно-конвейерном режиме.

6. Исполнимая логико-алгебраическая операционная модель функциональной организации вычислительных систем кластерного типа с повышенной отказоустойчивостью, основанная на динамическом резервировании и многократной замене узлов, позволяющая при реализации обеспечить отказоустойчивость кластера в условиях сильной деградации.

Апробация результатов исследования. Результаты исследования докладывались автором на XVII научно-практической конференции «Современные информационные технологии в управлении и образовании» ФГБУ НИИ

«Восход» (19 апреля 2018 г.), в секции «Построение облачной инфраструктуры для электронного государственного управления», доклад на тему «Перспективы развития элементной и микропроцессорной базы компьютеров и вычислительных систем»; на IX Международной научной конференции «ИТ-Стандарт 2019» (11–12 марта 2019 г.), проводившейся в рамках «Недели российского бизнеса», доклад на тему «Прогнозирование надежности по отношению к отказам программного обеспечения для сложных программно-аппаратных систем». Результаты исследования также докладывались в мае 2018 г. на 3-й научно-технической конференции РТУ МИРЭА, в секции «Вычислительные машины, комплексы и сети», тема доклада – «Прогнозирование надежности серверов и вычислительных систем».

Публикации. По результатам исследования автором опубликованы 13 печатных работ, в том числе 10 статей в журналах из перечня ВАК (в том числе 3 статьи категории K1, 5 статей категории K2), 3 статьи в других изданиях и учебное пособие.

Личный вклад автора. Все выносимые на защиту результаты получены автором диссертации лично и отражены в работах: [1–5] – статьи выполнены в соавторстве с научным руководителем, принявшим участие в определении направления исследования и концептуальных подходов, формулировании актуальных задач и методологии их решения; работы [6–8] выполнены автором лично; в учебном пособии и статьях [9–13], выполненных в соавторстве, личный вклад диссертанта состоял в принятии участия в определении проблемной ситуации, в выборе и применении методов решения задач.

Структура диссертации. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы и приложений с общим объемом 193 страницы, включая основной текст на 162 страницах (в том числе 25 рисунков и 17 таблиц), список литературы на 15 страницах из 177 наименований и два приложения на 16 страницах (в том числе приведены примеры текстов исполнимых моделей и четыре акта о внедрении).

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, сформулирована цель работы, изложены научная новизна, практическая и теоретическая значимость полученных результатов, сформулированы выносимые на защиту основные положения.

В первой главе на основе проведенного исследования разработок в области функциональной архитектуры кластерных вычислительных систем показана актуальность выполнения работ в данном направлении, что обусловлено увеличением спроса на доступные высокопроизводительные вычисления, растущим внедрением методов искусственного интеллекта и машинного обучения. Отличительной особенностью современных вычислительных систем

кластерного типа на основе локальных сетей с коммутаторами 2-го (канального) и 3-го (межсетевого) уровней от разработанных ранних суперкомпьютеров – однородных вычислительных систем (ОВС) являются большая гибкость архитектуры, обеспечивающая лучшие возможности реконфигурации на уровне связей между модулями, повышенная отказоустойчивость. Естественная распределенность развертывания узлов кластера часто может быть связана с его реализацией на базе локальной или реже на базе глобальной сети с предметной ориентацией на распределенные предприятия, филиалы, офисы, производственные участки и др. В первом случае кластер может быть развернут в среде, где каждый узел должен находиться рядом с управляемым объектом или с другим источником информации. Во втором случае кластер является ресурсом в облачной вычислительной среде.

Масштабируемость, гибкость модульных аппаратных и программных решений для кластерных вычислений позволяет организациям уменьшать или увеличивать размер кластера в зависимости от изменяющихся потребностей. Поэтому задача исследования и реализации функциональной архитектуры вычислительных систем кластерного типа является актуальной.

Анализ современных приложений показал, что для вычислительных систем кластерного типа характерно использование для решения задач вычислительного характера распараллеливаемых численных методов. Предложено расширить состав задач, определяющих функциональную архитектуру вычислительных систем кластерного типа, средствами для управления кластером при организации «круглых столов», конференций, собраний, т.е. задачами, требующими организации работы большого состава участников и имеющими приложения к обработке больших данных.

Предложена концепция организации функциональной архитектуры вычислительных систем кластерного типа, отличающейся базированием на исполнимых моделях, знаниях и правилах предметной области и позволяющей осуществлять предметную ориентацию вычислительных кластеров на конкретного пользователя, использующего кластер как конечный продукт, не требующий значительной модификации его функциональной архитектуры при внедрении.

Во второй главе описан новый метод разработки функциональной архитектуры вычислительных кластеров, основанный на исполнимых логико-алгебраических операционных моделях, на знаниях и правилах предметной области, на назначении вычислительных кластеров и на анализе событий, происходящих при работе приложений и управляющих программ. Метод и технология на его основе позволяют определять и совершенствовать функциональную архитектуру вычислительных систем кластерного типа, ее масштабируемость и соответствие предметной области. Описываются выполненная разработка и экспериментальная апробация на уровне имитационных моделей метода и технологии реализации функциональной архитектуры

вычислительных систем кластерного типа на основе высокоскоростных коммутаторов. Использован логико-вероятностный подход к созданию моделей, определяемый приложениями и программным обеспечением промежуточного уровня *middleware* функциональной архитектуры кластерных вычислительных систем, позволяющий ускорить создание имитационных моделей для ряда важных режимов использования кластера и осуществить переход к логико-алгебраическим формализованным спецификациям на программные приложения.

Построены имитационные логико-вероятностные и логико-алгебраические модели для ряда важных вариантов использования вычислительного кластера: «Круглый стол», «Резидент – агенты», «Собрание 1» и «Собрание 2», проведены необходимые статистические эксперименты с данными моделями, давшие обоснования к реализациям соответствующему моделям программному обеспечению.

Рассматриваемая модель «Круглый стол» позволяет выполнить макроанализ производительности вычислительного кластера в следующей содержательной постановке. В кластере реализуется вычислительный процесс, сопровождаемый межузловым обменом запросами и затребованными данными. Как отправляющие запросы узлы (узлы-источники), так и узлы, принимающие запросы (узлы-приемники), выбираются в соответствии с заданным распределением вероятностей. Каждый запрос предваряется выполнением некоторой подготовительной программы на узле-отправителе. Ответ на запрос сопровождается выполнением некоторой программы на узле-приемнике запроса. Естественным ограничением является тот факт, что номера узла-источника и узла-приемника не могут совпадать. Запросы, подаваемые с одного и того же узла, могут повторяться, что может привести к задержкам при передаче и подготовке запроса. К задержкам также приводят обращения некоторых узлов-источников к одним и тем же узлам-приемникам. Таким образом в вычислительной системе кластерного типа будут взаимодействовать конкурирующие процессы, и образование очередей ожидания не нарушит работу системы.

Для решения этой задачи выбрано построение моделей, основанных на исчислении предикатов первого порядка. Отличием от других логических моделей, использующих исчисление предикатов, является придание модели динамичности за счет использования операций модификации предикатов, определенных на множествах предметной области.

Ядром логико-вероятностной модели является полный орграф $G = (V, U)$, где V – множество вершин; $U = V^2$ – полное множество дуг, $V^2 = V \times V$. Введем $S_x: V_x \rightarrow \{true, false\}$ – унарный предикат, определенный на подмножестве вершин-источников запросов в кластерной вычислительной системе; $S_y: V_y \rightarrow \{true, false\}$ – унарный предикат, определенный на подмножестве вершин-приемников запросов в кластере; подмножества $V_x \subset V$ и $V_y \subset V$ здесь имеют переменный состав и формируются в процессе моделирования. Истин-

ные значения предикатов S_x и S_y , играющих соответственно роли характеристических функций для подмножеств V_x и V_y , будут соответствовать назначению соответствующих вершин из множества V на участие в передаче данных диалогов от источников к приемникам. Очевидно, что V_x и V_y , согласно постановке задачи, – непересекающиеся подмножества ($V_x \cap V_y = \emptyset$) множества V . Предикаты S_x , S_y и множества V_x , V_y изменяются (эволюционируют) в процессе моделирования, причем на любом этапе моделирования множества V_x и V_y равноможны. Моделирование диалоговой системы «Круглый стол» состоит в формировании последовательности реализаций орграфа паросочетаний $H_0, H_1, H_2, \dots, H_m$, имитирующих подключение и отключение участников диалога в кластерной вычислительной системе. Одновременно с этим производится сбор статистических данных, характеризующих загрузку и производительность системы. В теории графов паросочетание, или независимое множество дуг в орграфе, – это набор попарно несмежных дуг. Таким образом, орграфы паросочетаний будут содержать множества некоторых упорядоченных пар (x, y) , где $x \in V_x, y \in V_y, x \neq y$. Задача моделирования упрощается тем, что x и y ($x \neq y$) выбираются из одного и того же множества V , т.е. из множества участников диалога в кластерной вычислительной системе.

Реализации орграфа паросочетаний не могут иметь петель, так как полагается, что участнику «круглого стола» нет необходимости в передаче сообщений самому себе. Если реализация паросочетания орграфа покрывает все его вершины, то это соответствует связи максимального числа пар. Очевидно, что при четном числе вершин n число таких пар максимально и равно $n/2$. Число пар меньше данной величины в случаях, когда наблюдается конфликт запросов при обращении на передачу к одним и тем же приемникам.

Эволюцию орграфа паросочетаний при моделировании предлагается организовать при помощи следующего логико-вероятностного выражения (сценария), сочетающего в себе как декларативные, так и процедурные знания о предметной области:

$$\begin{aligned}
 L_0 = [& (\exists_{Any} x \in V) \neg S_x(x)] [(\exists_{One} (x, y) \in V^2) \neg S_y(y) \& (x \neq y)] \\
 & (\{S_x(x) \leftarrow \text{true}, S_y(y) \leftarrow \text{true}, R(x, y) \leftarrow \text{true}, \text{Transfer}(x, y) \leftarrow \text{true}, \\
 & \text{Delay}(x, y) \leftarrow \text{true}, S_x(x) \leftarrow \text{false}, S_y(y) \leftarrow \text{false}, R(x, y) \leftarrow \text{false}, \\
 & \text{Delay}(x, y) \leftarrow \text{false}, \text{Transfer}(x, y) \leftarrow \text{false}\} \vee \text{Ret}) \vee \text{Ret}),
 \end{aligned} \tag{1}$$

где в квадратные скобки заключены условия успешного выполнения заданных операторов. В фигурные скобки заключены операции модификации предикатов. Квадратные и фигурные скобки являются элементами синтаксиса выражения (1) и доопределяют его операционную семантику. Оператор $(\exists_{Any} x \in V) \neg S_x(x)$ реализует в модели псевдослучайный выбор будущей вершины x графа паросочетаний H при условии, что ранее эта вершина в граф не входила, т.е. выбор реализуется успешно, если истинно высказывание $\neg S_x(x)$ при конкретном вхождении x .

При выполнении данной операции используется генератор целочисленных псевдослучайных величин с заданным законом распределения; вид выбранного распределения вероятностей может задавать приоритетный выбор запросов от участников «круглого стола». При равномерном распределении моделируется равновероятный выбор запросов. Далее оператор $\exists_{One}(x, y) \in V^2$ осуществляет выбор кортежа (x, y) из декартова произведения $V^2 = V \times V$ с учетом того факта, что первый элемент кортежа x уже известен в результате предыдущего выбора и истинно высказывание $\neg S_y(y) \& (x \neq y)$. Отметим, что операторы $\exists_{Any}$ и $\exists_{One}$ не удаляют элементы или кортежи из множеств, а лишь предварительно отмечают их в структурах – списках. Включение или удаление этих элементов осуществляется путем модификации соответствующих предикатов – характеристических функций множеств. В фигурные скобки заключены операции модификации унарных $S_x(x) \leftarrow \text{true}$, $S_y(y) \leftarrow \text{true}$ и бинарного $R(x, y) \leftarrow \text{true}$ предикатов, где значения предметных переменных x, y известны. В результате подобных модификаций добавляются две вершины и инцидентная им дуга в очередной конфигурации графа H паросочетаний. Это построение является ядром моделирующей программы и в дальнейшем используется при реализации программного обеспечения кластерной системы.

Орграф паросочетаний имеет переменную (эволюционирующую) структуру, и его текущее состояние формально может быть определено следующим образом:

$$H = (V_x, V_y, S_x, S_y, R). \quad (2)$$

На основе выражения (1) далее формируется выражение (сценарий), пригодное для спецификации программного обеспечения реальной кластерной вычислительной системы. При этом нужно учесть, что программа должна выполняться в каждом компьютере, подключенном к коммутатору вычислительного кластера. Первый шаг трансформации заключается в том, что выражений, подобных выражению (1), должно быть m , т.е. по одному выражению для будущих программ на каждый компьютер кластера. Ниже представлено логико-алгебраическое выражение, которое использовано в качестве прототипа спецификации сетевого приложения для i -го компьютера кластера:

$$\begin{aligned} M_i = [& Start(x_i)](\{Start(x_i) \leftarrow \text{false}, Master(x_i) \leftarrow \text{true}, \\ & Work^*(x_i) \leftarrow \text{true}\} \vee Ret), ([(\exists_{One}(x_i, y_j) \in V^2) \neg S_y(y_j) \& (x_i \neq y_j)] \\ & (\{S_x(x_i) \leftarrow \text{true}, S_y(y_j) \leftarrow \text{true}, R(x_i, y_j) \leftarrow \text{true}, Transfer(x_i, y_j) \leftarrow \text{true}, \\ & Proc^*(x_i, y_j) \leftarrow \text{true}, S_x(x_i) \leftarrow \text{false}, S_y(y_j) \leftarrow \text{false}, R(x_i, y_j) \leftarrow \text{false}, \\ & Transfer(x_i, y_j) \leftarrow \text{true}, Proc^*(x_i, y_j) \leftarrow \text{false}, \\ & Master(x_i) \leftarrow \text{false}, Work^*(x_i) \leftarrow \text{false}\} \vee Ret), \\ & i, j = 1, 2, \dots, m; i \neq j. \end{aligned} \quad (3)$$

От выражения (1) выражение (3) отличается введением сетевых операторов, индексов i и j , которым в реальном сетевом приложении соответствуют

адреса компьютеров, подключенных к коммутатору. Задано новое имя выражения M_i (первая буква от имени *Master*, что означает, что передающий узел кластера получает статус мастера). Введены новые унарные предикаты *Start*, *Master*, *Work**, по-новому интерпретируется бинарный предикат *Proc**. Звездочки при предикатных именах *Work** и *Proc** означают, что при реализации прототипного программного обеспечения кластера этим предикатам соответствуют вычислительные процедуры.

Например, при проведении статистических экспериментов с моделью «Круглый стол» задавалось время подготовки запроса узлом-отправителем («обдумывания темы беседы»), распределенное в соответствии со смещенным экспоненциальным распределением $F(t) = 1 - \exp(-\lambda(t - t_0))$ с параметрами $t_{cp} = 100$ мс, $t_0 = 10$ мс, и время обработки запроса узлом-приемником («обдумывание ответа») с теми же параметрами. Число портов коммутатора полагалось равным числу N узлов кластера. Результаты моделирования представлены в таблицах 1 и 2.

Таблица 1

$n_{п/п}$	Число узлов N	T_{Pair} , мс	T_1 , мс	K_{Slow}
1	8	803	220	3,65
2	16	980	220	4,45
3	24	1077	220	4,9
4	32	1150	220	5,22
5	64	1300	220	5,9
6		1449	220	6,59
7	256	1595	220	7,25
8	512	1732	220	7,87
9	1024	1870	220	8,5

Таблица 2

$n_{п/п}$	Число узлов N	Количество портов N_P	T_{Pair} , мс	K_{Lim}
1	1024	1024	1870	1
2	1024	512	1873	1
3	1024	256	1873	1
4	1024	128	1905	1,02
5	1024	64	2440	1,21
6	1024	32	4017	2,15
7	1024	24	5130	2,74
8	1024	16	7404	3,96
9	1024	8	14339	7,67

Результаты, приведенные в таблице 1, получены при числе портов коммутатора, равном числу N узлов кластера. Здесь T_{Pair} – среднее время полного цикла обслуживания запроса – от его отправки до получения ответа при учете задержек при занятии портов коммутатора; $T_1 = 2 \cdot (t_{cp} + t_0)$ – среднее время полного обслуживания запроса без учета задержек, появление которых обусловлено конфликтами между запросами при попытках занятия портов коммутатора. В последней колонке таблицы 1 приведено значение коэффициента замедления обработки запроса из-за указанных конфликтов $K_{Slow} = T_{Pair} / T_1$. Полученные в результате статистических экспериментов значения данного коэффициента показывают большую зависимость замедления обслуживания запросов от числа участников «Круглого стола», которое выбрано равным числу узлов кластера и числу портов коммутатора.

В реальной ситуации число портов может оказаться избыточным, поэтому была проведена еще одна серия статистических экспериментов при числе узлов и участников вычислительного кластера $N = 1024$ и при изменении числа портов коммутатора N_P . В последней колонке иллюстрируется зависимость значения коэффициента K_{Lim} от уменьшения числа портов коммутатора. Здесь $K_{Lim} = T_{Pair} / T_{Pair,min}$. Например, при $T_{Pair} = 1905$ $K_{Lim} = 1905 / 1870 = 1,02$. Результаты таблицы 2 в целом показывают, что число портов коммутатора может быть уменьшено до 64 без заметных потерь в производительности.

Предложенные и реализованные методы, новые исполнимые модели вычислительных систем кластерного типа на макроуровне позволяют расширить диапазон их функциональности и применимости: логико-алгебраические операционные и логико-вероятностные статистические модели «Круглый стол», «Резидент – агенты», «Собрание 1», «Собрание 2».

Проведенные статистические эксперименты с данными моделями позволяют оценить характеристики производительности при различных уровнях параллельной обработки данных в кластерах, выбрать число узлов кластера и портов коммутатора.

В третьей главе предложены новые автоматные, вероятностные автоматные и логико-алгебраические операционные модели вычислительных систем кластерного типа на *микроуровне*. В этих моделях учитываются особенности работы приложений на уровне сетевых операций, и на их основе возможно давать оценку производительности кластера в целом при реализации последовательно-параллельных, параллельно-конвейерных и других приложений.

На основе исполнимых автоматных и логико-алгебраических моделей рассмотрено выполнение на кластере последовательно-параллельных и параллельно-конвейерных приложений с реализацией произвольных алгоритмов. Построение и исследование автоматных моделей на основе конечных частичных автоматов Мура проведены для методов в общих рамках модели SPMD (Single Program, Multiple Data – одиночная («клонированная») программа, несколько потоков данных). В дальнейшем построены и другие исполнимые модели на основе логико-алгебраического подхода в рамках методов MPMD (Multiple Programs, Multiple Data – множество программ, множество различных данных).

В предлагаемых автоматных СКУ-моделях (СКУ – системы канонических уравнений, которые в качестве начального языка предложил использовать Н. П. Вашкевич; вопросы использования автоматных моделей в программировании развиты в работах А. А. Шалыто) использованы следующие понятия. Операторам поставлены во взаимно-однозначное соответствие так называемые частные события, представленные унарными предикатами вида $A_i(t)$, определенными на множестве значений дискретного времени t . Частичные входные переменные, или входные условия, определяющие условные переходы, представлены унарными предикатами вида $x_j(t)$, также определенными на множестве значений дискретного времени t . Введены также унарные предикаты вида $z_k(t)$, которые могут принимать истинные значения только после того, как

соответствующие им события вида $A_k(t)$ уже произошли, что соответствует тому факту, что оператор A_k завершил свою работу. Таким образом, при $z_k(t) = 0$ (ложь) событие $A_k(t)$ сохраняется, а при $z_k(t) = 1$ (истина) – не сохраняется. Первое условие для события $A_k(t)$ позволяет продлить его выполнение, а при выполнении второго, противоположного, условия событие $A_k(t)$ завершается. Условие зарождения события соответствует переходу из предшествующего ему события.

Последовательно-параллельная сеть частичных автоматов с вложенными состояниями, описывающая выполнение приложения в кластере в режиме обработки SPMD, представлена рисунке 1. Сетевая SKU-модель последовательно-параллельной сети автоматов представлена выражением следующего вида:

$$SPMD1: SP_{Seq}^*(ParReplicate(1..8)MD_{Clon}), \quad (4)$$

где SP_{Seq} – SKU частичного подавтомата, описывающая начальный последовательный участок приложения; $(ParReplicate(1..8)MD_{Clon})$ – SKU для частичных подавтоматов, соответствующих группе параллельных структурированных состояний $Group\ C_1-C_8$. Звездочке (*) соответствует операция последовательной композиции автоматов.

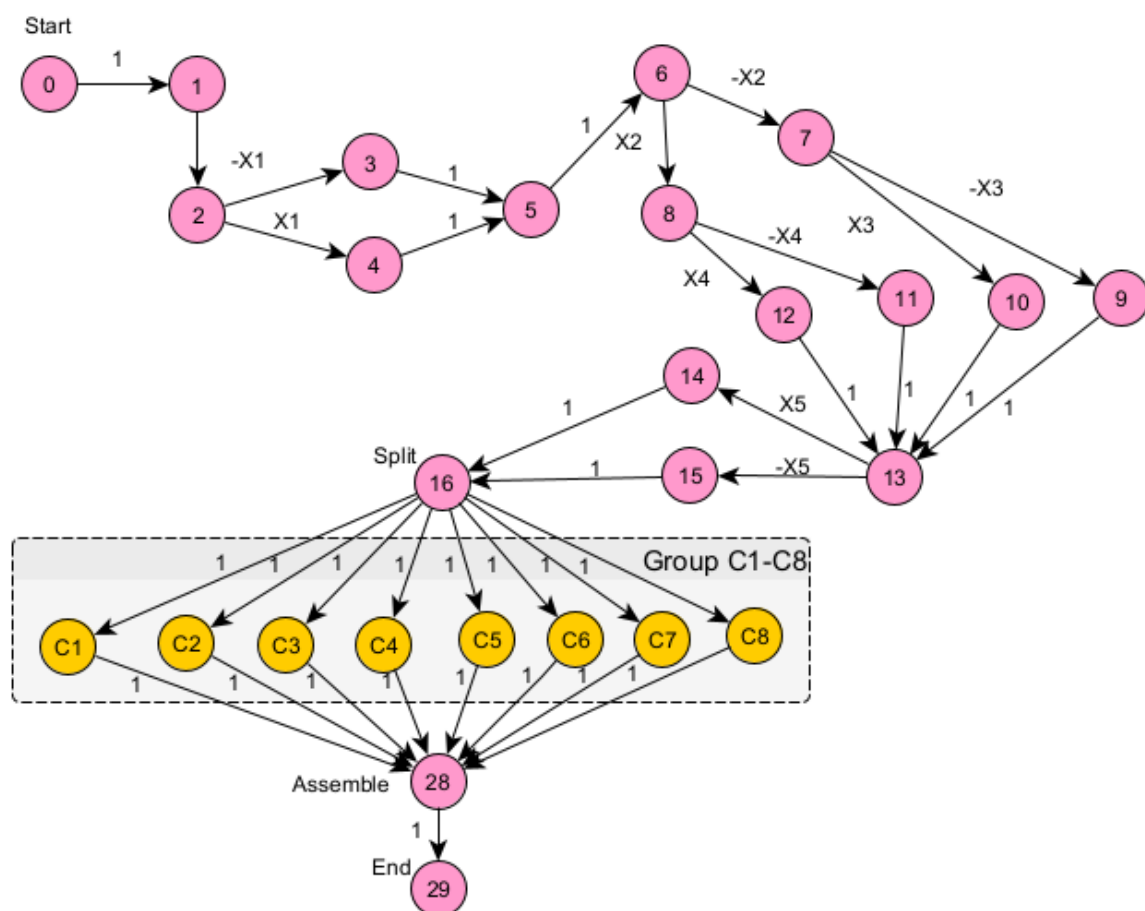


Рисунок 1 – Последовательно-параллельная сеть автоматов с вложенными состояниями выполнения приложения в кластере в режиме обработки SPMD

Полученные рекуррентные предикатные уравнения SKU SP_{Seq} для модели начального последовательного участка приложения имеют следующий вид:

$$\begin{aligned}
A_0(t+1) &= A_0(t) \& \neg x_0(t) \vee x_{begin}(t); \\
A_1(t+1) &= A_0(t) \& x_0(t) \vee A_1(t) \& \neg z_1(t); \\
A_2(t+1) &= A_1(t) \& z_1(t) \vee A_2(t) \& \neg z_2(t); \\
&\dots \\
A_{15}(t+1) &= A_{13}(t) \& z_{13}(t) \& \neg x_5(t) \vee A_{15}(t) \& \neg z_{15}(t); \\
A_{16}(t+1) &= A_{14}(t) \& z_{14}(t) \vee A_{15}(t) \& z_{15}(t) \vee A_{16}(t) \& \neg z_{16}(t).
\end{aligned}$$

Уравнение для события A_{16} связывает SKU SP_{Seq} и SKU MD_{Clon} в единую SKU для сетевой автоматной модели. Рекуррентные предикатные уравнения автоматной SKU MD_{Clon} имеют следующий вид:

$$\begin{aligned}
A_{17}(t+1) &= A_{16}(t) \& z_{16}(t) \vee A_{17}(t) \& \neg z_{17}(t); \\
A_{18}(t+1) &= A_{17}(t) \& z_{17}(t) \& x_6(t) \vee A_{20}(t) \& z_{20}(t) \vee \\
&\vee A_{23}(t) \& z_{23}(t) \vee A_{18}(t) \& \neg z_{18}(t); \\
A_{19}(t+1) &= A_{18}(t) \& z_{18}(t) \& x_7(t) \vee A_{19}(t) \& \neg z_{19}(t); \\
A_{20}(t+1) &= A_{19}(t) \& z_{19}(t) \& x_8(t) \vee A_{20}(t) \& \neg z_{20}(t); \\
A_{21}(t+1) &= A_{17}(t) \& z_{17}(t) \& \neg x_6(t) \vee A_{21}(t) \& \neg z_{21}(t); \\
A_{22}(t+1) &= A_{18}(t) \& z_{18}(t) \& \neg x_7(t) \vee A_{22}(t) \& \neg z_{22}(t); \\
A_{23}(t+1) &= A_{22}(t) \& z_{22}(t) \& \neg x_9(t) \vee A_{23}(t) \& \neg z_{23}(t); \\
A_{24}(t+1) &= A_{19}(t) \& z_{19}(t) \& \neg x_8(t) \vee A_{22}(t) \& z_{22}(t) \& x_9(t) \vee \\
&\vee A_{21}(t) \& \neg z_{21}(t) \vee A_{24}(t) \& \neg z_{24}(t); \\
A_{25}(t+1) &= A_{24}(t) \& z_{24}(t) \& \neg x_{10}(t) \vee A_{25}(t) \& \neg z_{25}(t); \\
A_{26}(t+1) &= A_{24}(t) \& z_{24}(t) \& x_{10}(t) \vee A_{26}(t) \& \neg z_{26}(t); \\
A_{27}(t+1) &= A_{25}(t) \& z_{25}(t) \vee A_{26}(t) \& z_{26}(t) \vee A_{27}(t) \& \neg z_{27}(t).
\end{aligned}$$

При включении (реплицировании) SKU MD_{Clon} в общую SKU к номерам имен всех предикатов автоматически добавляются вторые индексы – номера параллельных подавтоматов.

Копии модуля приложения, составленного по SKU SP_{Seq} , загружаются на все вычислительные узлы кластера и выполняются в параллельном режиме, обрабатывая одни и те же данные, или осуществляют ввод однотипных данных.

Достоинством метода формализации алгоритмов с использованием систем канонических уравнений является компактное логическое описание функций переходов, значительно упрощающее реализацию программной

модели. Аналогично структурированным состояниям C_1 – C_8 вводятся одноименные структурированные события. «Вложением» в каждое структурированное событие являются уже составленные автоматные СКУ MD_{Clon} .

Зарождение и продолжение событий C_1 – C_8 (инициирование и параллельная работа не зависящих друг от друга программных модулей) описываются СКУ S_C .

Системой канонических уравнений S_D описываются зарождение и сохранение событий.

Последнее уравнение D_9 описывает процесс ожидания наступления всех событий-индикаторов D_1 – D_8 завершения параллельной работы всех ветвей.

Уравнения, описывающие переход к завершающим событиям A_{28} , A_{29} и далее к событию A_1 управляющего модуля, имеют следующий вид:

$$A_{28}(t + 1) = D_9(t) \& v_9(t) \vee A_{28}(t) \& \neg z_{28}(t);$$

$$A_{29}(t + 1) = A_{28}(t) \& z_{28}(t) \vee A_{29}(t) \& \neg z_{29}(t).$$

$$A_1(t + 1) = A_{29}(t) \& z_{29}(t) \vee A_1(t) \& \neg z_1(t).$$

Для реализации общей сетевой СКУ-модели выполнения приложения в кластере в режиме последовательно-параллельной обработки SPMD все уравнения необходимо объединить в следующей последовательности:

$$SPMD2: SP_{Seq} * S_C * S_D * (ParReplicate(1..8) MD_{Clon}) * D_9 * A_{28} * A_{29}. \quad (5)$$

Результирующая общая СКУ-модель сети автоматов выполнения приложения в кластере в режиме последовательно-параллельной обработки SPMD относится к классу исполнимых моделей. На ее основе построена статистическая имитационная модель, позволяющая исследовать функционирование кластерной системы на микроуровне.

Результаты статистического моделирования сведены в таблице 3. Коэффициент ускорения k вычислен при указанном значении времени $T = t_{Посл.}$ выполнения последовательного участка, выраженном в единицах модельного времени (здесь – в миллисекундах, мс). Вычисленные значения коэффициента ускорения по результатам статистических экспериментов полностью соответствуют вычисленным значениям по формуле Амдала: $k = (t_{Посл.} + N \cdot t_N) / (t_{Посл.} + t_N)$, что является одним из критериев правильности разработанной модели.

Таблица 3

Число узлов N	$T = 0$	$T = 10$	$T = 20$	$T = 30$	$T = 40$	$T = 50$
1	1	1	1	1	1	1
2	2	1,98	1,96	1,94	1,92	1,9
4	4	3,88	3,77	3,67	3,57	3,48
8	8	7,46	7,0	6,6	6,25	5,93
16	16	13,8	12,25	11,0	10,0	9,25
32	32	24,2	19,6	16,5	14,3	12,6

Организация работы и моделирование вычислительного кластера в режиме MPMD относятся к сложным задачам. Для ускорения вычислений предлагается организовать выполнение приложений при их запуске в параллельно-конвейерном режиме NETWORK-MPMD (рисунок 2). Запуск вычислительного кластера в режиме MPMD соответствует случаю, когда на разных узлах кластера в локальной сети выполняются разные программы, реализующие модули одного и того же распределенного алгоритма. В процессе реализации сетевого алгоритма модули передают друг другу управление и при необходимости промежуточные результаты. Логико-алгебраическая операционная модель построена при использовании графа переходов состояний, изображенного на рисунке 2, для чего необходимо было соблюдать правило формирования условий операторами, в том числе операторами ввода условий.

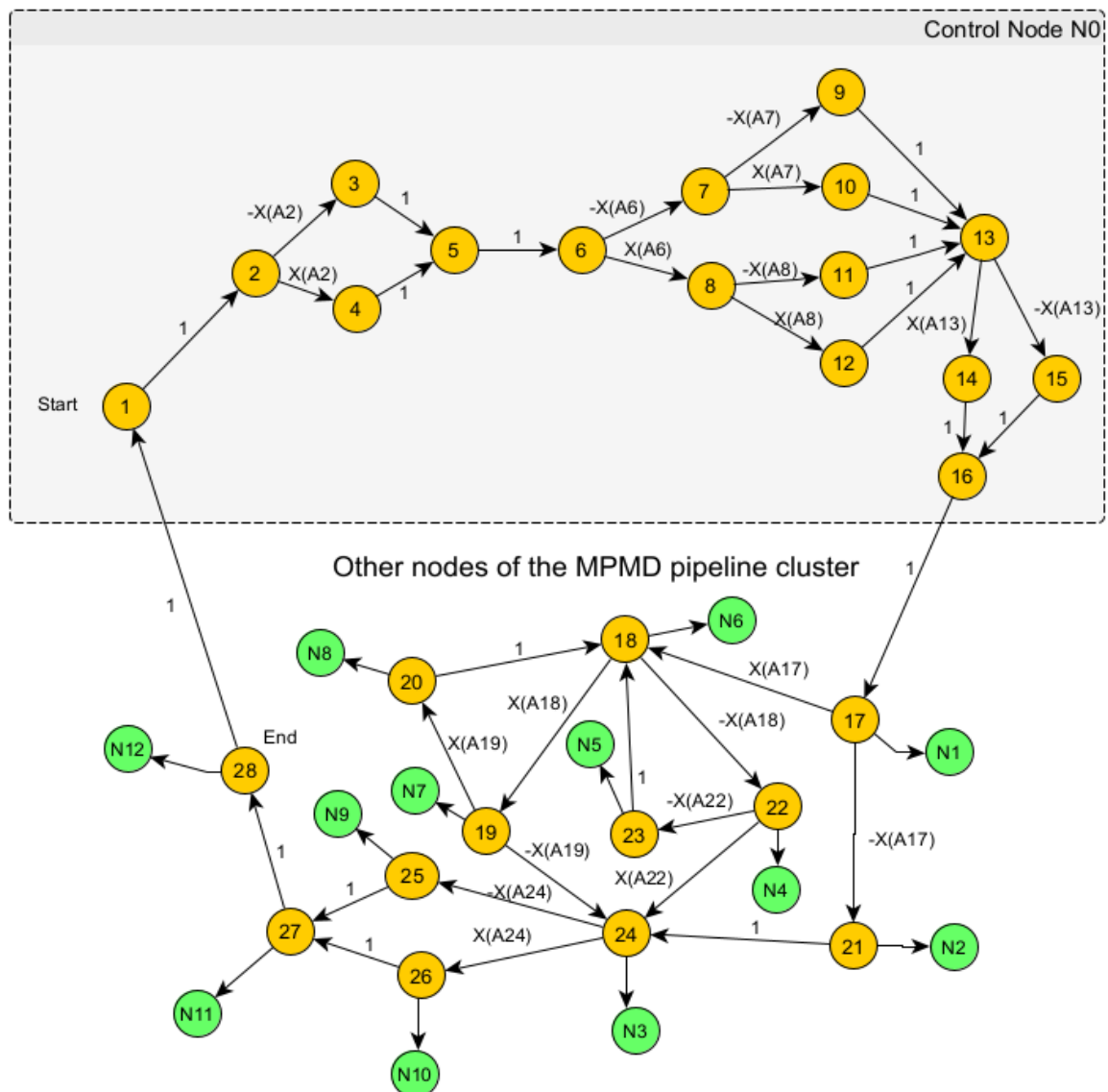


Рисунок 2 – Граф переходов состояний вычислительного кластера, соответствующий работе распределенного многомодульного приложения в параллельно-конвейерном режиме NETWORK-MPMD (нижняя часть рисунка); последовательный подготовительный этап (верхняя часть)

Система NETWORK-MPMD логико-алгебраических операционных выражений для сетевого приложения содержит формализованные сетевые спецификации. Для проверки корректности работы кластера в параллельно-конвейерном режиме использовался дополнительно аппарат сетей Петри.

Обобщенные результаты статистических экспериментов с моделью NETWORK-MPMD параллельно-конвейерного выполнения приложения сведены в таблице 4. Здесь T – интервал между последовательными моментами запуска однотипных (но с разными данными) независимых по управлению и данным сетевых кластерных приложений; управляющая системная программа отправляет очередное приложение на обработку в сеть в соответствии с графом переходов состояний на рисунке 2; λ – интенсивность инициализаций приложений; $T_{\text{ср.}}$ – оценка математического ожидания времени выполнения одного приложения из нескольких, задействованных в параллельно-последовательном режиме; K_F – коэффициент, характеризующий относительный выигрыш по интенсивности запуска приложений при организации работы кластера в параллельно-конвейерном режиме, это в данном случае главный параметр, характеризующий эффективность использования вычислительного кластера; $K_{\text{зам.}}$ – коэффициент, характеризующий замедление работы приложений при взаимном влиянии друг на друга при ожидании начала обработки в очередях к узлам.

Таблица 4

T (мс)	λ (1/с)	$T_{\text{ср.}}$ (мс)	K_F	$K_{\text{зам.}}$
50	20	49,4	1	1
20	50	51,1	1,07	1,03
10	100	63,7	1,20	1,29
9	111,1	72,5	1,33	1,47
8	125	95,3	2,78	1,93
7,5	133,3	134,7	6,66	2,72

Например, при значении $T = 8$ мс получена оценка среднего времени выполнения приложения $T_{\text{ср.}} = 95,3$, значение $K_F = 125/20 = 2,78$ (т.е. интенсивность λ потока запусков приложений увеличилась почти в 3 раза по сравнению с последовательной работой приложений, которая наблюдалась при $\lambda = 20$ 1/с); $K_{\text{зам.}} = 95,3/49,4 = 1,93$ – это значение определяется «накладными расходами» на организацию очередей ожидания к узлам, на которых расположена та или иная часть приложения, реализующая соответствующий оператор распределенного приложения.

Созданные исполнимые имитационные модели кластерных вычислительных систем позволяют оценить основные временные характеристики и загрузку узлов, что особенно важно при использовании кластера при конвейерном запуске приложений. Отличительной особенностью разработанных

моделей является то, что основой их построения являются автоматная и логико-вероятностная модели выполнения кластерного приложения, которые легко модифицируются и после некоторых дополнений могут использоваться в качестве исполнимых формализованных спецификаций при создании реального приложения.

В четвертой главе предлагаются метод и модели функциональной организации вычислительных систем кластерного типа с повышенной отказоустойчивостью. Целью данной главы являются выбор и дальнейшая реализация подхода к формированию работоспособной (рабочей) структуры вычислительного кластера в условиях его частичной деградации, возможно, вызванной не только отказами программного обеспечения и оборудования, но и перегрузкой компьютеров. Для упрощения программирования, доступного для прикладного или системного программиста, предложена масштабируемая логико-алгебраическая операционная исполнимая модель процесса восстановления работоспособности вычислительного кластера.

В реальном приложении в качестве констант (имен узлов) $y_0, y_1, y_2, \dots, y_k$ рассматриваются *MAC*-адреса (*Medium Access Control addresses* – адреса управления доступом к среде передачи данных), или аппаратные адреса Ethernet, – физические адреса узлов вычислительного кластера, являющихся единицами сетевого оборудования в стандарте компьютерных сетей Ethernet. В качестве адресов при передаче данных внутри подсетей для идентификации узлов кластера как компонентов сети InfiniBand используются *LID* (*Local ID*, локальные идентификаторы), которые также можно использовать в качестве предметных констант, т.е. значений предметной переменной y . Имена подзадач, решаемых узлами кластера, задаются в модели множеством $X = \{x_1, x_2, \dots, x_N\}$. Для множества X введена характеристическая функция $Q(x)$.

Выполнение модели происходит в четыре этапа.

Этап 1. Управляющий узел вычислительного кластера определяет готовые к использованию узлы, которые имеются в наличии, путем опроса каждого; от каждого узла ожидается ответ о готовности до истечения тайм-аута t_1 . По завершении первого этапа модели должно быть сформировано множество Y_1 с характеристической функцией $S_1(y)$, содержащее все работоспособные узлы, способные работать в кластере. Далее рассматриваются следующие ситуации: если узлов достаточно для образования кластера, то следом выполнится загрузка узлов программами и данными; если годных узлов оказалось больше, чем было необходимо для решения задачи, то они включаются в резерв, и, наконец, если узлов меньше минимально необходимого числа для решения задачи, то управляющий узел, а через него и пользователь оповещаются об этом.

Этап 2. Формирование состава рабочих узлов кластера, именование или нумерация этих узлов для использования имен или номеров в программах. Происходит аварийный выход, если не хватает узлов для формирования кластера для заранее определенного числа подзадач X ; отношение R_0 , связываю-

щее подзадачи из множества X с узлами кластера из множества Y_1 , передается всем узлам кластера.

Этап 3. Загрузка программ и данных в узлы кластера в режиме широковещания; последующая обработка данных в каждом узле кластера; передача результатов в управляющий узел y_0 ; формирование и передача в узел y_0 значений предиката $R_2(x, y)$, связывающего результаты выполнения подзадач с работоспособными узлами.

Этап 4. Производится проверка результатов на наличие узлов, не выполнивших свои подзадачи. Если все подзадачи выполнены, что соответствует в модели истинности всех значений предиката $R_2(x, y)$, то работа кластера признается успешной с выдачей результата пользователю через интерфейс *middleware*. В противном случае производятся исключение отказавших узлов из множества Y , определение неисполненных подзадач, исключение уже исполненных запросов из множества X и повторное выполнение для неисполненных запросов действий, предписанных логико-алгебраическими выражениями, которыми задаются сценарии действий этапов 1, 2 и 3.

Модель допускает вариативность и масштабирование – от простого варианта, когда состав рабочих узлов к использованию узлов кластера определяется однократно перед началом обработки подзадач, на которые разбита задача, до варианта, когда отказы возникают при выполнении подзадач, но выполнение таких не выполненных ранее задач повторяется до тех пор, пока не останется, по крайней мере, один годный узел в условиях деградации кластера, вызванной внешними и внутренними причинами (отказами, перегрузками узлов и др.).

Модель расширена на случай анализа отказоустойчивости систем распределенных грид-вычислений и волонтерских кластеров с большим и нестабильным числом узлов (до 100–150 и более вычислительных узлов).

Статистическая модель построена на основе использования логико-алгебраической операционной исполнимой модели и включает четыре описанных выше этапа их применения. Статистическая модель работы вычислительного кластера проверялась путем вычисления некоторых ее характеристик на основании формулы Бернулли.

Статистические эксперименты показали, что работоспособность кластера сохраняется при его деградации до одного узла с соответствующим понижением производительности, но позволяет выполнить задание полностью в большинстве случаев. Результат приведенного анализа базового примера показал, что в условиях сильной, но не стопроцентной, деградации вычислительного кластера было не только обеспечено выполнение разбитого на части задания, но и достигнуто ускорение вычислений в 5,33 раза за счет частичного сохранения параллельного режима. Модель расширена на случай анализа отказоустойчивости систем распределенных грид-вычислений и волонтерских кластеров с большим и нестабильным числом узлов.

В заключении обобщены основные теоретические и практические результаты диссертационной работы.

В приложении А приведены примеры реализации исполнимых моделей вычислительных кластеров на языке высокого уровня и на языке микроассемблера для микроконтроллеров.

Приложение Б содержит акты о внедрении результатов диссертационной работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ И ВЫВОДЫ

1. На основе проведенного анализа современного состояния и развития архитектуры вычислительных систем предложена концепция организации функциональной архитектуры вычислительных систем кластерного типа, *отличающаяся* базированием на исполнимых моделях, знаниях и правилах предметной области и *позволяющая* осуществлять предметную ориентацию вычислительных кластеров на конкретного пользователя, использующего кластер как конечный продукт.

2. Предложен метод разработки функциональной архитектуры вычислительных кластеров, основанный на исполнимых логико-алгебраических операционных моделях, *отличающийся* от известных тем, что он основан на знаниях и правилах предметной области, на назначении вычислительных кластеров и на анализе событий, происходящих при работе приложений и управляющих программ, и *позволяющий* создавать новые исполнимые модели для совершенствования вычислительных систем кластерного типа, определяющие их функциональную архитектуру, масштабируемость и соответствие предметной области.

3. Предложены новые исполнимые модели вычислительных систем кластерного типа на *макроуровне*, расширяющие диапазон их функциональности и применимости: логико-алгебраические операционные и статистические модели «Круглый стол», «Резидент – агенты», «Собрание 1», «Собрание 2», *позволяющие* реализовывать современные диалоговые параллельные взаимодействия при обработке данных в больших сообществах пользователей при помощи масштабируемых приложений, реализуемых кластером. Время выполнения статистических моделей при 1000–10 000 прогонах занимает не более 2–25 с, что позволяет оперативно получать характеристики производительности. Проведенные статистические эксперименты с данными моделями позволяют оценить характеристики производительности при различных уровнях параллельной обработки данных в кластерах, выбрать число узлов кластера и портов коммутатора вычислительных кластеров и других вычислительных систем с аналогичной системной и функциональной архитектурой.

4. Разработаны новые детализированные сетевые автоматные, вероятностные автоматные и логико-алгебраические операционные модели вычислительных систем кластерного типа на *микроуровне*, *отличающиеся* исполнимым

характером и *позволяющие* произвести проектирование приложений на уровне сетевых операций и дать оценку производительности кластера в целом. Статистические эксперименты продемонстрировали полное соответствие полученных характеристик производительности для базовых примеров закону Амдала.

5. Предложенный метод повышения производительности вычислительных кластеров при выполнении сетевых приложений *позволил*, в отличие от известных, повысить их пропускную способность при развертывании операторов распределенного алгоритма на узлах кластера и выполнять вычисления, определяемые передачами управления и данных между узлами в параллельно-конвейерном режиме. Статистические эксперименты продемонстрировали повышение пропускной способности в 2–5 раз.

6. Предложена исполнимая логико-алгебраическая операционная модель функциональной организации вычислительных систем кластерного типа с повышенной отказоустойчивостью, основанная на динамическом резервировании и многократной замене узлов и позволяющая при ее реализации обеспечить отказоустойчивость кластера в условиях сильной деградации, обусловленной внутренними и внешними причинами. Статистические эксперименты показали, что работоспособность кластера сохраняется при его деградации до одного узла с соответствующим понижением производительности, но позволяет выполнить задание полностью в большинстве случаев. Результат проведенного анализа типового примера показал, что в условиях сильной, но не стопроцентной, деградации вычислительного кластера было не только обеспечено выполнение разбитого на части задания, но и достигнуто ускорение вычислений в 5,33 раза за счет частичного сохранения параллельного режима. Модель расширена на случай анализа отказоустойчивости систем распределенных грид-вычислений и волонтерских кластеров с большим и нестабильным числом узлов (до 100–150 и более вычислительных узлов).

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в изданиях из перечня ВАК при Минобрнауки России

1. Петушков, Г. В. Формализация и реализация логико-вероятностных и логико-алгебраических операционных моделей функциональной архитектуры кластерных вычислительных систем / Г. В. Петушков, А. С. Сигов // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2025. – № 3. – С. 26–62. – ISSN 2072-3059. (K2)

2. Петушков, Г. В. Анализ и выбор структуры многопроцессорной вычислительной системы по критерию быстродействия / Г. В. Петушков, А. С. Сигов // Russian Technological Journal. – 2024. – Т. 12, № 6. – С. 20–25. – ISSN 2782-3210. (K1)

3. Петушков, Г. В. Техничко-экономический анализ серверов как вычислительных модулей вычислительных систем класса WSC / Г. В. Петушков,

А. С. Сигов // Russian Technological Journal. – 2025. – Т. 13, № 1. – С. 49–58. – ISSN 2782-3210. (K1)

4. **Петушков, Г. В.** Организация кластерных вычислительных систем с функциональной архитектурой, определяемой исполнимыми моделями. Логико-алгебраические исполнимые модели обработки информации / **Г. В. Петушков**, А. С. Сигов // XXI век: итоги прошлого и проблемы настоящего плюс. – 2025. – Т. 14, № 3 (71). – С. 10–22. – ISSN 2221-951X. (K2)

5. **Петушков, Г. В.** Функциональная организация вычислительных кластеров с повышенной отказоустойчивостью / **Г. В. Петушков**, А. С. Сигов // XXI век: итоги прошлого и проблемы настоящего плюс. – 2025. – Т. 14, № 3 (71). – С. 55–64. – ISSN 2221-951X. (K2)

6. **Петушков, Г. В.** Методика анализа надежности и мониторинга функционирования вычислительных систем / **Г. В. Петушков** // Известия Юго-Западного государственного университета. – 2025. – Т. 29, № 1. – С. 155–172. – ISSN 2223-1560. (K2)

7. **Петушков, Г. В.** Оценка производительности вычислительной системы / **Г. В. Петушков** // Известия Юго-Западного государственного университета. – 2025. – Т. 29, № 2. – С. 201–220. – ISSN 2223-1560. (K2)

8. **Петушков, Г. В.** Организация и исследование кластерных вычислительных систем с функциональной архитектурой, определяемой исполнимыми моделями. Автоматные исполнимые модели обработки информации / **Г. В. Петушков** // Russian Technological Journal. – 2025. – Т. 13, № 6. – С. 7–24. – ISSN 2782-3210. (K1)

9. Коваленко, С. М. Развитие информационных технологий – ближайшие перспективы / С. М. Коваленко, О. В. Платонова, **Г. В. Петушков** // Научно-технический вестник Поволжья. – 2018. – № 10. – С. 172–174. – ISSN 2079-5920. (K2)

10. Коваленко, С. М. Перспективы развития аппаратной платформы компьютеров / С. М. Коваленко, А. М. Романов, **Г. В. Петушков** // Современные наукоемкие технологии. – 2018. – № 12. – С. 67–70. – ISSN 1812-7320. (K2)

Публикации в других изданиях

11. Kovalenko, S. M. Structure Selection Technique of Multi-Processor Computing Systems / S. M. Kovalenko, **G. V. Petushkov**, O. V. Platonova // International Journal of Engineering and Technology. – 2018. – Vol. 7, Iss. 4. – P. 29–31. – ISSN 2319-8613.

12. Kovalenko, S. M. Method of Estimating the Reliability of Highly Reliable Redundant Systems Based on Continuous Models / S. M. Kovalenko, **G. V. Petushkov**, O. V. Platonova // International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies. – 2019. – Vol. 10, Iss. 7. – P. 913–916. – ISSN 2228-9860.

13. Коваленко, С. М. Прогнозирование надежности по отношению к отказам программного обеспечения для сложных программно-аппаратных

систем / С. М. Коваленко, **Г. В. Петушков**, О. В. Платонова, М. М. Расулов // ИТ-Стандарт 2019 : сборник трудов IX Международной научной конференции. – Москва : Проспект, 2019. – С. 212–215. – ISBN 978-5-98597-402-7.

Учебное пособие

1. Коваленко, С. М. Проектирование вычислительных комплексов : учебное пособие / С. М. Коваленко, Ю. С. Асадова, **Г. В. Петушков**. – Москва : РТУ МИРЭА, 2023. – ISBN 978-5-7339-1749-8. – URL: <https://e.lanbook.com/book/368978>

Научное издание

ПЕТУШКОВ Григорий Валерьевич

**ФУНКЦИОНАЛЬНАЯ АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
КЛАСТЕРНОГО ТИПА НА ОСНОВЕ ИСПОЛНИМЫХ МОДЕЛЕЙ
И ПРАВИЛ ПРЕДМЕТНОЙ ОБЛАСТИ**

Специальность 2.3.2. Вычислительные системы и их элементы
(технические науки)

Редактор *Т. Н. Судовчихина*
Технический редактор *Р. Б. Бердникова*
Компьютерная верстка *Р. Б. Бердниковой*

Подписано в печать 24.10.2025. Формат 60×84¹/₁₆.
Усл. печ. л. 1,4. Заказ № 535. Тираж 100.

Издательство ПГУ.
440026, г. Пенза, ул. Красная, 40.
Тел./факс: (8412) 66-60-49, 66-67-77; e-mail: iic@pnzgu.ru